RCGA-S/RCGA-SP Methods to Minimize the Delta Test for Regression Tasks

Fernando Mateo[1], Dušan Sovilj[2], Rafael Gadea[1], and Amaury Lendasse[2]

[1] Institute of Applications of Information Technologies and Advanced Communications, Polytechnic University of Valencia, Spain
fermaji@upvnet.upv.es,rgadea@eln.upv.es
[2] Laboratory of Information and Computer Science, Helsinki University of Technology, Finland
dusans@cis.hut.fi,lendasse@hut.fi

Abstract. Frequently, the number of input variables (features) involved in a problem becomes too large to be easily handled by conventional machine-learning models. This paper introduces a combined strategy that uses a real-coded genetic algorithm to find the optimal scaling (RCGA-S) or scaling + projection (RCGA-SP) factors that minimize the Delta Test criterion for variable selection when being applied to the input variables. These two methods are evaluated on five different regression datasets and their results are compared. The results confirm the goodness of both methods although RCGA-SP performs clearly better than RCGA-S because it adds the possibility of projecting the input variables onto a lower dimensional space.

Key words: real-coded genetic algorithm, global search, variable selection, delta test, input scaling, input projection

1 Introduction

The size of datasets often compromises the models that can employ them for a determined regression or classification task. A linear increase in the number of variables results in an exponential increase in the necessary number of samples to successfully represent the solution space. This burden is called curse of dimensionality [1] and affects many real-life problems usually characterized by a high number of features. In these cases, it is highly convenient to reduce the number of involved features in order to reduce the complexity of the required models and to improve the interpretability.

In the recent years, many studies have intended to address variable selection for regression using a variety of search strategies and convergence criteria. One of the most successful criteria to determine the optimal set of variables in regression applications is a nonparametric noise estimator called Delta Test (DT) ([2], [3]).

With regard to the search strategy, some authors propose local search strategies for DT minimization (e.g. forward search [4], backward search, forward-backward search ([5], [6])), because of their high speed, but they suffer from severe sensitivity to local minima. Global search strategies (e.g. exhaustive search,

tabu search [7], genetic algorithms (GA) [8]) explore more efficiently the solution space but are much slower too. A tabu based approach to DT minimization has been reported in [6] and [9]. Parallel schemes that combine tabu and GAs have also been implemented to ease the slow convergence drawback [6].

This paper aims to optimize the choice of relevant inputs in an automated manner, by introducing a combination of a GA-based global search strategy with two different fitness approaches: scaling and scaling enhanced with projection. The use of real-valued scaling factors is already a great improvement that minimizes the DT beyond the limit imposed by pure selection, because a variable can not only be selected or not, but also be given a weight according to its relative importance. Projection takes a further step as it includes the possibility of projecting the input vectors into a lower dimensional space. Both methods have been compared in [4] using a forward search method but their integration in a global search framework remains unexplored so far.

This paper is organized as follows: Section 2 introduces the DT and its theoretical background. Section 3 describes the developed genetic algorithm and its main parameters, paying special attention to the two custom fitness functions created. Section 4 presents a performance study of both methods on a variety of datasets and discusses the results. Finally, Section 5 summarizes the conclusions.

2 The Delta Test

The DT, firstly introduced by Pi and Peterson for time series [2] and proposed for variable selection in [10], is a technique to estimate the variance of the noise, or the mean squared error (MSE), that can be achieved without overfitting. Given N input-output pairs $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, the relationship between $\boldsymbol{x}_i$ and y_i can be expressed as

$$y_i = f(\boldsymbol{x}_i) + r_i, \qquad i = 1, ..., N \ , \tag{1}$$

where f is the unknown function and r is the noise. The DT estimates the variance of the noise r.

The DT is useful for evaluating the nonlinear correlation between input and output variables. It can also be applied to input variable selection: the set of input variables that minimizes the DT is the selected one. The DT is based on hypotheses coming from the continuity of the regression function. If two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close in the input variable space, the continuity of regression function implies the outputs $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ will be close enough in the output space. If this is not accomplished, it is due to the influence of the noise.

The DT can be interpreted as a particularization of the Gamma Test (GT) [11] considering only the first nearest neighbor. This yields a fully nonparametric method as it removes the only hyperparameter (number of neighbors) that had to be chosen for the GT. Let us denote the nearest neighbor of a point $\boldsymbol{x}_i \in \mathbb{R}^d$ as $\boldsymbol{x}_{NN(i)}$. The nearest neighbor formulation of the DT estimates $\mathrm{Var}[r]$ by

$$\mathrm{Var}[r] \approx \delta = \frac{1}{2N} \sum_{i=1}^{N} (y_i - y_{NN(i)})^2 \ , \tag{2}$$

where $y_{NN(i)}$ is the output of $\boldsymbol{x}_{NN(i)}$. For a proof of convergence the reader should refer to [11].

3 Real-coded genetic algorithms for global search

The use of GAs for variable selection has been widely reported in the literature ([12], [13], [14], [15], [16]). The purpose of the GA in this work is the global optimization of the scaling weights and projection matrix that minimize the DT when applied to the input vectors. This study intends to find the optimal DT value in a fixed number of generations. Pure selection would clearly outperform scaling in terms of speed but the best DT found is often sub-optimal. Scaling or projection are necessary to get closer to the optimal set of solutions. For that reason, a real-coded GA (RCGA) is proposed to optimize a population of chromosomes that encode arrays of potential solutions. The two following Subsections describe the fitness functions that were built and applied to the RCGA: one for scaling and another combining scaling and projection.

3.1 Real-coded genetic algorithm with scaling: RCGA-S

The target of performing scaling is to optimize the value of the DT beyond the minimum value that can be obtained with pure selection. When performing scaling, the selected variables are weighted according to their influence on the output variable. Let us consider f as the unknown function that determines the relationship between the N input-output pairs of a regression problem, $y = f(\boldsymbol{x}) + r$, with $\boldsymbol{x} \in \mathbb{R}^d, y \in \mathbb{R}$ and $r \in \mathbb{R}$ is a random variable that represents the noise. Thus, the estimate of the output, $\hat{y} \in \mathbb{R}$, can be expressed as $\hat{y} = g(\boldsymbol{x_s}) + r$, with $\boldsymbol{x_s} = \boldsymbol{s} \cdot \boldsymbol{x} \in \mathbb{R}^d$ and g is the model that best approximates the function f. The objective is to find a scaling vector $\boldsymbol{s} \in \mathbb{R}^d$ such that

$$\hat{y} = g(s_1 x_1, s_2 x_2, \ldots, s_d x_d) + r \tag{3}$$

minimizes Var$[r]$ for the given problem.

In the existing variable selection literature there are several applications of scaling to minimize the DT, but often keeping a discrete number of weights ([4], [5], [6]) instead of using unconstrained real values like in this study. In each generation, each individual (array of scaling factors) is multiplied element by element by the i-th input sample from the dataset:

$$X^i_{S(1\times d)} = X^i_{(1\times d)} \times S_{(1\times d)}, \qquad i = 1, \ldots, N\ , \tag{4}$$

where X is the $N \times d$ input matrix, X_S is the scaled version of X and S is the scaling vector.

The DT is calculated by obtaining the Euclidean distances among the weighted input samples X_S. Once done this, the first nearest neighbor of each point is selected and the DT is obtained from their corresponding outputs, according to Eq. 2. When a predefined number of generations has been evaluated, the GA returns the fittest individual and its corresponding DT.

3.2 Real-coded genetic algorithm with scaling + projection: RCGA-SP

A projection can be used to reduce the number of variables by applying a linear (idempotent) transformation, represented by a matrix $P_{(d\times k)}$, to the matrix of input samples $X_{(N\times d)}$, resulting in a lower dimensional matrix $X_{P(N\times k)}, k < d$:

$$X_{P(N\times k)} = X_{(N\times d)} \times P_{(d\times k)} \ . \tag{5}$$

Although it might seem counterproductive, the idea if the developed method that combines scaling and projection is to add a new variable to the input space (the projection of the input vectors on one dimension, i.e. with $k = 1$). Equations 6 and 7 describe this approach:

$$X_{P(N\times 1)} = X_{(N\times d)} \times P_{(d\times 1)} \ , \tag{6}$$

$$X_{SP(N\times(d+1))} = [X_{S(N\times d)}, X_{P(N\times 1)}] \ , \tag{7}$$

where X_S is the scaled version of X as calculated in equation 4, X_P is the projected version of X and X_{SP} is the new scaled/projected input matrix. In this case, the length of the chromosome will be twice the length of the ones used for the scaling approach, i.e. $2d$, as a global optimization of the projection vector P must be carried out along with the optimization of the scaling vector S.

4 Experiments

The experiments were carried out using MATLAB 7.5 (R2007b, The Mathworks Inc., Natick, MA, USA), partly using the Genetic Algorithm and Direct Search Toolbox v2.2, and several custom functions. The parts of the code that are critical for speed, like the computation of pairwise distances among points, were coded in C++ and compiled as MATLAB executables (MEX).

The populations are initially created using a custom function that assigns a uniform initialization to a percentage of the population and the rest can be customized by the user, specifying how many of the remaining individuals are initialized randomly and how many of them are left as zeros. The function is flexible in the sense that the custom percentage of the initial population can be further split into more subsets, each one with a customizable percentage of randomly initialized individuals.

The crossover and mutation operators have also been implemented as custom functions. The mutation operator is a pure random uniform function whereas the crossover operator was BLX-α [17] because of its better performance compared to the one-point, two-point and uniform crossover operators [8]. Regarding the selection operator, the binary tournament was chosen because of its better performance and speed than the roulette wheel. Three population size values were tested: 50, 100 and 150. Values higher than 150 were discarded in order to keep reasonable run times. To sum up, the GA parameters were set as follows:

- Number of averaged runs: 10
- Number of generations evaluated: 50
- Population sizes: 50, 100, 150
- Population initialization: 20% uniform / 80% custom (with 90% zeros and 10% random genes)
- Crossover operator: BLX-α (α=0.5)
- Selection function: Binary tournament
- Crossover rate: 0.85
- Mutation rate: 0.1
- Elitism: 10%
- Mutation function: Random uniform
- Fitness function: S/SP

4.1 Datasets

The described methods (RCGA-S and RCGA-SP) have been evaluated on five regression datasets with different sample/variable ratios to assess their performance in different types of scenarios. The dimensionality and number of samples of each dataset are listed in Table 1. Santa Fe and ESTSP 2007 are time series, so regressors of 12 and 55 variables, respectively, were built.

Table 1. Datasets used in the experiments.

Dataset	Instances	Input variables
Boston Housing[1]	506	13
Tecator[2]	215	100
Anthrokids [3]	1019	53
Santa Fe[4]	987	12
ESTSP 2007[3]	819	55

[1] http://archive.ics.uci.edu/ml/datasets/Housing
[2] http://lib.stat.cmu.edu/datasets/tecator
[3] http://www.cis.hut.fi/projects/tsp/index.php?page=timeseries
[4] http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html

All datasets were normalized to zero mean and unit variance, to prevent variables with high variance from dominating over those with low variance. Therefore, all DT values shown in this paper are normalized by the variance of the output. The normalization was done variable-wise for all datasets except for Tecator, in which variable selection works better with sample-wise normalization.

4.2 Results

The results of the experiments appear listed in Table 2. In all tests, the population size of 50 chromosomes gave worse results than 100 or 150. The population

Table 2. Performance of RCGA-S and RCGA-SP after 50 generations.

Method	Pop. size	Measurement	Housing	Tecator	Anthrokids	Santa Fe	ESTSP
RCGA-S	50	Mean±StDev DT ($\times 10^{-4}$)	570±14	108±11	75±3	107±12	127.5±1.3
		Min DT ($\times 10^{-4}$)	544.54	92.60	71.91	89.99	125.01
		Max DT ($\times 10^{-4}$)	583.09	132.75	79.37	119.73	128.74
	100	Mean±StDev DT ($\times 10^{-4}$)	559±7	98±13	72.0±1.9	92±12	123.8±2.1
		Min DT ($\times 10^{-4}$)	544.78	75.74	69.13	77.33	120.65
		Max DT ($\times 10^{-4}$)	569.82	109.52	74.83	111.41	126.78
	150	Mean±StDev DT ($\times 10^{-4}$)	553±16	98±12	71.8±1.2	85±8	123.7±2.1
		Min DT ($\times 10^{-4}$)	528.47	83.60	69.81	75.46	121.43
		Max DT ($\times 10^{-4}$)	578.18	113.47	72.98	98.54	128.93
RCGA-SP	50	Mean±StDev DT ($\times 10^{-4}$)	570±60	39±3	71±5	83±15	125±5
		Min DT ($\times 10^{-4}$)	523.56	35.02	66.08	69.96	119.02
		Max DT ($\times 10^{-4}$)	692.22	43.53	77.49	111.66	132.3
	100	Mean±StDev DT ($\times 10^{-4}$)	537±22	38.1±2.4	69±4	72±8	123±4
		Min DT ($\times 10^{-4}$)	512.47	35.14	62.54	62.15	116.71
		Max DT ($\times 10^{-4}$)	583.14	43.36	75.33	82.85	128.97
	150	Mean±StDev DT ($\times 10^{-4}$)	530±17	36.8±2.1	69±4	68±5	122±4
		Min DT ($\times 10^{-4}$)	514.15	33.12	61.82	62.83	115.62
		Max DT ($\times 10^{-4}$)	557.88	40.74	73.71	77.86	129.18

size of 150 minimized the DT for most datasets, either with RCGA-S or RCGA-SP. Nonetheless, the values of DT obtained with 100 individuals are often very similar to the ones obtained with 150, and the high increase in computational time might not always be worthwhile.

The average, minimum and maximum DT values are improved in all cases by using RCGA-SP instead of RCGA-S. The rate of improvement depends on each particular dataset, and is specially noticeable for Tecator (>64%) or Santa Fe (>20%). Another important result is that the RCGA-S method is more precise than RCGA-SP as the standard deviation is generally lower. Predictably, the fact of doubling the chromosome size increases runtimes too.

An analysis of the initialization function was carried out using the GAs with the best specifications among the tested (Population size = 150). The results, for several custom/uniform initialization ratios, are listed in Tables 3 and 4. The best mean DT for each dataset is marked in bold. As before, 90% of the custom part was composed of zeros while the remaining 10% was randomized. The results confirm the goodness of the custom initialization with respect to the pure uniform, as the mean DT is reduced in most cases when a high rate of custom-initialized genes is used. Again, the best improvement is found for Tecator dataset (>68% in some cases).

5 Conclusions

The methodology presented is a combination of a real-coded GA with custom fitness functions that perform scaling (RCGA-S) and scaling + projection (RCGA-SP), which has proved to accurately minimize the DT in a variety of scenarios.

Table 3. Mean and standard deviation of DT values ($\times 10^{-4}$) calculated by RCGA-S for several initialization ratios (Population size = 150).

Custom/Uniform	Housing	Tecator	Anthrokids	Santa Fe	ESTSP
0%/100%	554±5	135.2±1.2	83±3	100±7	123.0±0.9
10%/90%	551±4	133.3±2.0	79±3	99±9	**121.7**±1.9
20%/80%	553±8	127±9	77±3	96±10	123.4±1.0
30%/70%	554±11	124.5±7	76±3	99±11	123.0±1.3
40%/60%	550±9	118±10	75±3	91±9	123.0±1.6
50%/50%	552±8	109±12	72.7±2.0	91±9	122.8±1.3
60%/40%	549±11	110±7	72.7±2.0	91±9	122.4±1.8
70%/30%	**548**±11	105±9	73.3±2.1	87±6	123.5±1.0
80%/20%	553±16	98±12	71.8±1.2	85±8	123.7±2.1
90%/10%	549±10	92±11	**70.9**±0.9	**80**±5	123.0±1.2
100%/0%	605±40	**87**±10	72.60±0.10	**80**±6	125.6±2.1

Table 4. Mean and standard deviation of DT values ($\times 10^{-4}$) calculated by RCGA-SP for several initialization ratios (Population size = 150).

Custom/Uniform	Housing	Tecator	Anthrokids	Santa Fe	ESTSP
0%/100%	543±19	42.2±1.4	72±3	77±12	**119**±3
10%/90%	537±18	40.8±1.6	78±8	83±13	120±3
20%/80%	535±16	41.3±1.9	82±11	76±10	119.8±1.7
30%/70%	539±18	40.0±2.5	86±14	85±21	120±3
40%/60%	531±15	40±4	80±8	76±10	119.2±1.8
50%/50%	541±22	38.7±1.9	80±7	67±4	120±3
60%/40%	536±13	38.9±2.1	76±8	71±8	122±3
70%/30%	540±21	40±3	77±8	68±5	**119**±3
80%/20%	**530**±17	36.8±2.1	69±4	68±5	122±4
90%/10%	539±23	36.2±2.0	66±6	68±5	123±3
100%/0%	555±21	**34.8**±1.6	**65**±6	**66**±3	124±4

In particular, the RCGA-SP method has proved to find lower values of DT than RCGA-S in all tests. Furthermore, the custom initialization proposed enables a refinement of the final value of DT obtained and performs better than the uniform initialization in most cases. The minimum DT values found are lower than the lowest values attained in previous works, either using local or global search strategies ([4], [6]) for the tested datasets. The main drawback of RCGA-SP is the computational time involved, but this issue could be alleviated in the future using parallel implementations. Scaling + projection to higher dimensional spaces ($k > 1$) is also to be further examined.

Acknowledgments Fernando Mateo received financial support from projects AGL 2004-07549-C05-02/ALI and FPA 2007-65013-C02-02 from the Spanish Ministry of Science and Innovation, and from a research grant.

References

1. Verleysen, M., François, D.: The curse of dimensionality in data mining and time series prediction. In Cabestany, J., Prieto, A., Hernandez, F., eds.: Lecture Notes in Computer Science. Volume 3512., Springer (2005) 758–770
2. Pi, H., Peterson, C.: Finding the embedding dimension and variable dependencies in time series. Neural Computation **6**(3) (1994) 509–520
3. Liitiäinen, E., Corona, F., Lendasse, A.: On nonparametric residual variance estimation. Neural Processing Letters (to be published) (2008)
4. Yu, Q., Séverin, E., Lendasse, A.: A global methodology for variable selection: application to financial modeling. In: Proc. of MASHS 2007, ENST-Bretagne, France (May 2007)
5. Mateo, F., Lendasse, A.: A variable selection approach based on the delta test for extreme learning machine models. In: Proc. of ESTSP 2008, European Symposium on Time Series Prediction, Porvoo, Finland (Sept. 2008) 57–66
6. Guillén, A., Sovilj, D., Mateo, F., Rojas, I., Lendasse, A.: Minimizing the delta test for variable selection in regression problems. Int. J. on High Performance Systems Architecture (2009) In Press.
7. Glover, F., Laguna, F.: Tabu Search. Kluwer Academic Publishers, Norwell, MA, USA (1997)
8. Holland, J.: Adaption in natural and artificial systems. University of Michigan Press (1975)
9. Sovilj, D., Sorjamaa, A., Miche, Y.: Tabu search with delta test for time series prediction using OP-KNN. In: Proc. of ESTSP 2008, European Symposium on Time Series Prediction, Porvoo, Finland (Sept. 2008) 187–196
10. Eirola, E., Liitiäinen, E., Lendasse, A., Corona, F., Verleysen, M.: Using the delta test for variable selection. In: Proc. of ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium (April 2008) 25–30
11. Jones, A.: New tools in non-linear modelling and prediction. Computational Management Science **1**(2) (Jul. 2004) 109–149
12. Oh, I.S., Lee, J.S., Moon, B.R.: Local search-embedded genetic algorithms for feature selection. Proc. of the 16th Int. Conference on Pattern Recognition **2** (2002) 148–151
13. Oh, I.S., Lee, J.S., Moon, B.R.: Hybrid genetic algorithms for feature selection. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(11) (Nov. 2004) 1424–1437
14. Punch, W., Goodman, E., Pei, M., Chia-Shun, L., Hovland, P., Enbody, R.: Further research on feature selection and classification using genetic algorithms. In Forrest, S., ed.: Proc. of the Fifth Int. Conf. on Genetic Algorithms, San Mateo, CA, Morgan Kaufmann (1993) 557–564
15. Raymer, M., Punch, W., Goodman, E., Kuhn, L., Jain, A.: Dimensionality reduction using genetic algorithms. IEEE Trans. on Evolutionary Computation **4**(2) (Jul. 2000) 164–171
16. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19) (2007) 2507–2517
17. Eshelman, L., Schaffer, J.: Real-coded genetic algorithms and interval schemata. In Darrell Whitley, L., ed.: Foundation of Genetic Algorithms 2, Morgan-Kauffman Publishers, Inc. (1993) 187–202