Genetic programming to obtain mathematical models from turbine data using symbolic regression

Erick Jacob Romero Pichardo

Centro de Investigación de Matemáticas, A.C.

erick.romero@cimat.mx

ABSTRACT

This document describes how to find a mathematical model from data of turbines using genetic programming, particularly is explained how symbolic regression was used to obtain the model.

INTRODUCTION

Turbines require control based in mathematical models, but sometimes this models are not gotten easily. Symbolic regression can be an option .

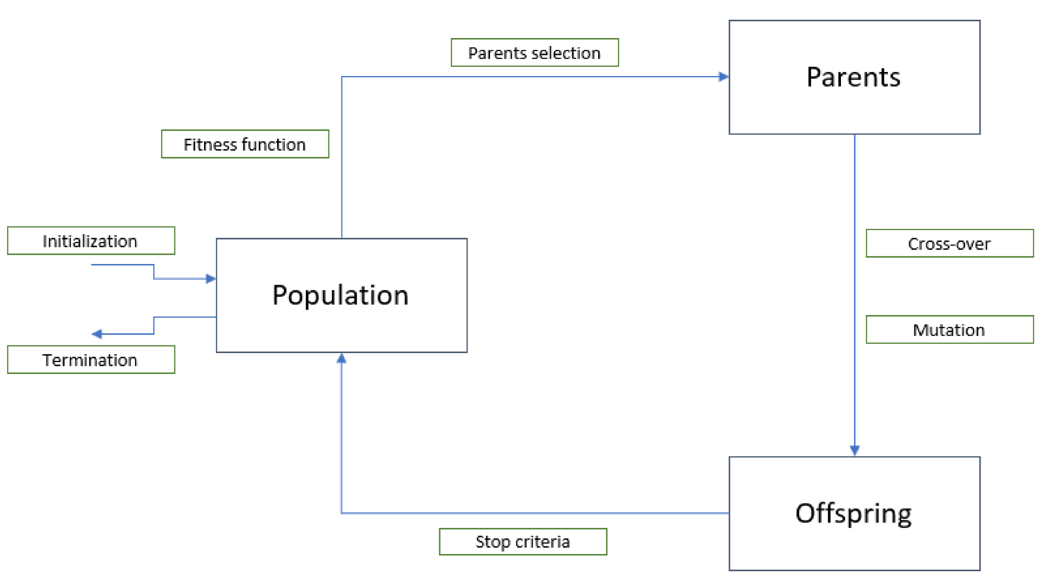
The information provide by the turbine sensors is the inlet temperature (Te), fuel Flow (Qc), inlet pressure (Pe), angular velocity (rpm), outlet pressure (Ps) and outlet temperature (Ts).

The purpose of the mathematical model is to describe the relationship between the inlet variables (Te, Qc, Pe, rpm) with the outlet variables (Ts and Ps), so at the final we should have two mathematical models, one for the outlet pressure (Ps) and another for the outlet temperature (Ts).

GENETIC PROGRAMMING

Genetic algorithms are adaptive methods, generally used in parameter search and optimization problems, based on sexual reproduction and the principle of survival of the fittest (Fogel, 2000-2006).

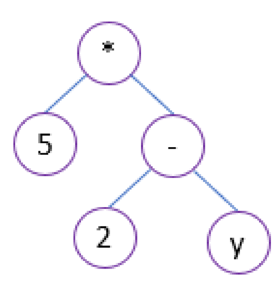
The biological base of genetic programming is exactly the same as genetic algorithms. For this reason, the operation is very similar. The difference between one technique an the other one is the form to codify the problems, so this allows using genetic programming in series of situations that genetic algorithms can not be applied.



SYMBOLIC REGRESSION

Symbolic regression is an application of genetic programming, this tool has the same goal as linear regression, but the space of search is major.

Symbolic regression looks for an algebraic expression which identifies the behavior between the input and output data accessing to all kind of functions and mathematical operations.



Representation example

INITIALIZATION

Initialization is the process to create a population of size *tp*. Each individual is represented by an operation tree and Bakus Naur algebra is used.

BAKUS NAUR'S ALGEBRA

Metasymbol	*Meaning*
:=	It is define like
|	Alternative option
$(a \mid b)$	a or b, but not both
$[sthg]$	0 or 1 times can see something
$< sthg >$	0 or more times of something
$'x1, x2, ...'$	terminal symbol
.	end of the rule[a]

Example

BINARY STRING

00000010	00001001	00000101	00000100	00001010	00000001	00000011

2	9	5	4	10	1	3

INTEGER STRING

S:=<expr> :=<expr> * <expr > (0)
| expr> + < expr> (1)
| expr> − < expr> (2)
| expr> / < expr> (3)
| [X1] (4)
| [X2]. (5)

S :=< expr >	
< expr > − < expr >	$(2\ MOD\ 6 = 2)$
(< expr >/< expr >) − < expr >	$(9\ MOD\ 6 = 3)$
(X2/< expr >) − < expr >	$(5\ MOD\ 6 = 5)$
(X2/X1) − < expr >	$(4\ MOD\ 6 = 4)$
(X2/X1) − (X1)	$(10\ MOD\ 6 = 4)$
not used	$(1\ MOD\ 6 = 1)$
not used	$(3\ MOD\ 6 = 3)$

Final expression

$(X2/X1){-}X1.$

FITNESS FUNCTION

The fitness function is the sum of some coefficients, the first is the Pearson's coefficient (*r*), this indicates the relationship between data come from sensors (*ye*), and data obtained by the model (*yt*), the second R2 indicates how the curve is fitted utilizing residuals, then is subtracted a restriction due to the operation number (*no*), where is the parsimony coefficient, its value is 0.01, and it is to avoid growth of the trees, problem known like bloat. Then the fitness function is:

$$max \qquad f(x, ye, yt) = r + R^2 - \alpha * no$$

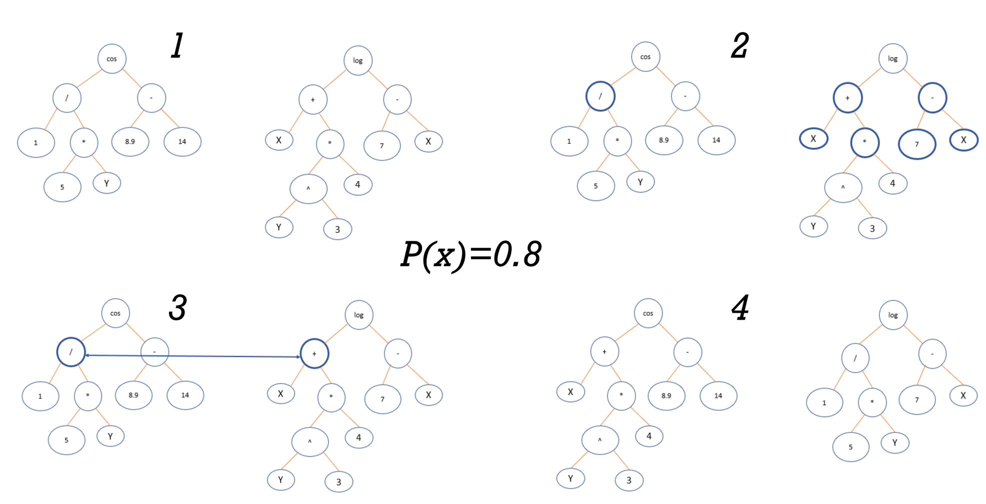
$$r = \frac{cov(ye, yt)}{S_{ye} * S_{yt}}, \quad cov(ye, yt) = \frac{\sum_{i=1}^{n} (ye_i - \bar{ye})(yt_i - yt)}{n-1}$$

$$S_{ye} = \sqrt{\frac{\sum_{i=1}^{n} (ye_i - \bar{ye})^2}{n-1}}, \quad S_{yt} = \sqrt{\frac{\sum_{i=1}^{n} (yt_i - \bar{yt})^2}{n-1}}$$

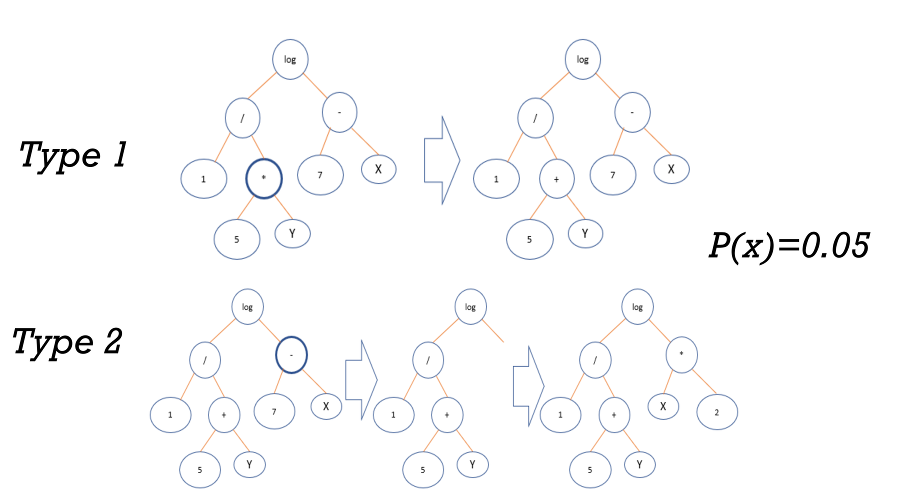
$$R^2 = \frac{St - Sr}{St}, \quad St = \sum_{i=1}^{n} (ye_i - \bar{ye})^2, \quad Sr = \sum_{i=1}^{n} (ye_i - yt_i)^2$$

Evaluation is carried out through the fitness function, 10% of population with major fitness is saved, this process is known like elitist strategy. In addition to this a selection by roulette of probabilities is made.

CROSS-OVER



MUTATION



TESTING

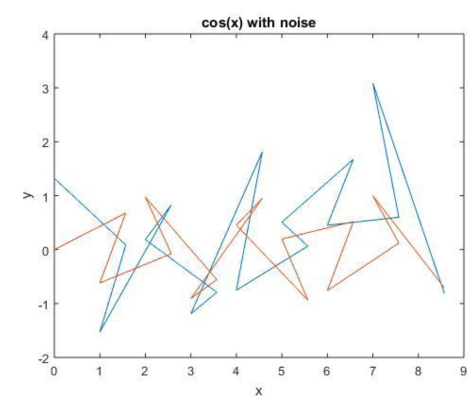
Several function were tested to see how was the behavior of the algorithm. Some white noise was added and here are the results for cosine function and 10db of it.

COS(X) FUNCTION

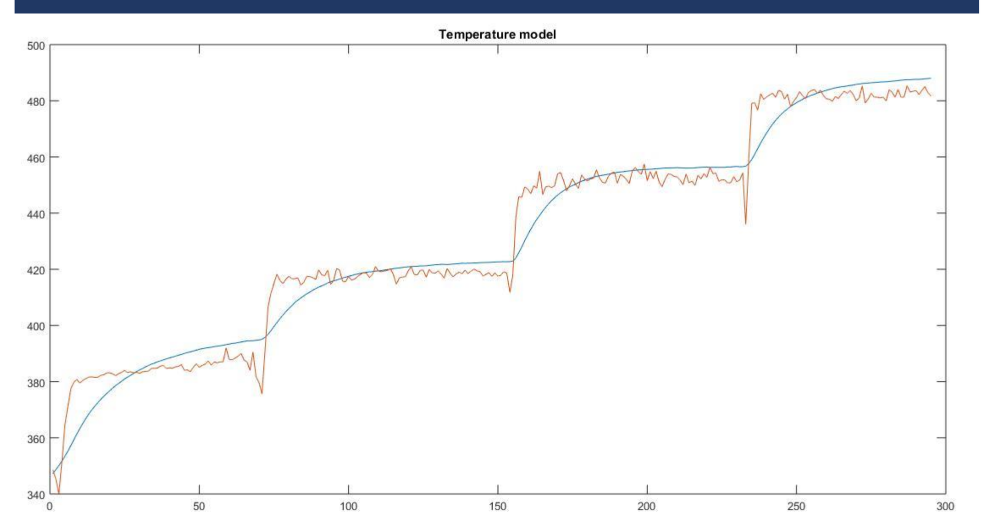
Function	Generations	Population size	Pearson coeff. + R^2
cos(x)	50	5	2
cos(x)	40	10	2

COS(X) FUNCTION WITH NOISE

Function	Generations	Population size	Pearson coeff. + R^2
exp(x/5)+4*sen(x^3)	20	10	-18.5890
(-)sen(x-7)	20	10	0.3271
sen(x^8+3)	50	15	0.7519
sin(x/sin(cos(2)))	80	15	0.9666

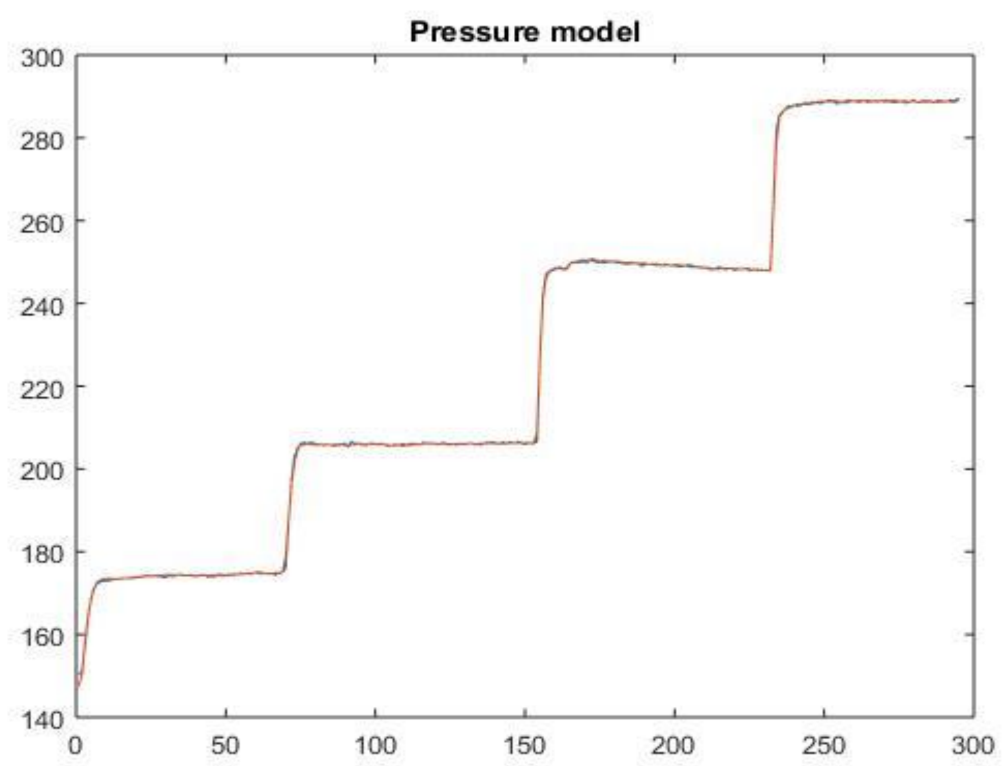


RESULTS



$$Ts_m = 6122 - 57.57Pe - 33120sen(Qc) + 1.037 \times 10^{-7} rpm^2$$

This model has a Pearson's coefficient of 0.9825, the *R2* coefficient is 0.9641 and the fitness 1.9467, although the fitness is good the mean square error is 47.2563. However the model tries to fit the curve.



The pressure model has a Pearson's coefficient of 1, the *R2* coefficient is 0.9999 and the fitness 1.9999, the mean square error here is 0.1695, so this model is an excellent model obtained by the algorithm.

$$Ps_m = 38.24 + 4.325 \times 10^{-13} Pe^7 + 2.141 \times 10^{-3}(Qc * Te * rpm) - 3.232 \times 10^6 (Te * Qc^3) + 3.337 \times 10^{21} \left(\frac{Qc^8}{log(Pe^6)}\right)$$

CONCLUSION

The algorithm was able to approximate the athematical functions proposed and obtain mathematical models for the turbine. Even this algorithm can approximate some model when there sensors with noise.

REFERENCES

(Ref.1) Rafael Alberto Moreno Parra. (2007 (Enero - Junio)). Programación genética: La regresión simbólica. Sistemas, 3, 76-85.
(Ref.2) Marcos Gestal, Daniel Rivero, Juan Ramón Rabuñal, Julián Dorado y Alejandro Pazos. (2010). Introducción a los Algoritmos Genéticos y la Programación Genética. España: Universidad de la Coruña.
(Ref.3) Omar Alexander Reina Flores. (2019). Estudio de la programación gramatical. España: Universidad Politécnica de Madrid.
(Ref.4)Farzad Noorian, Anthony M. de Silva, Philip H. W. Leong. (2016). gramEvol: Grammatical Evolution in R. Journal of Statistical Software, Volume 71, Issue 1.
(Ref.5) Angélica María Ramírez Botero. (2014). Estudio de un método basado en programación genética para la solución de ecuaciones diferenciales ordinarias y parciales de dos variables. 05/08/2019, Sitio web: https://pdfs.semanticscholar.org/d2e8/744ccd843a31b9acda042a4fad01effa8e2b.pdf

International Symposium on Mathematical Methods Applied to the Sciences, 2020