Linear Regression Model Selection Using A Simplex Reproduction Genetic Algorithm

Reto Grüter and Jean-Marc Vesin

Signal Processing Laboratory
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland
grueter@lts.de.epfl.ch

RÉSUMÉ

Dans cet article on propose une nouvelle méthode de sélection de modèles linéairement regressifs. Un algorithme génétique (AG) basé sur une petite population est introduit. Le codage simple des paramètres et la convergence rapide rend la complexité modeste. La première application montre comme des modèles autoregressifs et des modèles polynomiaux nonlinéaires peuvent être correctement selectionnés. La complexité est donnée en fonction du nombre de générations et du nombre d'évaluations nécessaires jusqu'à la convergence de l'algorithme. Ensuite, l'AG est appliqué en tant qu'algorithme d'entraînement pour des réseaux à fonctions radiales de base. La stratégie de sélection des centres est comparée à l'algorithme d'entraînement par moindres carrés. Pour des sous–ensembles à diffŕentes dimensions, l'AG fournit de meilleurs modèles que l'algorithme des moindres carrés.

ABSTRACT

In this paper a novel method for linear regression model selection is proposed. A reproduction genetic algorithm (GA) suited for small populations is introduced. The simple encoding of the parameters and fast convergence provide modest computational complexity. The first applications show how linear autoregressive models and nonlinear polynomial models are correctly selected. The complexity is given in terms of number of generations and number of necessary model evaluation until convergence. Furthermore, the GA is applied as a learning algorithm for radial basis function (RBF) networks. The center selection strategy is compared to the well–known orthogonal least squares learning (OLS) algorithm. For center subsets of various dimensions, the GA provides better models than the OLS algorithm.

1 Introduction

Various problems in signal processing consist in finding solutions to a linear regression model. Moreover, an adequate subset selection for the regressors is often a crucial factor for adequate prediction.

In this paper we show how a simple genetic algorithm (GA) using a simplex reproduction scheme can efficiently be applied to regression model selection. We investigate the number of necessary iterations to find the correct regressors of an autoregressive model. Then, we expand the model to nonlinear polynomials. The number of model evaluations is computed until the GA converges to the correct synthetical model. As a further example we apply the GA to the problem of selecting centers of radial basis functions (RBF). RBF networks are a particular class of neural networks where the parameters in the hidden called centers undergo a learning procedure. The GA presented here determines the dimension and the location of the subset of centers building the model. The method is compared to the orthogonal least squares (OLS) learning algorithm. For both methods the mean square error (MSE) provided by a selected model is computed and compared to the complexity of the model. Therefore, the minimum description length (MDL) criterion is applied that validates a model according to its coding cost.

2 The Genetic Algorithm

GA's have been developed to mimic some of the processes observed in natural evolution. They are based on the principle of the selection of the fittest constellations out of a basic ensemble. In other words, they can perform adequate subset selection for various optimization problems. Basically, a GA is applied to a set of chromosomes that constitute a population. During the process of reproduction, a population evolves from the current generation to the next generation.

In this study, we apply a three-operator GA consisting of reproduction, mutation and crossover. In [2], an approach for the successful use of small populations has been introduced. The algorithm presented here has been derived from the simplex reproduction algorithm in [6]. Here, a chromosome is a binary string where each position represent a specific regressor in a linear regression model. The binary value indicates the presence or the absence of this regressor for the considered model.

The important steps of the algorithm are summarized below :

- Initialization : Generate N chromosomes randomly where N is a small odd number.
- Evaluation of each chromosome fitness.
- Application of the operators :

- Reproduction : Select the chromosome providing best fitness. It goes unchanged into the next generation.
- Mutation : *(N-1)/2* mutations from the fittest are passed to the next generation. One bit randomly selected in the chromosome changes per mutation.
- Crossover : Each chromosome competes with its neighbor. The losers are discarded whereas the winners are crossed and placed in the next generation. The competitors always consist of a mutated and crossed chromosome respectively.

– The iterations continue until convergence is achieved. The population is considered to have converged when none of the chromosomes in the current population differs from the fittest by more than two bits.

– At convergence the fittest chromosome is retained and *N-1* chromosomes are randomly generated. The trials continue until a satisfactory solution is obtained.

One advantage of this approach is that the best solution is carried unchanged into the next generation. This elitist reproduction ensures that no better solution has been found up to that point.

The fitness of the chromosomes is evaluated is using the minimum description length (MDL) criterion [11] :

$$MDL(k) = N \cdot ln(\sigma_k^2) + k \cdot ln(N) \qquad (1)$$

where k is the number of parameters, σ_k^2 is the residual mean square error (MSE) and N is the number of samples of the input signal. This criterion is based on a compromise between model coding cost and residual coding cost. The first term on the right side corresponds to the encoding of the residuals and decreases with an increasing k whereas the second term corresponds to the encoding of the model and increases with k. Thus a minimum exists for the MDL function, and eventually, the selected k is the value corresponding to this minimum.

3 Applications

3.1 Polynomial Fitting Models

A polynomial fitting model can be expressed as

$$y(n) = \sum_{m=0}^{M} a_m p_m(n) + e(n) \quad n = 1, \ldots, N \qquad (2)$$

where a_m are the parameters, p_m the regressors or predictors and $e(n)$ is the fitting error. The regressors are polynomials over delayed signal samples. Classical exhaustive methods to explore the significant regressors are often time consuming [8]. On the other hand a comparison with the genetic approach looks promising [5].

Each position inside a chromosome indicates the presence of a regressor with its binary value. In the case of an autoregressive (AR) model, the length of the chromosomes is the maximum sample delay. Initially, each position in the chromosomes is set randomly to a binary value. Then, the fittest chromosome according to the MDL criterion is retained and the next generation is created. When the fittest chromosome has not changed for a certain number of generations the algorithm stops.

Table 1 shows the average number of iterations until convergence is reached as a function of the population and the length of a chromosome. The simulations have been run 100 times for the following example :

$$y(n) = a_1 \cdot y(n-4) + a_2 \cdot y_(n-8) + e(n) \qquad (3)$$

We observe that the total number of evaluations is much lower than what an exhaustive search on all possible regressor combinations would require. It is obvious that the number of iterations increases as a function of the chromosome length. On the other hand, larger population sizes require in general fewer generations. The GA can also be applied to more

	Chromosome length		
Population	10	20	30
9	7.67±3.7	18.5±6.7	31.05±9.0
11	5.84±2.3	14.23±5.2	29.48±8.3
13	6.85±3.3	12.96±4.3	23.80±7.8

TAB. 1 — **Average number of iterations**

complex polynomial models as given below :

$$\begin{aligned} y(n) &= a_1 \cdot x_(n-3) + a_2 \cdot x_(n-1) \cdot x_(n-3) \\ &+ a_3 \cdot x_(n-2) \cdot x_(n-4) \\ &+ a_4 \cdot x_(n-3) \cdot x_(n-5) + e(n) \end{aligned} \qquad (4)$$

$y(n)$ is the output of a nonlinear polynomial filter where the input signal $x(n)$ is white noise with zero mean and unity variance. Again, the task of the GA is to converge

Population	Iterations	Evaluations
7	24.97±8.8	150.82
9	17.88±6.2	144.04
11	15.33±5.1	154.30
13	14.57±4.6	175.84

TAB. 2 — **Average number of iterations and evaluations**

to the given model selecting the correct regressors. The chromosome lengths corresponds to the maximum number of possible regressors. If all linear and second order models are considered with a maximum delay of 5 samples the resulting chromosome length is 20. Figure 2 shows the number of iterations and evaluations for a given population size. The number of evaluations indicates how many models have been evaluated until convergence. We observe that the optimal GA configuration is obtained for a population size of 9 chromosomes. Again, the computational complexity required by an exhaustive model search is incomparably higher.

3.2 Radial Basis Functions

An RBF network may be considered as a three layer feed forward neural network. The input layer is made up of source

nodes. The second layer is a hidden layer and the output layer supplies the response of the network to the activation patterns. An RBF network output with Gaussian functions can be expressed as

$$y(n) = \sum_{i}^{M} a_i e^{-\|\mathbf{c}_i - \mathbf{x}\|/\beta_i^2} + e(n) \quad n = 1, \ldots N. \qquad (5)$$

where $\mathbf{x}$ is the input vector, $\mathbf{c}_i$ is a center, β_i is a real constant (width) and M the number of centers. The transformation from the input space to the hidden-unit space is nonlinear whereas it is linear in the parameters. In prediction problems the task is to select the centers and the widths such as to minimize the resulting MSE. Different training methods have been developed [4, 1] for this purpose.

In our approach, we simply recall the regressive model given in (2) where the radial functions can be considered as regressors. For initialization, a number of centers is selected randomly from the input space. This number corresponds to the length of the chromosomes which are encoded into binary values. At evaluation, each *1* in a chromosome enables the corresponding center to participate in the RBF output layer whereas centers with a designated *0* are not taken into account. The fitness of each chromosome is again computed according to the MDL criterion. The GA selects subsets of these centers and computes the MSE after each generation. For a given set of centers, the widths are computed as

$$\beta_i = \frac{1}{N} \sum_{j=1}^{N} \|\mathbf{c}_i - \mathbf{x}_j\| \quad i = 1, \ldots M. \qquad (6)$$

In [12] a cooperative-competitive GA has been presented which encodes the center and widths into binary streams. However, our method is much simpler since the centers are directly selected from the input space. In our approach, the task of the GA is to select adequate subsets providing a MSE.

In a first attempt we compared the GA to the Kohonen self–organizing map [7] where the GA found significantly better models than the clustering algorithm. Here, we compare the GA with the orthogonal least squares (OLS) learning algorithm [1]. This method selects the centers from the input space as the GA does and may therefore be better suited for comparison. The RBF network is also considered as a regressor model where the regressors are changed into a orthogonal basis. This allows us to add centers to a given set where the MSE of the new model depends only on the new centers added and is computed easily. Here, the same search method is applied as introduced in [3] : The best model with one center only is expanded to a model with two centers where all the centers of a subset are tested. More centers are added and the expansion continues until optimal model complexity is reached according to the MDL criterion.

Figure 1 compares the GA with the OLS algorithm for the data set of sunspot series. It depicts simultaneously the number of selected centers, the residual MSE and the MDL as a function of the number of centers in the subset. The simulations have been run 100 times for each initial subsets of 10, 30, 50, 80 and 100 centers. The centers of a subset consist of randomly selected input data. In Fig. 1 we see the mean of the number of selected centers and the corresponding

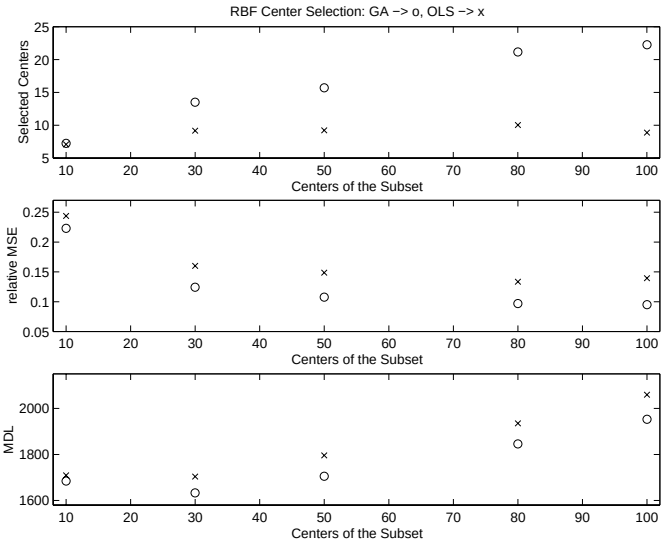

FIG. 1 — Comparison of the GA and the OLS

MSE. For an increasing dimension of the subset the GA also selects an increasing number of centers for the best model. On the other hand, the number of centers selected by the OLS algorithm seems to be independent of the dimension of the subset. We observe that the MSE provided by the GA is lower than the MSE of the models trained by the OLS algorithm. It is therefore necessary to validate the models by using a model selection criterion. This is done here using again the MDL criterion given in Equ.(1). We observe that the coding cost for the models found by the OLS algorithm is higher than for the models provided by the GA. Therefore, we conclude that the latter is more suited for model selection than the OLS algorithm.

3.3 Further Applications

The presented model selection algorithm may be applied to further applications. In a piecewise linear model for example, one linear sub-model is associated with each region of a state-space decomposition. Existing methods are mainly based on adaptive algorithms and geometric interpretations [10, 9]. The problem can be reformulated as a regressor model where the sub-models are selected with the proposed GA.

The same ideal holds for signal segmentation. A signal may be divided into linear segments where the GA is used to find the segment borders.

Piecewise linear modeling and signal segmentation are subjects of further studies.

Conclusion

We have shown how a simple genetic algorithm using a simplex reproduction scheme can be efficiently used to select the regressors of a linear regression model. The performance has been measured in term of iterations and number of evaluations until convergence on a simple linear autoregressive model and on a nonlinear polynomial model. The computational

complexity is much lower than an exhaustive search method would require. In a second application, we have applied the GA as a center learning algorithm of a RBF network. Model complexity and residual mean square error have been compared to the well–known OLS learning algorithm. The model complexity provided by the GA depends linearly on the number of centers available for selection whereas the number of parameters given by the OLS algorithm remains constant. On the other hand the GA selects models with a lower residual MSE. However, according to the MDL criterion that assumes that the best models are those who permit the shortest encoding of the data, the GA performs better model selection than the OLS algorithm.

Finally, we propose to use the GA for further applications such as linear sub-model selection in piecewise linear modeling or signal segmentation.

Références

[1] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Networks*, 2 :302–309, March 1991.

[2] D.E. Goldberg. Sizing Populations for Serial and Parallel Genetic Algorithms. *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pages 70–79, 1989.

[3] R. Grüter, J.M. Vesin, E. Pruvot, C. Seydoux, and M. Fromer. Linearity Assessment of Heart Rate Time Series. *Proc. 1996 IEEE Engineering in Medicine and Biology Society (EMBS), Amsterdam, The Netherlands*, November 1996.

[4] S. Haykin. *Neural Networks*. Macmillan Publishing Company, 1994.

[5] H. Kargupta and R.E. Smith. System Identification with Evolving Polynomial Networks. *Proc. 4th Int. Conf. on Genetic Algorithms, San Diego, CA*, July 1991.

[6] C.L. Karr. *Air–Injected Hydrocyclone Optimization via Genetic Algorithm, Handbook of Genetic Algorithms edited by L. Davis*. Van Nostrand Reinhold, 1991.

[7] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78 :1464–1480, 1990.

[8] M. J. Korenberg and L. D. Paarman. Orthogonal Approaches to Time-Series Analysis and System Identification . *IEEE SP Magazine*, pages 29–43, July 1991.

[9] M. Mattavelli, J.M. Vesin, E. Amaldi, and R. Grüter. A New Approach to Piecewise Linear Modeling of Time Series. *Proc. 1996 IEEE Digital Signal Processing Workshop, Loen, Norway*, August 1996.

[10] C.J. Pantaleón Prieto, I.S. Santamaría Caballero, and A.R. Figueiras Vidal. Competitive local linear modeling. *Signal Processing*, 49 :73–83, 1996.

[11] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[12] B.A. Whitehead and T.D. Choate. Cooperative Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction. *IEEE Transactions on Neural Networks*, 7 :869–880, July 1996.