Wind Prediction Using Genetic Algorithms and Gene Expression Programming

Juan J. Flores
juanf@zeus.umich.mx

Mario Graff
mgraffg@lsc.fie.umich.mx

Erasmo Cadenas *
ecadenas@zeus.umich.mx

División de Estudios de Postgrado
Facultad de Ingeniería Eléctrica
Universidad Michoacana de San Nicolás de Hidalgo

Abstract

The work reported in this paper explores the applicability of Genetic Programming (GP) to wind prediction. Unlike other AI tools, GP provides closed-form models for the time series under analysis. By changing the Function Set, GP is able to provide ARIMA like models, linear and non-linear, and arbitrary models, generally non-linear. The former models are more understandable, but the latter provide an alternative, perhaps with not so intuitive forms. We used a form of Gene Expression Programming (GEP) for the implementation of our modelling tools. GP has shown to be a good alternative to provide models for wind prediction, it has beaten an ARIMA model produced using Minitab.

1 Introduction

The availability of the Eolic energy depends on the wind speed, which is a random variable; this random variable is a problem in the planning and preoffice of eolic energy. In order to approach this problem, diverse prediction techniques have been used: ARIMA, Kalman Filters, Neural Networks,

*División de Estudios de Postgrado de la Facultad de Ingeniería Mecánica

etc, most of them with satisfactory results. In the present work we use Genetic Algorithms, Genetic Programming (The implementation follows the idea of Gene Expression Programming as defined by C. Ferreira [1] which is a formed of Genetic Programming), and a statistical modelling procedure ARIMA (this model was produced using Minitab [5]), taken as reference for comparison purposes. The purpose of this work is to present an alternative solution to the problem of time series prediction; the obtained results are satisfactory and even superior to the traditional method ARIMA. The techniques are applied to the time series formed by monthly averages of the wind speed in the Isthmus of Tehuantepec Mexico. The measurements were made at different heights (20, 30 and 40 ms) with sensors of high quality and exactitude.

Section 2 describes the related work. Section 3 describes Genetic Algorithms and Gene Expression Programming. Section 4 describes the implementation of ECSID . Section 5 shows the results and finally Section 6 shows the conclusions.

2 Related Work

Lie et.al. [8] created a time prediction program using gene expression programming [1], their program used two methods for time series prediction: The slide window prediction method to find relations between future and past data and the differential equation prediction method to mine ordinary differential equations from the training set and predict future trends based on the initial conditions, their method does not include the prediction errors.

Szpiro [7] used genetic algorithms to find equations that model the behavior of a time series. He used the slide window prediction method to find the relations between the past and future trends. He codified the problem as a variable string containing variables, constants, and parenthesis. The string represents a function that models the observed data. His method does not include the prediction errors.

3 Genetic Algorithms and Gene Expression Programming

Genetic Algorithms [3] (GA) and Gene Expression Programming [1] (GEP) are evolutionary tools inspired in the Darwinian principle of natural selection and survival of the fittest individuals. These methods used an initial random population and apply genetic operations to this population until the algorithm finds an individual that satisfies some termination criteria.

In order to simulate the evolutionary process both GA and GEP follow the next steps.

Initialization Creates an initial random population.

Evaluation Evaluates all the individuals and test whether or not the best one satisfies the termination criteria.

Selection Use fitness proportional selection and apply the genetic operations to this population.

GA uses a fixed chromosome structure, which can be an array of bits, numbers, characters, etc. To use GA the problem is codified as a fixed chromosome and then the problem is solved using an evolutionary process. The genetic operations more widely used are crossover, selection and mutation.

GEP is similar to Genetic Programming [4] (GP), it is an evolutionary algorithm that evolves computer programs. The basic idea behind GEP is a clever representation for the chromosomes (a string instead of a tree), which leads to an easier implementation. Figure 1 shows this representation.

4 Implementation

ECSID uses GA and GEP to obtain a formula that models the training set, using a slide window prediction method [8]. The slide window prediction method uses a window of size h; the window contains the actual data, and our model computes the synthetic time-series and the prediction errors for that time window. ECSID uses a window of size $h = 16$. Equation 1 shows the general ARIMA model where $f(i)$ represents the time series at instant i and $e(i)$ is the vector of prediction errors.

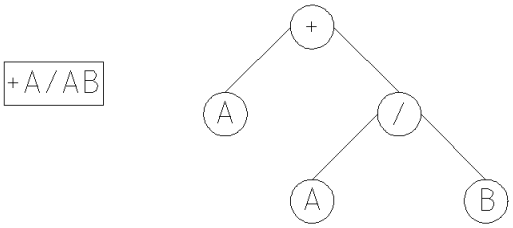


Figure 1: Convert GEP's chromosome to a tree expression

$$f(n) = \Sigma_{i=t-h}^{t-1} a_i f(i) + \Sigma_{i=t-h}^{t-1} b_i e(i) \quad (1)$$

A GA chromosome is formed by real values, each one representing the coefficient of previous data or errors in the prediction. EvaluationGA shows the evaluation algorithm for GAs where F is the past values of the actual data (i.e. the times-series) and E is the vector of prediction errors.

```
EVALUATIONGA(chromosome c)
 1  real result ← 0
 2  integer count ← 1
 3  integer i ← 1
 4  while count <= LENGTH(c)/2
 5  do result ← result + c[count] * F(n − count)
 6      count ← count + 1
 7  while count <= LENGTH(c)
 8  do result ← result + c[count] * E(n − i)
 9      count ← count + 1
10      i ← i + 1
11  return result
```

A GEP chromosome is formed by constants, terminals or functions. The

terminals are pointers to past values and prediction errors. We use a novel function for the function set. We introduce a function called "coefficient". By introducing this function we disallow the product function and make sure the evolved functions are always a linear combination of the ARIMA terms.

Both GA and GEP use fitness proportional selection and the best individual is always inserted to the new generation.

5 Results

We performed two kinds of experiments. One using the prediction errors, and other one without taking into account prediction errors. All the experiments involving GA or GEP were run 10 independent times [4] for 2500 generations. Table 1 shows the results for the first kind of experiments and Table 2 shows the results for the second set of experiments.

The first row of the Table 1 shows the result using a statistical method, used as a reference for comparisons. The first model is an ARIMA model, where statistics was used to determine which components were relevant, and minimum squares were used to determine their numerical values [6]. The following rows are the results using GEP (Row 2,3 and 4) and GA (Row 5). The column $\Sigma|e|$ is the sum of absolute errors in the training set and $\Sigma|e_v|$ is the sum of absolute errors in the validation set. The models obtained by each of the experiments are shown in Equations 1 to 5.

Method	Function set	Equation	$\Sigma\|e\|$	$\Sigma\|e_v\|$
Statistics	N/A	2	121.22153	42.5854969
GEP	$\{+, -, \text{coefficient}\}$	3	76.090004	39.1250448
GEP	$\{+, -, *, /\}$	4	70.036970	40.5830182
GEP	$\{+, -, *, /, \sqrt{}, \sin, \cos, \log\}$	5	66.127730	34.6902514
GA	N/A	6	46.327045	33.6182107

Table 1: Results with prediction errors

$$\begin{aligned} f(n) = f(n-1) + f(n-2) - f(n-13) - 0.997e(n-1) \\ - 0.7976e(n-12)0.7956e(n-13) \end{aligned} \quad (2)$$

$$f(n) = e(n-13) + 9.1124784879 \tag{3}$$

$$f(n) = \frac{e(n-13)(f(n-15) + 0.114028e(n-13))}{f(n-12)} \tag{4}$$

$$\begin{aligned} f(n) = {} & \sqrt{(}\sin(f(n-15)))\sqrt{(}\sin(e(n-13))) \\ & \sin(e(n-13)))) + \sqrt{(}\sin(e(n-13)))) \\ & + e(n-13) + 9.1094265 \end{aligned} \tag{5}$$

$$\begin{aligned} f(n) = {} & -0.0021162033f(n-3) - 0.00884676f(n-9) - 0.17873764f(n-10) \\ & + 0.31975746f(n-11) + 0.69887733f(n-14) + 0.117706776f(n-16) \\ & + 0.04591465e(n-2) - 0.0021162033e(n-3) + 0.0049743652e(n-5) \\ & + 0.17252827e(n-8) + 0.7262335e(n-9) + 0.20020199e(n-10) \\ & - 0.27034664e(n-11) + 0.9445238e(n-12) + 0.92421293e(n-13) \\ & - 0.6009078e(n-14) + 0.039580822e(n-15) + 0.078193665e(n-16) \end{aligned} \tag{6}$$

You can see from Table 1 that all the experiments are better than the statistical method in the training data, but they are not so good in the validation set $\Sigma|e_v|$. This fact occurs because all models involve prediction errors that are zero in the validation set.

In the second experiment the prediction errors were not included in the models. As you can see in Table 2 all models are better than the statistical method in both, the training and validation sets. Our best result in the validation set is shown in the last row of Table 2, this result was obtained thorough Gene Expression Programming running the experiment 10 times during 5000 generations. The models obtained by each of the experiments are shown in Equations 6 to 10.

$$f(n) = f(n-12) + 0.210271784 \tag{7}$$

$$f(n) = 0.33388233f(n-12) + 0.33388233(f(n-1) + 8.92381) \tag{8}$$

Method	Function set	Equation	$\Sigma\|e\|$	$\Sigma\|e_v\|$
GEP	$\{+,-,\text{coefficient}\}$	7	85.06000	17.8399994
GEP	$\{+,-,*,/\}$	8	69.94014	23.4684712
GEP	$\{+,-,*,/,\sqrt{},\sin,\cos,\log\}$	9	62.22119	22.4415914
GA	N/A	10	69.30226	22.9705186
GEP	$\{+,-,*,/,\sqrt{},\sin,\cos,\log\}$	11	50.99468	15.0940472

Table 2: Results without prediction errors

$$\begin{aligned} f(n) = {} & 2\sqrt{(}f(n-12))\cos(\cos(f(n-12)) - 6.25254) \\ & + 1.1831827\sqrt{(}f(n-12)) + \sqrt{(}f(n-1)) \end{aligned} \tag{9}$$

$$\begin{aligned} f(n) = {} & 0.081225395f(n-1) + 0.081225395f(n-2) \\ & + 0.081225395f(n-4) + 0.081225395f(n-10) \\ & + 0.47588778f(n-12) + 0.081225395f(n-13) \\ & + 0.081225395f(n-14) + 0.081225395f(n-15) \end{aligned} \tag{10}$$

$$\begin{aligned} f(n) = {} & 0.1408\cos(\frac{\sin(\ln(f(n-12)))}{\sin(f(n-14))} + f(n-2)) \\ & + \cos(\frac{8.5209}{\frac{6.8594}{\ln(\ln(f(n-3)))} - 5.6555} + f(n-12) - 6.8962) \\ & + \frac{(0.7323f(n-12) + 4.1514)\cos(2\sin(f(n-12)))}{\frac{f(n-9)}{f(n-14)} - f(n-2)} \\ & + 0.5986f(n-12) \end{aligned} \tag{11}$$

Figure 2 shows the time series and the results of the statistical model for the training and validation sets.

Figure 3 shows the results for the model of Equation 11 (best model). Figure 3(b) shows the validation set and the actual data, you can see that this prediction is better than the one observed in the Figure 2(b)

6 Conclusions

We presented the results of modeling a time series, specifically a time series that represents wind speed, and use the resulting model to forecast wind for one year. All experiments were developed using an implementation of the Evolutionary Computing methods called ECSID . ECSID has found a better model than the one using a statistical model. To find a proper model ECSID uses evolutionary computation and the process is done without human intervention. ECSID found linear, non-linear, and ARIMA models and all of them are better than the ARIMA model produced by Minitab.

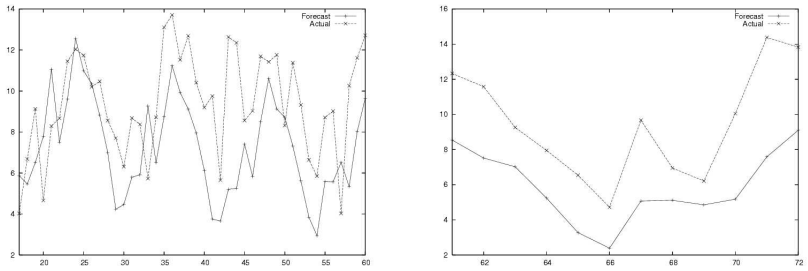
We have not been able to find any related work that use genetic programming in wind the time series.

- GP has proved its effectiveness to reproduce the statistical procedure, without the use of any kind of statistical knowledge.
- Linear models can be important, because the engineers are used to use them.

ECSID can be downloaded from [2].

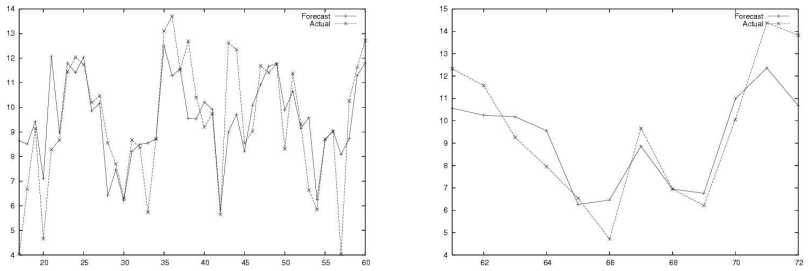
References

[1] C. Ferreira. Gene expression programming: A new adaptive algorithm for solving problems. In *Complex Systems*, volume 13, pages 87–129,



(a) Training set (b) Validation set

Figure 2: Training and validation set of Equation 2



(a) Training set (b) Validation set

Figure 3: Training and validation set of Equation 11

2001.

[2] Mario Graff and Juan J. Flores, January 2005. http://sourceforge.net/projects/ecsid.

[3] J. H. Holland. *Adaptation in Natural and Artificial Systems 2nd ed.* University of Michigan Press, 1992.

[4] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems).* The MIT Press, 1992.

[5] Minitab. http://www.minitab.com/.

[6] Steven C. Whellwright Spyros Makridakis and Rob J. Hyndamn. *Forecasting Methods and Applications 3er ed.* John Wiley & Sons, Inc., 1992.

[7] George G. Szpiro. Forecasting chaotic time series with genetic algorithms. Physical Review E., 1997.

[8] Li Chuan Chen Anlong Zuo Jie, Tang Changjie and Yuan Chang'an. Time series prediction based on gene expression programming. International Conference for Web Information (WAIM04), Springer Verlag, 2004.