Exploring Polynomial Structure of Data with Genetic Algorithms

Francisco Coelho	João Neto
fc@di.uevora.pt	jpn@di.fc.ul.pt

Antes de submeter, temos de considerar Polynomial regression na wikipedia e a regressão não-linear em geral.

Como este trabalho anda no meio de áreas muito trabalhadas, temos de ter muito cuidado com a questão da **investigação original**.

Abstract

Many applications require models that have no acceptable linear approximation and many nonlinear models are defined by polynomials. The use of genetic algorithms to find polynomial models is decades old but still poses challenges due to the complexity of the search and different definitions of "optimal" solution. This work describes a general method based in genetic algorithms to find "empirical" polynomial regressions.

GIVE A SUMMARY DESCRIPTION AND RESULTS OF "OUR" APPROACH.

Introduction

With notable exceptions (*e.g.* neural networks), most machine learning regression techniques are based on linear models. This assumption has many advantages including, for example, reduced computational complexity and strong theoretical framework. However nonlinearity in unavoidable in many application scenarions *e.g.* phase transitions or systems with feedback loops so common in ecology, cybernetics, robotics and other areas. Nevertheless the variety and number of phenomena that can be adapted into a linear model is amazing.

Polynomials, one of the most studied subjects in mathematics, generalize linear functions and define, perhaps, the simplest and most used nonlinear models. Applications of polynomial include colorimetric calibration (APPL1-2005), explicit formulae for turbulent pipe flows (APPL2-1999), computational linguistics (APPL8-2009) and more recently, analytical techniques for cultural heritage materials (APPL3-2010), liquid epoxy molding process (APPL4-2011), B-spline surface reconstruction (APPL5-2012), product design (APPL6-2012) or forecasting cyanotoxins presence in water reservoirs (APPL7-2013). Besides the huge range of quite different areas, the work for each one of these polynomial models used, somewhere, a genetic algorithm.

Genetic algorithms (GA) where, arguably, one the most popular "hot" topics of research in the recent decades but with good reason since they outline an optimization scheme easy to conceptualize and with very broad application. If a nonlinear (or otherwise) model requires parameterization GAs provide a simple and often effective approach to search for localy optimal parameters. Research related to genetic algorithms abound and spans from the 1950's seminal work of Nils Aall Barricelli (BARRICELLI-1962) in the Institute for Advanced Study of Princeton to today's principal area of study of thousands of researchers, covered in hundreds of conferences, workshops and other meetings. Perhaps the key impulse to GAs come from John Holland's work and his book (BOOK-JOHNHOLLAND).

One interesting "flavour" of genetic algorithms, named *genetic programming* by John Koza (KOZA1-1992), proposed the use of GAs to search the syntatic stucture of complex functions. This syntatic structure search is keen to the central ideas of deep learning (DEEPLEARNING1-2012 and DEEPLEARNING2-2009), the subarea of machine learning actually producing the most promising results (e.g. DLAPPL1-2013). It is also related to the work presented in this paper in the sense that, unlike linear models that have a simple structure ($y = \sum_i \beta_i x_i$) nonlinear (in particular polynomial) models pose an aditional "structure" search problem.

The idea of using GAs to find a polynomial regression is not new (GAPOLY5-2006, GAPOLY2-2008 and GAPOLY4-2009) but still generates original research (GAPOLY1-2011 and GAPOLY3-2011). In line with that research this work describes a general method to find a polynomial regression of a given dataset. The optimal regression minimizes a cost function that accounts for both the *mean square error* and a *regularization* factor to avoid overfitting by penalizing polynomials that are "too complex".

A method that produces adequate models "directly" from observed complex data has many uses. For example by a scientist to better understand the source of the data or by an autonomous agents adapting to the environment.
MORE REASONS?

RESULTS OUTLINE

The remainder of this paper is organized as usual: the next section describes the details of our method and is followed by a presentation of some performance results. The last section draws some conclusions and points future research tasks.

Polynomial Regression with Genetic Algorithms

The canonical representation of a polynomial is a sum

$$p\left(x_1, \ldots, x_n\right) = \sum_i \beta_i m_i.$$

In this sum each m_i is a monomial, $m_i = \prod_{j \in M_i} x_j^{\alpha_{ij}}$, the exponents are non-negative integers, $\alpha_{ij} \in \mathbb{N}_0$, and the coeficients are real valued, $\beta_i \in \mathbb{R}$. For example $p(x_1, x_2, x_3) = 2x_1 + x_2x_3 + \frac{1}{2}x_1^2x_3$ has monomials $m_1 = x_1, m_2 = x_2x_3$ and $m_3 = x_1^2x_3$ coefficients $\beta_1 = 2, \beta_2 = 1$ and $\beta_3 = 1/2$ and exponents $\alpha_{1,1} = 1, \alpha_{2,2} = 1, \alpha_{2,3} = 1, \alpha_{3,1} = 2, \alpha_{3,3} = 1$ and all other $\alpha_{ij} = 0$.

> **NOTA** Este parágrafo deu-me uma ideia: os α definem uma matriz em que as linhas são monómios e as colunas variáveis; "Multiplicando" esta matriz por um vector, obtemos o polinómio. Além disso, podemos continuar a definir a condição de regularização como antes, mas também podemos considerar condições *na* matriz (por exemplo, o número de entradas não nulas ser esparso ou a sua soma ser dominada por $\log nm$)...

This makes the problem of structure search very clear: except for the trivial cases, the number of possible monomials given n variables and a maximum joint degree d grows exponentialy with either n or d. But more importantly, the polynomial regression problem can be split into two subproblems:

1. for a given set of monomials $q_1, \ldots, q_n$, find regression coefficients $\beta_1, \ldots, \beta_n$ that minimize an adequate cost function;
2. find the fittest set of monomials;

Our line of work is to solve the first problem with linear regression and the second with GAs.

[EVALUATION STRATEGY]

[RESULTS]

HOW DO WE DO THE

- polynomial encoding
- cost function
- genetic operators
- selection of other search parameters (*eg* population size, maxiter, *etc*)

Polinomial Encoding

Describe our method to encode polynomial instances

Cost Function

Define the cost function

Given a dataset $D = \left\{ x_1^{(i)}, \ldots, x_m^{(i)}, y^{(i)} \in \mathbb{R}^{m+1} : i = 1, \ldots, n \right\}$ with n observations of $m+1$ variables $X_1, \ldots, X_m, Y$. We want to find a polynomial that fits $y = h\left(x_1, \ldots, x_m\right)$ and for that purpose we apply the usual *root-mean-square error* as a cost function of a candidate fitting polynomial h_Θ defined by parameters Θ:

$$J_{fit}\left(\Theta; D\right) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - h_\Theta \left(x_1^{(i)}, \ldots, x_m^{(i)} \right) \right)^2$$

Genetic Operators

Define the genetic operators

Other Search Parameters

Explain the remaining search parameters

Results

- Measured quantities
- error
- number of iterations to convergence
- memory usage
- F1, ROC, ?

- Selection of datasets and regression algorithms
- Summary Figures and Numeric results

Conclusion

References

the references are defined "here" in the source but hidden in the resulting documents

1. Ghai, Dhruva, Saraju P. Mohanty, and Garima Thakral. "Fast Analog Design Optimization using Regression based Modeling and Genetic Algorithm: A Nano-CMOS VCO Case Study." Proceedings of the 14th IEEE International Symposium on Quality Electronic Design (ISQED). 2013.

2. Rezania, Mohammad, Akbar A. Javadi, and Orazio Giustolisi. "Evaluation of liquefaction potential based on CPT results using evolutionary polynomial regression." Computers and Geotechnics 37.1 (2010): 82-92.

3. Zain, Azlan Mohd, Habibollah Haron, and Safian Sharif. "Genetic algorithm and simulated annealing to estimate optimal process parameters of the abrasive waterjet machining." Engineering with computers 27.3 (2011): 251-259.

4. Garg, A., and K. Tai. "Comparison of regression analysis, artificial neural network and genetic programming in handling the multicollinearity problem." Modelling, Identification & Control (ICMIC), 2012 Proceedings of International Conference on. IEEE, 2012.