Information Sciences 181 (2011) 5227–5239

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multi-stage genetic programming: A new strategy to nonlinear system modeling

Amir Hossein Gandomi [a,*], Amir Hossein Alavi [b]

[a] College of Civil Engineering, Tafresh University, Tafresh, Iran
[b] School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 12 August 2010
Received in revised form 6 April 2011
Accepted 5 July 2011
Available online 23 July 2011

Keywords:
Multi-stage genetic programming
Nonlinear system modeling
Engineering problems
Formulation

ABSTRACT

This paper presents a new multi-stage genetic programming (MSGP) strategy for modeling nonlinear systems. The proposed strategy is based on incorporating the individual effect of predictor variables and the interactions among them to provide more accurate simulations. According to the MSGP strategy, an efficient formulation for a problem comprises different terms. In the first stage of the MSGP-based analysis, the output variable is formulated in terms of an influencing variable. Thereafter, the error between the actual and the predicted value is formulated in terms of a new variable. Finally, the interaction term is derived by formulating the difference between the actual values and the values predicted by the individually developed terms. The capabilities of MSGP are illustrated by applying it to the formulation of different complex engineering problems. The problems analyzed herein include the following: (i) simulation of pH neutralization process, (ii) prediction of surface roughness in end milling, and (iii) classification of soil liquefaction conditions. The validity of the proposed strategy is confirmed by applying the derived models to the parts of the experimental results that were not included in the analyses. Further, the external validation of the models is verified using several statistical criteria recommended by other researchers. The MSGP-based solutions are capable of effectively simulating the nonlinear behavior of the investigated systems. The results of MSGP are found to be more accurate than those of standard GP and artificial neural network-based models.

1. Introduction

Modeling of nonlinear engineering systems can be performed by using different kinds of methods. Owing to the large variety of methods available in this field, no one method can be used a universally applicable solution. The modeling of engineering problems is a difficult task because of the need to estimate both the structure and the parameters of such systems. Different criteria can be characterized for model classification while dealing with a system modeling task [44]. A model can be classified as phenomenological or behavioral [33]. A phenomenological model is derived by considering the physical relationships governing a system. As a result, the structure of the model is selected according to prior knowledge about the system. It is not always possible to design phenomenological models for structural engineering systems because of their complexity. To deal with this issue, behavioral models are commonly employed. These models approximate the relationships between the inputs and outputs on the basis of a measured set of data, without the need for prior knowledge about the mechanisms that produced the experimental data. Behavioral models can provide very good results with minimal effort [33]. Traditional statistical regression techniques are commonly used for behavioral modeling purposes. However, regression

* Corresponding author.
E-mail addresses: a.h.gandomi@gmail.com (A.H. Gandomi), ah_alavi@hotmail.com, am_alavi@civileng.iust.ac.ir (A.H. Alavi).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.07.026

analysis can have large uncertainties. Further, it has major drawbacks in terms of idealization of complex processes, approximation, and averaging widely varying prototype conditions. Regression analysis often assumes linear, or in some cases nonlinear, relationships between the output and the predictor variables; these assumptions do not always hold. Another major constraint in the application of regression analysis is the assumption of normality of residuals. Several alternative computer-aided pattern recognition and data classification approaches have been developed for behavioral modeling. For instance, pattern recognition systems learn adaptively from experience and extract various discriminators. Artificial neural networks (ANNs) are the most widely used pattern recognition procedures. ANNs have been used for a wide range of engineering problems (e.g., [4,32,38,41]). Despite the acceptable performance of ANNs in most cases, they do not usually provide a definite function to calculate the outcome. In addition, ANNs require the structure of the network (e.g. transfer functions, number of hidden layers and neurons, etc.) to be identified a priori. The ANN approach is mostly appropriate to be used as part of a computer program.

Genetic programming (GP) [26] is a new behavioral modeling approach with completely new characteristics. GP is an extension of genetic algorithms (GAs). In general, it may be defined as a supervised machine learning technique that searches a program space instead of a data space. The programs created by traditional GP are represented as tree structures and expressed using a functional programming language [26]. The main advantage of GP-based approaches over regression and ANN techniques is their ability to generate prediction equations without assuming the form of the existing relationships. GP and its variants have been successfully applied to various real world problems (e.g., [1–3,8,9,11,12,17,30,45–48]). Several alternative approaches have been developed to improve the efficiency of the standard GP. Folino et al. [16] and Deschaine et al. [14] combined GP and simulated annealing (SA) to make a hybrid algorithm with better performance. In this hybrid algorithm, the SA strategy is used to decide the acceptance of a new individual. McKee and Lensberg [31] proposed a hybrid approach combining GP and rough sets to construct a bankruptcy prediction model. Brezocnik and Kovacic [7] presented a new integrated GP and GA approach to predict surface roughness in end milling. In this coupled GP and GA approach, GP is used to develop the prediction model and GA is employed to optimize the floating-point constants of the best model. Madar et al. [29] developed a new method coupling GP and orthogonal least squares (OLS) algorithms. This method uses GP to generate nonlinear models of dynamical systems represented as a tree structure. OLS is then employed to estimate the contribution of branches of tree to the model's accuracy. Chan et al. [10] proposed a GP-based fuzzy regression (GP-FR) approach to overcome the deficiencies of the commonly used methods for the modeling of manufacturing processes. In the GP-FR method, GP is used to generate model structures and an FR generator based on fuzzy regression is used to determine outliers in experimental data sets [10]. Ravisankar et al. [36] presented novel ANN and GP hybrids to predict the failure of dotcom companies. In these ANN-GP hybrids, one method is used to perform feature selection in the first phase and the other one is used as a classifier in the second phase. Nasseri et al. [34] developed a combined Extended Kalman Filter (EKF) and GP method for forecasting water demand. In this method, the EKF algorithm is applied to infer latent variables in order to make a forecasting based on GP results of water demand. Lee and Tong [27] proposed a hybrid forecasting model for nonlinear time series by combining autoregressive integrated moving average (ARIMA) with GP.

This study investigates the feasibility of using a multi-stage genetic programming (MSGP) strategy to simulate the complex behavior of engineering systems. This strategy is based on the incorporation of the effects of the predictor variables individually and then the interactions between the variables to derive a more efficient formulation for a problem. The formulation capabilities of the MSGP strategy are demonstrated by applying it to three practical engineering examples. Further, a comparative study is conducted using the results obtained via MSGP and those of the standard GP and its variants. These models are developed using reliable experimental results collected from the literature.

2. Genetic programming

GP is a symbolic optimization technique used to create computer programs for solving a problem following the principle of Darwinian natural selection [17,26]. A breakthrough in GP was made in the late 1980s by performing experiments on symbolic regression. GP was introduced by Koza [26] as an extension of GA. Most of the genetic operators used in GA can be implemented in GP with minor changes. The main difference between GP and GA is the representation of the solution. GP solutions are computer programs that are represented as tree structures and expressed using a functional programming language (such as LISP) [17,26]. GA creates a string of numbers that represents the solution. In other words, in GP, the evolving programs (individuals) are parse trees whose length can vary throughout the run and are not fixed-length binary strings [17,26]. As GP often evolves computer programs, the solutions can be executed without post-processing, whereas coded binary strings typically evolved by GA require post-processing. The traditional optimization techniques such as GA are generally used in parameter optimization to evolve the best values for a given set of model parameters. On the other hand, GP gives the basic structure of the approximation model together with the values of its parameters. GP optimizes a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task. The fitness of each program in a given population is evaluated using a fitness function. Thus, the fitness function is the objective function that GP aims to optimize [15,43].

In GP, a random population of individuals (computer programs) is created to achieve high diversity. A population member in GP is a hierarchically structured tree comprising functions and terminals. The evolved programs are denoted as GP parse trees. The computer programs are also called organisms or chromosomes [17]. The size and form of the programs