Computers and Geotechnics 48 (2013) 82–95

Contents lists available at SciVerse ScienceDirect

Computers and Geotechnics

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

Multivariate adaptive regression splines for analysis of geotechnical engineering systems

W.G. Zhang, A.T.C. Goh *

School of Civil & Environmental Engineering, Nanyang Technological University, Block N1, Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 5 July 2012
Received in revised form 26 September 2012
Accepted 27 September 2012
Available online 12 December 2012

Keywords:
Multivariate adaptive regression splines
Geotechnical system
Nonlinearity
Basis function
Multivariate problem
Neural networks

ABSTRACT

With the rapid increases in processing speed and memory of low-cost computers, it is not surprising that various advanced computational learning tools such as neural networks have been increasingly used for analyzing or modeling highly nonlinear multivariate engineering problems. These algorithms are useful for analyzing many geotechnical problems, particularly those that lack a precise analytical theory or understanding of the phenomena involved. In situations where measured or numerical data are available, neural networks have been shown to offer great promise for mapping the nonlinear interactions (dependency) between the system's inputs and outputs. Unlike most computational tools, in neural networks no predefined mathematical relationship between the dependent and independent variables is required. However, neural networks have been criticized for its long training process since the optimal configuration is not known a priori. This paper explores the use of a fairly simple nonparametric regression algorithm known as multivariate adaptive regression splines (MARS) which has the ability to approximate the relationship between the inputs and outputs, and express the relationship mathematically. The main advantages of MARS are its capacity to produce simple, easy-to-interpret models, its ability to estimate the contributions of the input variables, and its computational efficiency. First the MARS algorithm is described. A number of examples are then presented that explore the generalization capabilities and accuracy of this approach in comparison to the back-propagation neural network algorithm.

1. Introduction

Many geotechnical engineering problems rely on the use of empirical methods expressed in the form of equations or design charts, to determine the response of the system to input variables. This is usually because of an inadequate understanding of the physical phenomena involved in the multivariate problem, or the system is too complex to be described mathematically. A typical example is the determination of the undrained frictional resistance of piles in clay. Based on field load test data, empirical methods have been proposed in which the adhesion is related to the undrained shear strength as well as other factors such as the pile length by an empirical coefficient.

For problems involving several design (input) variables and nonlinear responses, particularly with statistically dependent input variables, regression methods are usually adopted. However, regression models become computationally impractical for problems involving a large number of design variables, particularly when mixed or statistically dependent variables are involved. Another criticism of regression methods lies in their strong model assumptions.

An alternative soft computing technique is the use of artificial neural networks (ANNs). An ANN has a parallel-distributed architecture with a number of interconnected nodes, commonly referred to as neurons. The neurons interact with each other via weighted connections. Each neuron is connected to all the neurons in the next layer. By far the most commonly used ANN model is known as the back-propagation (BP) algorithm [1]. In the BP algorithm, the ANN "learns" the complicated model relationship from examples of input and output patterns through modifying the connection weights to reduce the errors between the actual output values and the target output values. This is carried out by minimizing the defined error function (e.g., sum squared error) using the gradient descent approach. Validation of neural network performance is carried out by "testing" with a separate set of data that was never used in training process, to assess the generalization capability of the trained neural network model to produce the correct input–output mapping.

Generalization is influenced by factors such as the size of the training data, how representative the data is of the problem to be considered, and the physical complexity of the problem. Finding the optimal BP architecture is also important. The BP algorithm has been criticized for its computational inefficiency i.e. long process to determine the optimal network configuration since this is not known a priori. Too few hidden neurons may mean that the net-

* Corresponding author. Tel.: +65 6790 5271.
E-mail address: ctcgoh@ntu.edu.sg (A.T.C. Goh).

0266-352X/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compgeo.2012.09.016

work is unable to model the nonlinear problem correctly. An excessive number of neurons will result in unnecessary arithmetic calculations and high computation cost and may cause a phenomenon called "overfitting", in which the network learns insignificant aspects of the training set i.e. the intrinsic noise in the data. Determining the optimal number of hidden neurons is commonly carried out by a trial-and-error approach through repeatedly increasing the number of hidden neurons till no further improvement in the network performance is obtained. Aside from finding the optimal number of hidden neurons and the number of hidden layers, finding the optimal BP architecture is a difficult task that also involves determining the optimal transfer function and learning rate, as well as the maximum number of training cycles (epochs), all of which require considerable computational effort. Various self-pruning NN algorithms have also been proposed, for example initially starting with a network that is a purposely overfit model, and then trimming it down to the appropriate size. However, neural networks implemented with these algorithms are generally just as computationally intensive since retraining is required each time a hidden neuron or weighted connection is removed.

As highlighted by Shahin et al. [2], ANN has been successfully applied to a number of geotechnical engineering problems including pile capacity, settlement of foundations, soil properties and behavior, liquefaction, site characterization, earth retaining structures, dams, blasting and mining, slope stability, geoenvironmental engineering, rock mechanics, tunneling and underground caverns.

This paper explores the use of another promising procedure known as multivariate adaptive regression spline (MARS) [3] to model nonlinear and multidimensional relationships. As with neural networks, no prior knowledge of the form of the function is required in MARS. The main advantages of MARS are its capacity to find the complex data mapping in high-dimensional data and produce simple, easy-to-interpret models, and its ability to estimate the contributions of the input variables. Previous applications of MARS algorithm in civil engineering include modeling doweled pavement performance, predicting shaft resistance of piles in sand, estimating deformation of asphalt mixtures, analyzing shaking table tests of reinforced soil wall, and determining the undrained shear strength of clay [4–9]. In this paper, a number of examples are presented to demonstrate the function approximating capacity of MARS and its efficiency in a noisy data environment. In addition, comparative performance of the predictions between BP and MARS were carried out for six practical examples in geotechnical engineering.

2. Details of MARS

MARS is a nonlinear and nonparametric regression method that models the nonlinear responses between the inputs and the output of a system by a series of piecewise linear segments (splines) of differing gradients. No specific assumption about the underlying functional relationship between the input variables and the output is required. The end points of the segments are called knots. A knot marks the end of one region of data and the beginning of another. The resulting piecewise curves (known as basis functions), give greater flexibility to the model, allowing for bends, thresholds, and other departures from linear functions.

MARS generates basis functions by searching in a stepwise manner. An adaptive regression algorithm is used for selecting the knot locations. MARS models are constructed in a two-phase procedure. The forward phase adds functions and finds potential knots to improve the performance, resulting in an overfit model. The backward phase involves pruning the least effective terms. An open source code on MARS from Jekabsons [10] is used in carrying out the analyses presented in this paper.

Let $\boldsymbol{y}$ be the target output and $\boldsymbol{X} = (X_1, \ldots, X_P)$ be a matrix of P input variables. Then it is assumed that the data are generated from an unknown "true" model. In case of a continuous response this would be

$$\boldsymbol{y} = f(X_1, \ldots, X_P) + e = f(\boldsymbol{X}) + e \tag{1}$$

in which e is the distribution of the error. MARS approximates the function f by applying basis functions (BFs). BFs are splines (smooth polynomials), including piecewise linear and piecewise cubic functions. For simplicity, only the piecewise linear function is expressed. Piecewise linear functions are of the form $\max(0, x - t)$ with a knot occurring at value t. The equation max(.) means that only the positive part of (.) is used otherwise it is given a zero value. Formally,

$$\max(0, x - t) = \begin{cases} x - t, & \text{if } x \geqslant t \\ 0, & \textit{otherwise} \end{cases} \tag{2}$$

The MARS model $f(\boldsymbol{X})$, is constructed as a linear combination of BFs and their interactions, and is expressed as

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m \lambda_m(X) \tag{3}$$

where each $\lambda_m(X)$ is a basis function. It can be a spline function, or the product of two or more spline functions already contained in the model (higher orders can be used when the data warrants it; for simplicity, at most second-order is assumed in this paper). The coefficients β are constants, estimated using the least-squares method.

Fig. 1 shows a simple example of how MARS would use piecewise linear spline functions to attempt to fit data. The MARS mathematical equation is expressed as

$$y = 4.4668 + 1.1038 * \mathrm{BF1} - 3.997 * \mathrm{BF2} + 1.967 * \mathrm{BF3} \tag{4}$$

where BF1 = max(0, $x - 16$), BF2 = max(0, $16 - x$) and BF3 = max(0, $25 - x$). The knots are located at $x = 16$ and 25. They delimit three intervals where different linear relationships are identified.

The MARS modeling is a data-driven process. To fit the model in Eq. (3), first a forward selection procedure is performed on the

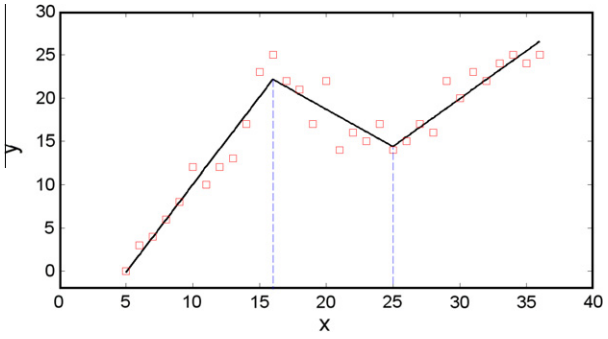

Fig. 1. Knots and linear splines for a simple MARS example.

Table 1
Calculation of error measures.

Measure	Calculation
Coefficient of determination (R^2)	$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_{(i)} - f(x_{(i)}))^2}{\frac{1}{n}\sum_{i=1}^{n}(y_{(i)} - \overline{y})^2}$
Mean Squared Error (MSE)	$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{(i)} - f(x_{(i)}))^2$
Mean Absolute Error (MAE)	$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert y_{(i)} - f(x_{(i)})\rvert$

$\overline{y}$ is the mean of the target values of $y_{(i)}$; $f(x_{(i)})$ is model predictions; n denotes the number of data points in the used set, training or testing set.