Scalars, A Way to Improve the Multi-Objective Prediction of the GAdC-Method

Dirk Devogelaere, Marcel Rijckaert
K.U.Leuven, Chemical Engineering Department
De Croylaan 46, B-3001 Leuven, Belgium
Email: {dirk.devogelaere}@cit.kuleuven.ac.be

Abstract

This paper describes a hybrid method for supervised training of multivariate regression systems. The proposed methodology relies on supervised clustering with genetic algorithms and local learning. Genetic Algorithm driven Clustering (GAdC) offers certain advantages related to robustness, generalization performance, feature selection, explanative behavior and the additional flexibility of defining the error function and the regularization constraints. In this contribution we present the use of GAdC for prediction of algae distributions. We highlight one of the advantages of this method namely, the use of scalars to obtain the sequence in which the prediction of algae distributions should be calculated. Using this sequence leads to an improvement of the prediction.

1. Introduction

This paper describes a model for a regression analysis tool that can be seen as a kind of multi-strategy data analysis system. GAdC 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). GAdC 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 explains the GAdC methodology by discussing in succession: (i) local learning, (ii) clustering with GAs for a variable number of clusters and (iii) genetic algorithm driven 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 GAdC method is very helpful to improve the quality of the models needed to make the predictions. Indeed based on the scalars a good sequence of calculations will be derived.

2. GAdC methodology

This section presents the GAdC algorithm developed by the authors and designed for regression analysis.

2.1. Local learning

Local learning [2] 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 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

1522-4899/00 $10.00 © 2000 IEEE

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:

$$\hat{o}_i = \sum_{l=1}^{L}\left(\left\|\vec{x}_i - \vec{x}_l\right\|^d t_l \Big/ \sum_{l=1}^{L}\left\|\vec{x}_i - \vec{x}_l\right\|^d\right)$$

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 square error measure, the total regression error becomes

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

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

2.2. 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 [3]. However, there are a number of good reasons to consider other clustering methods as well [4].

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 uniform cross-over and uniform mutation [5]. A chromosome of the GA represents the coordinates of all cluster centers. If the dimensionality of the data is D (here the number of descriptors), and there are K cluster centers, there will be D*K genes. While the choice 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 selection and reproduction schemes. The principal behind genetic algorithms is essentially Darwinian natural selection. Selection provides the driving force in a genetic algorithm, and the selection pressure is critical in it. The selection directs a genetic algorithm search toward promising regions in the search space. There are three basic issues involved in selection phase: sampling space, sampling mechanism, selection probability. In our implementation we used an enlarged sampling space, both individuals from the old population (size μ) and offspring (size λ) have the same chance of competing for survival. This strategy was originally used in evolution strategies [6]. With this strategy, μ individuals and λ offspring compete for survival and the μ best out of offspring and individuals of the old population are selected as individuals for the new generation. The amount of overlap defines the amount of individuals from the population that will be used as parents. Note that not all individuals are used as parents during each generation step. The sampling mechanism used to select the parents is called roulette wheel selection. The basic idea is to determine selection probability for each chromosome proportional to the fitness value. To prevent premature convergence, the fitness values as calculated by means of the fitness function is scaled before the selection probability is calculated. In our implementation a linear scaling mechanism was used. Linear scaling adjusts the fitness values of all chromosomes such that the best chromosomes get a fixed number of expected offspring and thus prevent it from reproducing too many.

Note that so far the number of clusters was predetermined. It is now possible to extend GA driven clustering to allow for a varying number of clusters [4]. 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.

2.3. Genetic Algorithm-driven 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 last 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 is now an alternative to a traditional artificial neural network. 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 genes corresponding to the dimensionality (D) of the chromosomes. 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 further generalize this feature selection scheme assigning a different set of scaling factors to each cluster. An additional term was added in our implementation for practical reason. This term (1-cluster penalty) was to overcome the problem during the training phase that a cluster would contain only one element for which it would not be possible to calculate a target value as described before.

3. Computational results

3.1. Problem statement

Recent years have been characterized by increasing concern at the impact man is having on the environment. The impact on the environment of toxic waste, from a wide variety of manufacturing processes, is well known. More recently, however, it has become clear that the more subtle effects of nutrient level and chemical balance changes arising from farming land run-off and sewage water treatment also have a serious, but indirect, effect on the states of rivers, lakes and even the sea. In temperate climates across the world summers are characterized by numerous reports excessive summer algae growth resulting in poor water clarity, mass deaths of river fish from reduced oxygen levels and the closure of recreational water facilities on account of the toxic effects of this annual algae bloom. Reducing the impact of these man-made changes in river nutrient levels has stimulated much biological research with the aim of identifying the crucial chemical control variables for the biological processes.

The data used in this problem comes from one such a study [7]. During the research study water quality samples were taken from sites on different European rivers of a period of approximately one-year. These samples were analyzed for various chemical substances including: nitrogen in the form of nitrates, nitrites and ammonia, phosphate, pH, oxygen, chloride. In parallel, algae samples were collected to determine the algae population distributions. It is well know that the dynamics of the algae community is determined by external chemical environment with one or more factors being predominant. 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 task is the prediction of alga frequency distributions (fig. 1, AG1) 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 flow velocity. The two last variables are given as qualitative variables.

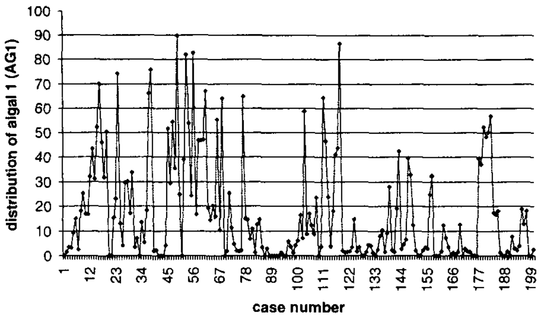

Fig 1: Algal distribution versus case number

3.2. Solution

The river pollution data set [7] 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 (samples) was used as blind set. There were 18 descriptors (variables) The first descriptor (season) was not used in this study. The remaining 17 descriptors were 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 (pH) or were calculated using the GAdC methodology.

Seven different models were build (all based on 10 descriptors, see results under sequence 'none' in table 1), one for each outcome (outcomes calculated independently of former calculated outcomes). The GAdC method started out with 17 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 800 and the bonus factor for empty clusters was set to 1 (penalty factor to discourage one element clusters was set to 5). The population size was set to 100 and the GA ran for 250 generations. The mutation and crossover probabilities were 0.01 and 0.6 respectively. A simple roulette selection procedure was followed for reproduction. The mean squared error is the value calculated for all the algae.

From biological point of view we do know that algae interact. Based on this fact we proposed to use previous calculated algae distributions in the prediction of next calculations (when we know the values for AG1, we will use this as an extra input to calculate the values for AG2). This involves that once the distribution of one algal is known, we are going to use this information in the calculations of the other algae. As prove of our assumption we calculate all values again in the sequence 1234567 (see table 1). The global error drops down significantly (11%)! As we have to calculate the outcome of 7 algae, this indicates that there exist 5040 different sequences.

Table 1: Results

Sequence of calculation	Mean squared error
None	95.3
1234567	84.7
1526743	90.0
1256743	87.3
1265743	85.1
1265734	84.2
1265347	84.4
1265437	84.8

For each outcome (AGi, i = 1 ... 7) we make a model based on the former used inputs and the values of the other six algae. Now we detect how important the other algae are in the prediction of this particular algae by calculating the product (importance indicator) between the scalar of the algal and the corresponding mean value. In this way we are able to rank the other algae for a specific algal. The results are presented in table 2. Based on this table we suggest to try out following sequences: 1526743; 1256743; 1265743; 12654734; 1265347 and 1265437. The obtained results are represented in table 1. The global mean squared error decreased 12%.

Table 2: Importance of other algae

AG1	AG2	AG3	AG4
1.40 AG6	3.50 AG1	3.06 AG1	1.53 AG1
1.39 AG5	1.71 AG5	2.20 AG6	1.36 AG2
0.80 AG7	1.32 AG6	1.08 AG5	1.25 AG5
0.72 AG3	0.75 AG3	0.39 AG4	1.05 AG3
0.59 AG2	0.66 AG4	0.32 AG7	0.73 AG6
0.36 AG4	0.45 AG7	0.30 AG2	0.55 AG7
AG5	**AG6**	**AG7**	
3.37 AG1	1.86 AG1	3.88 AG1	
1.92 AG6	1.82 AG5	1.27 AG2	
1.85 AG2	1.39 AG3	1.19 AG5	
1.14 AG3	1.38 AG2	0.82 AG6	
0.58 AG7	0.46 AG4	0.56 AG4	
0.50 AG4	0.33 AG7	0.28 AG3	

4. Conclusions

We presented a hybrid method for supervised training of multivariate regression systems. In general this method deals with ill-defined problems having little cases available each with a high number of variables. One of the main advantages is that the preprocessing phase can be very short (no scaling of values!). Furthermore it is easy to obtain a ranking of the variables depending on their importance for the prediction obtained. We applied our hybrid method to a real world environmental problem. In this problem we have little cases available with a moderate number of variables. Here we showed that the scalars are also useful to tell us between which outputs (algae) exist a "correlation". Using this information leads to the building of better models. By determining the influences between the algae, based on their importance to predict the outcome of one of them, it is possible to improve the prediction accuracy.

References

[1] D. Devogelaere, P. Van Bael, M. Rijckaert (1999) Regression through genetic algorithm driven clustering, *Proceedings of EUFIT99*, Aachen, Germany.

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

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

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

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

[6] H. Schwefel (1994). *Evolution and Optimum Seeking*, New York, John Wiley & Sons.

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