Regression through genetic algorithm driven clustering

Dirk Devogelaere, Patrick Van Bael and Marcel Rijckaert
K.U.Leuven – Chemical Engineering Department
De Croylaan 46, 3001 Heverlee Belgium
Phone: +32-16-322707, Fax: +32-16-322991
email:{dirk.devogelaere,patrick.vanbael,marcel.rijckaert}@cit.kuleuven.ac.be

ABSTRACT: Data mining is the search for valuable information in large volumes of data. This paper presents a genetic algorithm driven clustering (GAdC) to predict the value of targets based on known inputs. By the means of scalars the importance of the different features towards the target value can be detected. We first apply this approach to a classical classification problem, demonstrating the effectiveness of this method. Next we apply it to a real world problem, predicting the outlet concentration of a distillation column. In further ongoing research, refining of the genetic algorithm (GA) will be applied, including the application of huge data sets.

KEYWORDS: genetic algorithms, data mining, prediction, clustering, classification, feature extraction.

1 INTRODUCTION

Although data mining is an emerging field, it draws many of its basic principles from mature concepts in databases, machine learning and statistics [Fayyad 1996]. Two general categories of data-mining problems are prediction (classification, regression, time series) and knowledge discovery (deviation detection, database segmentation, clustering, association rules, ...).[Weiss 1998] The latter usually describes a stage prior to prediction, where information is insufficient for prediction. Prediction problems are described in terms of specific goals, which are related to past records with known answers. These are used to project to new cases.
The data mining approach we have developed is called the GAdC-regression, a genetic algorithm combined with a cluster algorithm to predict continuous features. Most regression problems tackled use neural networks to solve this prediction problem. One of the drawbacks of the neural networks is the difficulty to understand which features are important to create the desired outcome. By the use of scalars the GAdC-regression method gives a clear understanding of the importance of the different features. As in a neural network a training set is used to create clusters. New cases will be designated to one of the clusters and the desired target will be calculated depending on the “d” closest neighbors in this cluster.
Genetic algorithms have been used to solve difficult problems with objective functions that do not possess “nice” properties such as continuity, differentiability, etc. [Davis 1991; Goldberg 1989; Holland 1975; Michalewicz 1996]. These algorithms maintain and manipulate a family, or population, of solutions and implement a “survival of the fittest” strategy in their search for better solutions.
In the following sections we introduce the basic concept of the GAdC approach, then apply this approach to a classification problem (IRIS) and to a real word problem (distillation tower).

2 BASIC CONCEPT OF THE APPROACH

This section presents our Genetic Algorithm driven Clustering (GAdC) algorithm designed for the task of regression. This algorithm consists of two parts: the training part where the clustercentra and scalars are determined and a test part where the model is evaluated towards unknown test data. In the training part we use a steady state GA as described by De Jong. It uses overlapping populations with a user-specifiable amount of overlap. Fundamentally it contains all the parts of a genetic algorithm. So it requires the determination of six fundamental issues: chromosome representation, selection function, the genetic operators making up the reproduction function, the creation of the initial population, termination criteria, and the evaluation function. A standard GA library will implement most of those issues. A more complete discussion of genetic algorithms, including extensions and related topics, can be found in the books by Davis [Davis 1991], Goldberg [Goldberg 1989], Holland [Holland 1975] and Michalewicz [Michalewicz 1996]. The next subsections discuss several aspects of the design of GAdC.

2.1 CHROMOSOME REPRESENTATION

For any genetic algorithm (GA), a chromosome representation is needed to describe each individual in the population of interest. The representation scheme determines how the problem is structured in the GA and also determines the genetic operators that can be used. Each individual or chromosome is made up of a sequence of genes from a specified alphabet. In the GAdC it consists of floating point numbers with values within the variables upper and lower bounds. The variables are the clustercentra (one for each feature in each cluster) and the scalars (one for each feature).
In the design of the GAC-regression, determination of scalars and clustercentra are crucial to optimising performance, and strongly affect regression results. Ideally, the problem of determining the scalars on the one hand and choosing the right clustercentra on the other should not be considered independently. Researchers [Siedlecki 1988; Jain 1991] dealing with the classification approach made similar considerations. The ultimate goal is good regression and the intermediate determination of scalars and possible dimension reduction is, in a sense, subservient to that goal, rather than being an end in itself.

2.2 SELECTION FUNCTION

The selection of individuals to produce successive generations plays an important role in a genetic algorithm. A probabilistic selection is performed based upon the individual's fitness such that the better individuals have an increased chance of being selected. There are several schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, elitist models, and ranking methods [Goldberg 1989; Michalewicz 1996]. Here we use linear scaling in combination with roulette wheel selection [Holland 1975].

2.3 GENETIC OPERATORS

Genetic operators provide the basic search mechanism of the GA. The operators are used to create new solutions based on existing solutions in the population. There are two basic types of operators: crossover and mutation. Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution. Michalewicz [Michalewicz 1996] developed operators for real-values representations, i.e., an alphabet of floats. In GAdC uniform crossover and flip mutation are applied.

2.4 INITIALIZATION AND TERMINATION

The GA must be provided with an initial population. The most common method is to randomly generate solutions for the entire population, as we do in GAdC.
The GA moves from generation to generation selecting and reproducing parents until a termination criterion is met. Each generation the algorithm creates a temporary population of individuals, adds these to the previous population, then removes the worst individuals in order to return the population to its original size. The most frequently used stopping criterion is a specified maximum number of generations as we use in GAdC.

2.5 EVALUATION FUNCTION

The evaluation function is the driving force behind the GA. The evaluation function is called from the GA to determine the fitness of each individual generated during the search. The evaluation function consists of the sum of four terms: 1) a cluster distance penalty factor, 2) a misclassification penalty, 3) an empty_cluster penalty, 4) an 1_element_cluster penalty. These terms are elaborated in the following.
The training data consist of a set X of training vectors (or cases) Xi each of size N (N indicating the number of features being used), whereby $Xi = [x_{i1}, x_{i2}, \dots , x_{iN}]$. The scalars are represented by a vector S, whereby $S = [s_1, s_2, \dots, s_N]$. The cluster data consist of a set C of clustercentra Ck each of size N, whereby $Ck = [c_{k1}, c_{k2}, \dots , c_{kN}]$. Further more k is the maximal number of clusters, n is the number of cases, w_i is an user-defined factor, Y is a vector containing the real target values and T is a vector containing the calculated target values.

The cluster distance penalty factor:

This term is one of the critical elements of the GAdC. Here the Euclidean distance is calculated between the clustercentra Ck and the cases Xi, which are assigned to them. The GA will try to minimize this distance through

choosing the right value for the clustercentra Ck and the scalars S. It is an unsupervised clustering of the training data. We use the term:

$$w_1 * \sum_k \sum_i \sqrt{\sum_j s_j * (x_{ij} - c_{jk})^2} \quad (1)$$

The misclassification penalty:

Adding this term allows us to address the search towards regions where we obtain good prediction values (in T) for the real target values (in Y). In fact we calculate here the error. This term is defined as

$$w_2 * \sqrt{\sum_{i=1}^{n} (y_i - t_i)^2} \quad (2)$$

The prediction values (in T) are calculated as a weighted (means based on the distance) sum of the d nearest neighbors in the cluster to which the target has been assigned.

The empty_cluster penalty:

This term forces the algorithm to evolve towards a maximum of empty clusters. This term is defined as

$$w_3 * \text{counter1} \quad (3)$$

where w_3 is a user-defined factor, and counter1 is the number of detected empty clusters. This user-defined factor is a negative integer number.

The 1_element_cluster penalty:

This term forces the algorithm to evolve towards a minimum of 1-element clusters. This term is defined as

$$w_4 * \text{counter2} \quad (4)$$

where w_4 is a user-defined factor, and counter2 is the number of detected 1-element clusters. This user-defined factor is a positive integer number.

3 PRELIMINARY COMPUTATIONAL RESULTS AND DISCUSSION

We have done a preliminary, “proof-of-concept” experiment to evaluate the GAdC proposed in this paper. (We intend to do more extensive experiments in the near future). The data sets applied were a typical classification problem: IRIS, and a limited data set of a real world problem: a distillation tower.

3.1 THE IRIS DATA SET

This data set [Fisher 1936] consists of 150 cases and 4 features. There are three classes: setosa, versicolor and virginia which are represented by the numbers 1, 2 and 3. We divide the set into training set (100 cases) and test set (50 cases). The data mining task consisted of predicting the value of the class (1, 2 or 3) to which they belong. As mentioned before we use a steady state GA with the next settings: number of generations (200), population size (40), crossover rate (0.60), permutation rate (0.01) and replacement percentage (0.5). The results are listed in table I.

	N_runs	Train error	Test error	Scalar 1	Scalar 2	Scalar 3	Scalar 4
Vers. 1	30	1.4	3.8	0.16	0.49	0.26	0.75
Vers. 2	10	1.1	3.0	0.18	0.43	0.40	0.75
Vers. 3	10	1.4	1.9	0.26	0.41	0.44	0.70

Table I:. The comparison of results of different variable settings

In vers. 1 the misclassification penalty settings were d = 3 and w_2 = 500. In vers. 2 we altered the misclassification penalty (equation 2)as follows:

$$w_2 * \sum_i^n R_i \quad \text{and Ri = 1 if |yi – ti| > 0.01 else Ri = 0} \qquad (5)$$

The value of d = 5, this means that ti is calculated based on the five nearest neighbors. In vers. 3 we set w_2 = 800, other settings were the same as in vers. 2.
Just by making some minor changes, the results of the test error improved from 3.8 to 1.9. The scalars give an indication of the importance of the different features. Feature 4 is the most important and feature 1 the least one.

3.2 DISTILLATION COLUMN PROBLEM

In this real world problem form the chemical process industry, the task was to predict the recovery of the light component in the bottom product of a bicomponent distillation column. The plant is characterized by twelve parameters, of which eleven were used as input parameters (features) and one, the recovery of the light component in the bottom product, is applied as the desired output parameter (target value). Historical data was extracted out of the plant's database.
In the case of the neural networks a representative part of the data file (6000 patterns) was used to train the networks, while the remaining part (3000 patterns) was applied as test set. The recurrent networks were trained for up to 500 epochs with various combinations of parameters to obtain the optimal size and parameters for each network. [Meert 1997].
In the case of GAdC we selected random 400 patterns for training and 300 patterns for testing out of the 6000 patterns used in the neural networks for training.

Type of method used	Training		Testing	
	Optimum	Average	Optimum	Average
QPTT recurrent network	0.0224	0.0250 (0.0024)	0.0369	0.0407 (0.0011)
GADC	**0.0371**	**0.0399 (0.0016)**	**0.0413**	**0.0448 (0.0019)**
Elman recurrent network	0.0339	0.0389 (0.0025)	0.0454	0.0606 (0.0050)
Jordan recurrent network	0.0387	0.0406 (0.0011)	0.0539	0.0667 (0.0038)
MLRN recurrent network	0.0438	0.0448 (0.0007)	0.0628	0.0699 (0.0071)
RTRL recurrent network	0.0592	0.0594 (0.0001)	0.0856	0.0886 (0.0041)
BBPTT recurrent network	0.0530	0.0628 (0.0010)	0.0532	0.1287 (0.0058)
BPTT recurrent network	0.0359	0.1496 (0.0566)	0.0380	0.2543 (0.1070)

Table II:. The RMS errors for various methods used for the distillation column problem (the standard deviation is indicated in brackets).

From Table II it is evident that the QPTT network (12 nodes) outperformed the Elman, GAdC, Jordan, MLRN, RTRL, BBPTT and BPTT on training average by respectively 36%, 37%, 38%, 44%, 58%,60% and 83%. Similar conclusions can be derived when comparing the test averages. Here the GAdC performed better than the Elman and is only slightly different from the QPTT.

4. CONCLUDING REMARKS

In this article an algorithm for regression through genetic algorithm driven clustering (GAdC) was presented and tested on the IRIS benchmark and a chemical engineering problem, a bicomponent distillation column. The IRIS benchmark points out the possibilities of the scalars and the possibility to use the algorithm for classification. Further improvements are expected by optimising the settings and improving the used GA and cluster algorithm. Furthermore the results on the distillation column show that this method is an alternative for neural networks. Thorough investigation should point out the advantages of this new method in comparison to neural networks.

ACKNOWLEDGEMENT

This research was performed at the Expert Systems Application Development Group which is headed by Prof. M. Rijckaert.

REFERENCES

Davis, L., 1991, The Handbook of Genetic Algorithms, Van Nostrand Reingold, New York.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., 1996, From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, p1-34.

Fisher, R.A., Annals of Eugenics, 7 (1936) 179.

Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley.

Holland, 1975, J., Adaption in natural and artificial systems, The University of Michigan Press, Ann Arbor

Jain, A. K., 1991, Artificial Neural Networks and Statistical Pattern Recognition. Editor preface, Elsevier Science Publishers B.V.

Meert, K., Ludik, J., A Multilayer Real-Time Recurrent Learning Algorithm for Improved Convergence, Proceedings of International Conference on Artificial Neural Networks, 1997, 445-450.

Michalewicz, Z., 1996, Genetic Algorithms + Data Structures = Evolution Programs, Springer Berlin.

Siedlecki, W., Sklansky, J., On Automatic Feature Selection Internat. Journal of Pattern Recognition and Artificial Intelligence. Vol 2, No.2 1988, 197-220.

Weiss, S.M., Indurkhya, N., 1998, Predictive Data Mining, A Practical Guide, Morgan Kaufmann Publishers, Inc.