Supervised Scaled Regression Clustering: an Alternative to Neural Networks

Mark J. Embrechts (embrem@rpi.edu)
Department of Decision Sciences and Engineering Systems
Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Dirk Devogelaere (dirk.devogelaere@cit.kuleuven.ac.be) and Marcel Rijckaert
Department of Chemical Engineering
University of Leuven, De Croylaan 46, 3001 Heverlee, Belgium

ABSTRACT:

This paper describes a rather novel method for the supervised training of regression systems that can be an alternative to feedforward Artificial Neural Networks (ANNs) trained with the BackPropagation algorithm. The proposed methodology is a hybrid structure based on supervised clustering with genetic algorithms and local learning. Supervised Scaled Regression Clustering with Genetic Algorithms (SSRCGA) offers certain advantages related to robustness, generalization performance, feature selection, explanative behavior, and the additional flexibility of defining the fitness function and the regularization constraints. Computational results of SSRCGA are compared with backpropagation trained ANNs on a real-life environmental multivariate regression task.

KEYWORDS: neural networks, genetic algorithms, clustering, local modelling, prediction

INTRODUCTION

This paper describes a model for a regression analysis tool that can be seen as a kind of multi-strategy data analysis system and therefore can be used as an alternative to feedforward Artificial Neural Networks (ANNs). Supervised Scaled Regression Clustering with Genetic Algorithms (SSRCGA) is a hybrid model based on a GA semi-supervised clustering algorithm [1] augmented with local learning. The local learning in this method is supervised in the sense that the prediction quality is incorporated as a penalty term added to the fitness function of the Genetic Algorithm (GA). SSRCGA offers certain advantages related to robustness, generalization performance, feature selection, explanative behavior, and the additional flexibility of defining the fitness function with or without regularization constraints. This paper introduces the SSRCGA methodology by discussing in succession: (i) local learning, (ii) clustering with GAs for a variable number of clusters and (iii) scaled supervised regression clustering. The data analysis task concerns the environmental problem of determining the state of rivers and streams by monitoring and analyzing certain measurable chemical concentrations with the goal of inferring the biological state of the river, namely the density of algae communities. Typical of such real-life problems (prediction of seven different algae frequency distributions), the particular data set contains a mixture of qualitative (river size and its velocity), linguistic (season when the sample was taken) and numerical measurements values (chemical concentrations), with much of the data being incomplete. This paper demonstrates that the SSRCGA method compares favorably with ANNs trained with the BackPropagation (BP) algorithm [2] for this River Pollution Data set [3].

LOCAL LEARNING

Local learning [4] [5] belongs to a data analytic methodology whose basic idea lies behind obtaining the prediction for a case i (with vector coordinates $\mathbf{x}_i$) by fitting a parametric function in its neighborhood. This means that these methods are ‘locally parametric’ as opposed to, for instance, least squares linear regression. Moreover, these methods do not produce a ‘visible’ model of the data. Instead they make predictions based on local models generated on a query point basis. In spite of being considered a non-parametric regression technique, local learning does have several ‘parameters’ that must be tuned in order to obtain good predictive results. One of the most important is the notion of neighborhood. Given a query point q, we need to decide which training cases will be used to fit a local polynomial around the query point. This involves defining a distance metric over the multidimensional

0-7695-0619-4/00/$10.00 (C) 2000 IEEE

space defined by the input variables. With this metric, we can specify a distance function that allows finding the nearest training cases of any query point. Still, many issues remain open. Namely, weighting of the variables within the distance calculation can be crucial in domains with less relevant variables. Moreover, we need to specify how many training cases (L) will enter the local fit (usually known as the bandwidth selection problem, generally chosen as 3 or 5). Even after having a bandwidth size specification, we need to weight the contribution of the training cases within the bandwidth. Nearer points should contribute more into the local fit. This is usually accomplished through a weighting function (distance weighting factor d) that takes the distance to the query point into account (known as the kernel function). The outcome (o_i) for a case i (with vector coordinates $\mathbf{x}_i$) can now be estimated by local learning from the target outcomes of its L nearest neighbors (t_l) according to:

$$\hat{o}_i = \sum_{\ell=1}^{L} \left\| \vec{x}_i - \vec{x}_\ell \right\|^d t_\ell \Big/ \sum_{\ell=1}^{L} \left\| \vec{x}_i - \vec{x}_\ell \right\|^d$$

The first factor in the denominator of the expression above allows incorporating a distance-weighting scheme. Introducing the distance weighting factor (d) can control the specifics. For the traditional least squares error measure, the total regression error becomes the familiar

$$M_R = \sum_{i=1}^{N} \left(\hat{o}_i - t_i \right)^2$$

The correct tuning of all these modelling ‘parameters’ can be crucial for successful use of local learning.

GA-DRIVEN CLUSTERING WITH A VARIABLE CLUSTER NUMBER

Clustering is a classic machine learning problem. The most popular clustering method is the well-known K-means algorithm [6]. However, there are a number of good reasons to consider other clustering methods as well [7].

One alternative to the K-means clustering algorithm is to consider a genetic algorithm based clustering method where the GA determines the cluster centers in order to reduce the classical cluster dispersion measure (or any other measure related to cluster performance for that matter). A collection of N cases is partitioned into K groups according to:

$$J = \sum_{k=1}^{K} J_k = \sum_{k=1}^{K} \left(\sum_{i=1}^{N} \delta_{ik} \left\| \vec{x}_i - \vec{c}_k \right\|^2 \right)$$

where J is the cluster dispersion measure (to be minimized),
N is the number of cases,
K is the number of clusters,
δ_{ik} is 1 when case i belongs to cluster k, 0 otherwise,
$\mathbf{x}_i$ are the vector coordinates for case i,
$\mathbf{c}_k$ are the vector coordinates for cluster center k (to be determined).

It is straightforward to implement a genetic algorithm for “guessing” the cluster centers in order to minimize the objective function J. A genetic algorithm was implemented as a floating point GA with arithmetic cross-over and uniform mutation following Michalewicz [8]. The chromosomes of the GA represent the coordinates of the cluster centers. If the dimensionality of the data is D, and there are K cluster centers, there will be D*K chromosomes. While the selection of mutation and crossover rates is important for the performance of the GA, it was found that the GA is fairly robust with regard to the particular implementation details such as operator selection and reproduction schemes.

Note that so far the number of clusters was pre-determined. It is now possible to extend GA driven clustering to allow for a varying number of clusters [7]. Rather than following Bezdek’s suggestions, we had good success by starting out with a relatively large predescribed number of clusters and letting the number of clusters vary by adding a regularization term (i.e., in this case a penalty/bonus term for empty clusters) to the cluster dispersion, leading to the following fitness function:

$$\text{Fitness\_Function} = J \pm \gamma N_E$$

In the expression above, γ is a “dummy cluster” penalty/bonus factor and N_E is the number of empty clusters. A cluster is empty when it has no members. Such empty or “dummy clusters” do not effectively contribute to the cluster dispersion anymore. It depends on the particular application whether a penalty or bonus approach is more efficient. The choice of the penalty factor γ is determined by trial and error. We found generally acceptable performance when the contribution of regularization term to the cost function is of the same order of magnitude as the cluster dispersion measure.

SUPERVISED SCALED REGRESSION CLUSTERING

So far, a GA was introduced as an alternative to traditional clustering. The introduction of a dummy cluster regularization term offers an elegant way to vary the number of clusters and brings a significant advantage over traditional clustering methods. Up to this point, there is no supervised action going on. Combining the two former methods, we get a powerful prediction method. In a first step, the whole data set will be clustered and in each cluster the local learning method will be applied to calculate the outcome. Furthermore, the clustering itself will be influenced by the result of the local learning method. All that is needed in this case is to add an additional penalty term, related to the error measure, to the fitness function, according to:

$$\text{Fitness\_Function} = J \pm \gamma N_E + \alpha M_R$$

The later term in the expression above represents a penalty factor proportional to the total regression error (M_R). The proper choice for the regularization parameter (α) is problem dependent and needs to be specified by the user. α can be determined by trial and error. It was found that the particular choice for the regularization parameters is not crucial as long as each of the three terms in the cost function remains significant.

The GA driven regression clustering algorithm presented so far is now an alternative to a traditional feedforward ANN. One useful feature can still be added to regression clustering: *dimension scaling*. In the case that the data space has a very high dimensionality, it is generally desirable to reduce the dimensionality by selecting the most relevant features. Rather than combining the GA based regression clustering method with a traditional method for feature selection (e.g., by selecting the most correlated features with the outcomes), we propose to introduce adaptive scaling factors for each dimension. An easy way to implement this scheme is to add a number of chromosomes to the gene corresponding to the dimensionality (D). In order to discourage irrelevant features or dimensions, each dimension is multiplied by its corresponding scaling factor. The sum of the scaling factors is normalized to unity to avoid a trivial solution. The GA automatically adjusts appropriate scaling factors and the most relevant features for a particular application are the ones with the larger scaling factors. It is also possible to generalize this feature selection scheme further by assigning a different set of scaling factors to each cluster.

Supervised Scaled Regression Clustering with Genetic Algorithms (SSRCGA) has advantages and disadvantages compared to traditional neural network approaches. The advantages of SSRCGA relate to: (i) the simplicity of the idea; (ii) the flexibility of its implementation by allowing the user to modify the cost function and the penalty terms (e.g., the misclassification error measure); (iii) the possibility for a physical interpretation of what is going on; (iv) a straightforward methodology for feature selection via scaling; and (v) a good general performance, even for high-dimensional data. Disadvantages of the SSRCGA compared to ANNs relate to (i) possible excessive demands on computing time and memory; (ii) poor scaling of the speed of the algorithm with the number of data points; and (iii) the ad-hoc problem choice for problem dependent regularization parameters (i.e., penalty factors).

COMPUTATIONAL RESULTS FOR SSRCGA

In order to evaluate the performance of SSRCGA, two well-known problems were solved with SSRCGA and their results were compared with a traditional BP trained ANN [1]. These problems relate to the IRIS and the Wisconsin breast cancer data. The benchmark problems addressed in this paper are strictly speaking classification problems (rather than regression problems). In these comparative studies, the SSRCGA showed a better performance. In a challenging regression problem, a pilot version (GAdC) of SSRCGA [9] won the competition amongst 21 international entries.

In order to get a ‘fair’ comparison, the results (sum of the squared errors) are compared on the River Pollution Data set. For both methods, the missing values were replaced by the mean value. Comparing the results in this way makes it possible to set up a simple and clear regression benchmark comparison on a data set that everyone can access.

River Pollution Data

During the research study water quality samples were taken from sites on different European rivers of a period of approximately one year. This study is motivated by the increasing concern with the impact human activities are having on the environment. Identifying the key chemical control variables that influence the biological process associated with these algae has become a crucial task in order to reduce the impact of man activities. The samples were analyzed for various chemical substances. In parallel algae samples were collected to determine the algae population distributions. While the chemical analysis is cheap and easily automated, the biological part involves microscopic examination, requires trained manpower and is therefore both expensive and slow. The relationship between the chemical and biological descriptors is complex and can be expected to need the application of advanced techniques. Typical of such real-life problems, the particular data set for the problem contains a mixture of qualitative, linguistic and numerical measurement values, with much of the data being incomplete. The data analysis task consisted of prediction the algae frequency distribution on the basis of the measured concentrations of the chemical substances and the global information concerning the season when the sample was taken, the river size and its velocity. The two last variables are given as linguistic variables.

A very important step in solving this multivariate regression task is the data preprocessing. A disadvantage is that preprocessing in a certain way depends on the method that will be used later. For this reason a general method (because we want to compare SSRCGA and NN) is used to replace the missing values and all data were scaled (necessary for NN). This explains why the results obtained now are worse than those mentioned with GAdC in [9]. The river pollution data set [3] for this study consisted of 198 cases (after weeding out 2 cases, with most of the variables as missing data, from the original data set) used for training. Another set of 140 cases was used as blind set. There were 18 descriptors. The first descriptor (season) was not used in this study. The remaining 17 descriptors used. The last 7 descriptors of each data set are the distribution of different kinds of algae (AG1, AG2, … AG7) and represent the outputs to be predicted. The descriptors used to build a model are the river size, the fluid velocity (both categorical), and 8 chemical concentrations being nitrogen in the form of nitrates, nitrites and ammonia, phosphate, oxygen and others. The values of the categorical descriptors river size (small; medium; large) and fluid velocity (low; medium; high) were replaced by numerical values (0.0; 0.5; 1.0). The remaining missing values were replaced by the mean of the responding descriptor. All data was scaled (remember that normally the SSRCGA algorithm doesn’t need a scaled data set). Four hidden layers were used for the neural network study rather than the traditional one or two hidden layer structure to improve robustness. The neural network has a 10-8-8-8-5-1 neuron structure and was halted when the error fell below a threshold value. Four hidden layers rather than just one or two hidden layers were selected for the network in order to improve generalization and robustness. Seven different networks were built one for each outcome. A second neural network structure was built with nearly the same neuron structure. The difference is that this network has seven outputs, this means only one network is needed. The SSRCGA method started out with 6 clusters and used local learning by averring between the 5 nearest neighbors within that cluster. In the case that there were less than 5 neighbors within the cluster, all the available training samples for that cluster were used. The penalty factor for misclassification was set to 30 and the bonus factor for empty clusters was set to 5 (for outcomes 1, 2 and 3) and to 3 (for outcomes 4, 5, 6 and 7). These values were determined based on the general guideline that each penalty factor has to be significant in the cost function. The population size was set to 100 and the GA ran for 500 generations. The mutation and crossover probabilities were 0.03 and 0.8 respectively. A simple roulette selection procedure was followed for reproduction. Looking at the results (see Table 1) obtained for each of the algae and the overall mean squared error leads to the conclusion that our method tends to outperform the ANN.

Method	**AG1**	**AG2**	**AG3**	**AG4**	**AG5**	**AG6**	**AG7**	**global**
GAdC	186	104	29	8	72	173	21	84.7
SSRCGA	263	134	34	11	81	207	22	107.4
NN(10-8-8-8-5-1)	267	235	94	24	104	349	62	162.1
NN(10-8-8-8-5-7)	454	251	52	24	136	358	61	190.8

Table 1: The mean squared error

It is quite clear that building a neural network for each output descriptor is better than building one network for the whole problem. A scatterplot for the results of AG3, with on horizontal axis the desired response and on the vertical axis the actual, gives an indication about the region where our method performs better than the neural network method. Typical for the neural networks are the zero values even at higher values and some high values where low values are expected.

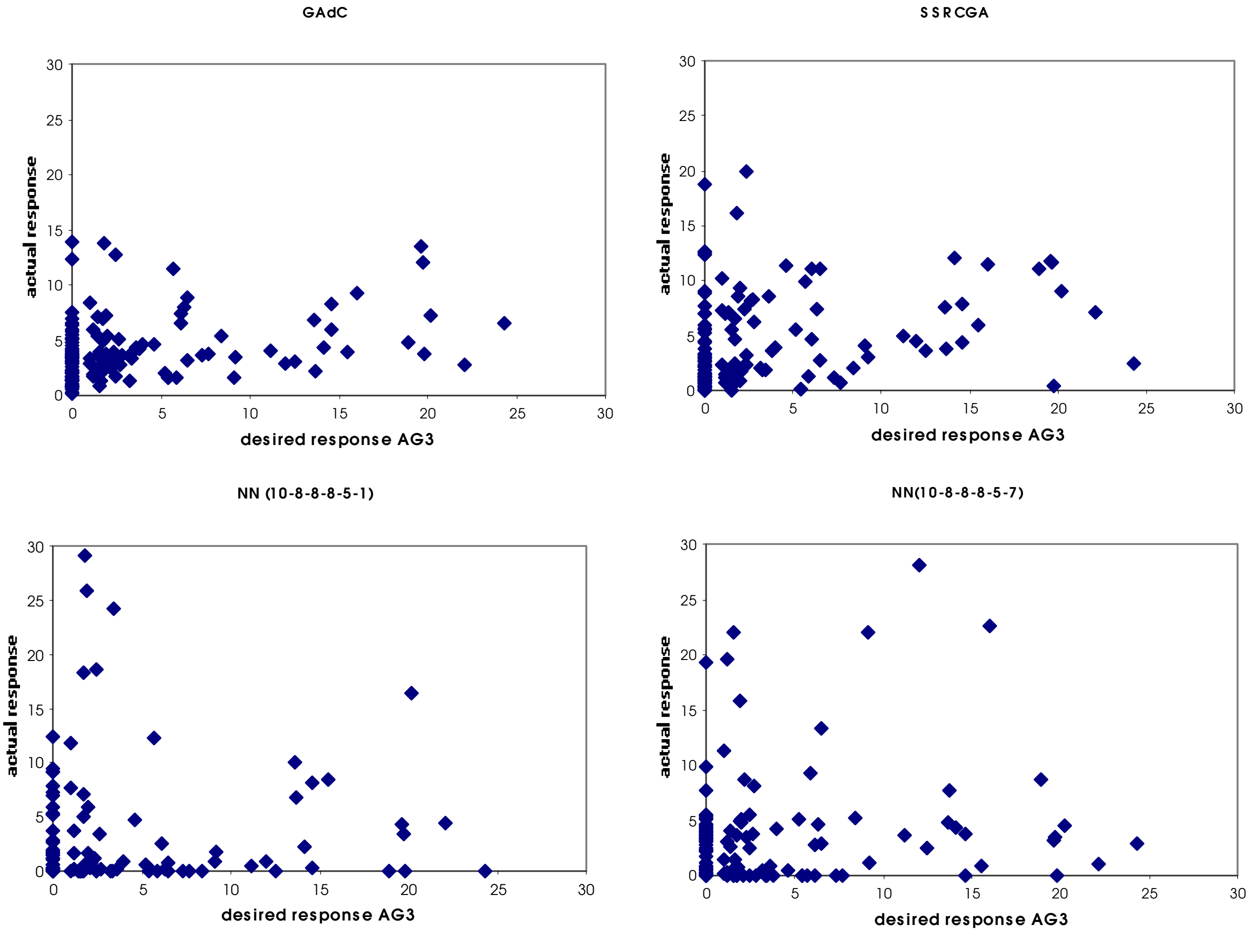

Fig. 1 The desired response AG3 versus the actual response AG3 for each method

CONCLUSIONS

This paper introduces the SSRCGA algorithm by gradually building on the idea of clustering with genetic algorithms and shows that SSRCGA is a viable alternative to traditional ANNs with both advantages and disadvantages. Preliminary computational studies show that the SSRCGA methodology can compare favorably with ANNs in regard to their forecasting performance.

ACKNOWLEDGEMENT

Part of this research was supported by the National Science Foundation under grant number IIS-9979860.

REFERENCES

[1] Demiriz, A. , Bennett, K. P., and Embrechts, M. J. (1999*). Semi-Supervised Scaled Clustering using Genetic Algorithms*. In Intelligent Engineering Systems through Artificial Neural Networks, Vol. 9, Cihan H. Dagli et. al., Eds., pp. 809 – 814, ASME Press, New York.

[2] Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.*D.* thesis, Harvard University. Reprinted in Werbos (1994).

[3] ftp.mitgmbh.de/pub/problem.zip

[4] Atkeson, C.G., Moore, A.W., and Schaal, S. (1997). *Locally Weighted Learning*. Artificial Intelligence Review, Vol. 11, pp 11 – 73.

[5] Cleveland, W. (1979). *Robust locally weighted regression and smoothing scatterplots*, Journal of the American Statistical Association, Vol. 74, pp. 829 - 836

[6] Krishnaiah, P. R., and Kanal, L. N., Eds. (1982). *Classification, Pattern Recognition, and Reduction of Dimensionality,* Vol. 2 of handbook of Statistics. North-Holland, Amsterdam.

[7] Kuncheva, L. I. and Bezdek, J.C. (1998). *Nearest Prototype Classification: Clustering, Genetic Algorithms or Random Search?* IEEE Transactions on Systems, Man, and Cybernetics C28, pp. 160-164.

[8] Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs (3^{rd}. Ed).* Springer-Verlag, Berlin.

[9] Devogelaere, D., Rijckaert, M., and Embrechts, M. J. (1999). *Third International Competition: Protecting Rivers and Streams by Monitoring Chemical Concentrations and Algae Communities Solved with the use of GAdC.* Proceedings of the 1999 European Conference on Intelligent Techniques and Soft Computing (EUFIT), September 7 – 10, Aachen, Germany.