Expert Systems With Applications 78 (2017) 347–357

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A regression tree approach using mathematical programming

Lingjian Yang[a], Songsong Liu[a,b], Sophia Tsoka[c], Lazaros G. Papageorgiou[a,*]

[a] Centre for Process Systems Engineerig, Department of Chemical Engineering, University College London, Torrington Place, London WC1E 7JE, UK
[b] School of Management, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, UK
[c] Department of Informatics, King's College London, Strand, London WC2R 2LS, UK

ARTICLE INFO

Article history:
Received 13 July 2016
Revised 4 February 2017
Accepted 5 February 2017
Available online 9 February 2017

Keywords:
Regression analysis
Surrogate model
Regression tree
Mathematical programming
Optimisation

ABSTRACT

Regression analysis is a machine learning approach that aims to accurately predict the value of continuous output variables from certain independent input variables, via automatic estimation of their latent relationship from data. Tree-based regression models are popular in literature due to their flexibility to model higher order non-linearity and great interpretability. Conventionally, regression tree models are trained in a two-stage procedure, i.e. recursive binary partitioning is employed to produce a tree structure, followed by a pruning process of removing insignificant leaves, with the possibility of assigning multivariate functions to terminal leaves to improve generalisation. This work introduces a novel methodology of node partitioning which, in a single optimisation model, simultaneously performs the two tasks of identifying the break-point of a binary split and assignment of multivariate functions to either leaf, thus leading to an efficient regression tree model. Using six real world benchmark problems, we demonstrate that the proposed method consistently outperforms a number of state-of-the-art regression tree models and methods based on other techniques, with an average improvement of 7–60% on the mean absolute errors (MAE) of the predictions.

1. Introduction

In machine learning, regression analysis seeks to estimate the relationships between output variables and a set of independent input variables by automatically learning from a number of curated samples (Sen & Srivastava, 2012). The primary goal of applying a regression analysis is usually to obtain precise prediction of the level of output variables for new samples. Examples of methodologies for regression analysis in the literature include linear regression (Seber & Lee, 2012), automated learning of algebraic models for optimisation (ALAMO) (Cozad, Sahinidis, & Miller, 2014; Zhang & Sahinidis, 2013), support vector regression (SVR) (Smola & Schlkopf, 2004), multilayer perception (MLP) (Hill, Marquez, O'Connor, & Remus, 1994), K-nearest neighbour (KNN) (Korhonen & Kangas, 1997), multivariate adaptive regression splines (MARS) (Friedman, 1991), Kriging (Kleijnen, 2015), and regression tree.

Quite often, one would like to also gain some useful insights into the underlying relationship between the input and output variables, in which case the interpretability of a regression method is also of great interest. Regression tree is a type of the machine learning tools that can satisfy both good prediction accuracy and easy interpretation, and therefore have received extensive attention in the literature. Regression tree uses a tree-like graph or model and is built through an iterative process that splits each node into child nodes by certain rules, unless it is a terminal node that the samples fall into. A regression model is fitted to each terminal node to get the predicted values of the output variables of new samples.

The Classification and Regression Tree (CART) is probably the most well known decision tree learning algorithm in the literature (Breiman, Friedman, Olshen, & Stone, 1984). Given a set of samples, CART identifies one input variable and one break-point, before partitioning the samples into two child nodes. Starting from the entire set of available training samples (root node), recursive binary partition is performed for each node until no further split is possible or a certain terminating criteria is satisfied. At each node, best split is identified by exhaustive search, i.e. all potential splits on each input variable and each break-point are tested, and the one corresponding to the minimum deviations by respectively predicting two child nodes of samples with their mean output variables is selected. After the tree growing procedure, typically an overly large tree is constructed, resulting in lack of model generalisation to unseen samples. A procedure of pruning is employed to remove

* Corresponding author.
E-mail addresses: lingjian.yang.10@ucl.ac.uk (L. Yang), songsong.liu@swansea.ac.uk (S. Liu), sophia.tsoka@kcl.ac.uk (S. Tsoka), l.papageorgiou@ucl.ac.uk (L.G. Papageorgiou).

http://dx.doi.org/10.1016/j.eswa.2017.02.013

sequentially the splits contributing insufficiently to training accuracy. The tree is pruned from the maximal-sized tree all the way back to the root node, resulting in a sequence of candidate trees. Each candidate tree is tested on an independent validation sample set and the one corresponding to the lowest prediction error is selected as the final tree (Breiman, 2001; Wu et al., 2008). Alternatively, the optimal tree structure can be identified via cross validation. After building a tree, an enquiry sample is firstly assigned into one of the terminal leaves (non-splitting leaf nodes) and then predicted with the mean output value of the samples belonging to the leaf node. Despite its simplicity, good interpretation and wide applications (Antipov & Pokryshevskaya, 2012; Bayam, Liebowitz, & Agresti, 2005; Bel, Allard, Laurent, Cheddadi, & Bar-Hen, 2009; Li, Sun, & Wu, 2010; Molinaro, Dudoit, & van der Laan, 2004), the simple rule of predicting with mean values at the terminal leaves often means prediction performance is compromised (Loh, 2011).

The conditional inference tree (ctree) tackles the problem of recursive partitioning in a statistical framework (Hothorn, Hornik, & Zeileis, 2006). For each node, the association between each independent input feature and the output variable is quantified, using permutation test and multiple testing correction. If the strongest association passes a statistical threshold, binary split is performed in that corresponding input variable; otherwise the current node is a terminal node. Ctree is shown to avoid the problem of building biased tree towards input variables with many distinct levels of values while ensuring the similar prediction performance.

Since almost all the tree-based learning models are constructed using recursive partitioning, an efficient yet essentially locally optimal approach, the evtree implements an evolutionary algorithm for learning globally optimal classification and regression trees (Grubinger, Zeileis, & Pfeiffer, 2014), and is considered an alternative to the conventional methods by globally optimising the tree construction. Evtree searches a tree structure that takes into account the accuracy and complexity, defined as the number of terminal leaves. Due to the exponentially growing size of the problem, evolutionary methods are employed to identify a quality feasible solution.

M5', also knows as M5P, is considered as an improved version of CART (Quinlan, 1992; Wang & Witten, 1997). The tree growing process is the same as that of the CART, while several modifications have been introduced in tree pruning process. After the full size tree is produced, a multiple linear regression model is fitted for each node. A metric of model generalisation is defined in the original paper taking into account training error, the numbers of samples and model parameters. The constructed linear regression function for each node is then simplified by removing insignificant input variables using a greedy algorithm in order to achieve locally maximal model generalisation metric. Tree pruning starts from the bottom of the tree and is implemented for each non-leaf nodes. If the parent node offers higher model generalisation than the sum of the two child nodes, then the child nodes are pruned away. When predicting new samples, the value computed at the corresponding terminal node is adjusted by taking into account the other predicted values at the intermediate nodes along the path from the terminal to the root node. The fitting of linear regression functions at leaf nodes improves the prediction accuracy of the regression tree learning model.

M5' is then further extended into Cubist (RuleQuest, 2016), a commercially available rule-based regression model, which has received increasing popularity recently (Kobayashi, Tsend-Ayush, & Tateishi, 2013; Minasny & McBratney, 2008; Moisen et al., 2006; Peng et al., 2015; Rossel & Webster, 2012). M5' is employed to grow a tree first, which is then collapsed into a smaller set of if-then rules by removing and combining paths from the root to the terminal nodes. It is noted here that the if-then rules resulted from Cubist method can be overlapping, i.e. a sample can be assigned into multiple rules, where all the predictions are averaged to produce a final value. This ambiguity decreases the interpretability of the rule model.

The Smoothed and Unsmoothed Piecewise-Polynomial Regression Trees (SUPPORT) is another regression tree learning algorithm, whose foundation is based on statistics (Chaudhuri, Huang, Loh, & Yao, 1994). Given a set of samples, SUPPORT fits a multiple linear regression function and computes the deviation of each sample. The samples with positive deviations and negative deviations are respectively assigned into two classes. For each input variable, SUPPORT compares the distribution of the two classes of samples along this input variable by applying two-sample t test. The input variable corresponding to the lowest P value is selected as splitting node and the average of the two class mean on this splitting variable is taken as break-point.

The Generalised, Unbiased, Interaction Detection and Estimation (GUIDE) adopts similar philosophy as the SUPPORT (Loh, 2002; Loh, He, & Man, 2015). Given a node, the same step of fitting samples with a linear regression model and separating samples into two classes based on the sign of deviations is employed. For each input variable, its numeric values are binned into a number of intervals before a chi-square test is used to determine its level of significance. The most significant input variable is used for binary split. In terms of break-point determination, either a greedy search or median of the two class mean on this splitting variable can be used.

More other variants of the above regression tree models also exist in the literature, including SECRET (Dobra & Gehrke, 2002), MART (Elish, 2009; Friedman, 2002), SMOTI (Malerbao, Esposito, Ceci, & Appice, 2004), MAUVE (Vens & Blockeel, 2006), BART (Chipman, George, & McCulloch, 2010) and SERT (Chen & Hong, 2010), etc.

In the above classic regression tree methodologies, the traditional means of node splitting are dominated by either exhaustively searching the candidate split corresponding to the maximum variance reduction by predicting of mean output values in two child nodes (Breiman et al., 1984; Quinlan, 1992; Wang & Witten, 1997), or examining distribution of sample deviations from fitting one linear regression function to all the samples in the parent node (Chaudhuri et al., 1994; Loh, 2002). However, it is noticed that for those algorithms where terminal leaf nodes are fitted with linear regression functions (Quinlan, 1992; Wang & Witten, 1997), the choice of splitting variable, break-point and regression coefficients are done sequentially, i.e. the splitting variable and break-point are estimated during tree growing procedure while regression coefficients for each child node are computed at pruning step.

A theoretically better node splitting strategy is to simultaneously determine the splitting feature, the position of break-point and the regression coefficients for each child node. In this case, the quality of a split can be directly calculated as the sum of deviations of all samples in either subset. A straightforward exhaustive search algorithm for this problem can be: for each input variable and each break-point, samples are separated into two subsets and one multiple linear regression is fitted for each subset. After examining all possible splits, the optimal split is chosen as the one corresponding to the minimum sum of deviations. The problem with this approach is, however, that as the numbers of samples and input variables grow, the quantity of multiple linear regression functions need to be evaluated increases exponentially, requiring excessive computational time. For example, given a regression problem of 500 samples and 10 input variables, we assume for each input variable, each sample takes a unique value. Then it requires construction of 9980 ($=499 \times 10 \times 2$) multiple linear regression functions in order to find the optimal split for only the root node, which will only become worse as the tree grows larger.