Available online at www.elixirpublishers.com (Elixir International Journal)

Statistics

Elixir Statistics82 (2015) 32419-32424

ISSN: 2229-712X

Algorithmic Modelling of Boosted Regression Trees' Environment's Big Data

M.A.M. Asri[1], N.Z. Yahaya[2], Sabri Ahmad[1], N.A. Ghazali[2] and Asyraf Afthanorhan[3]

[1]School of Informatics, and Applied Mathematics, University Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, Malaysia.
[2]School of Ocean Engineering, University Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, Malaysia.
[2]Faculty of Economics and Management Science, Universiti Sultan Zainal Abidin, 21030 Kuala Terengganu, Terengganu, Malaysia.

ARTICLE INFO

Article history:
Received: 11March 2015;
Received in revised form:
30 April 2015;
Accepted: 6 May 2015;

Keywords
Boosted Regression Trees,
Big Dataset,
Algorithms' Modelling,
Data Simulation,
Meteorological Environment.

ABSTRACT

In tackling with a big dataset, a new and better approach is crucial to be used for. As in this paper, to develop an algorithm modelling for Boosted Regression Trees (BRT), author are decided to use the programming R statistical data analysis tool. The data used in this research, is a one-hour time range of data collected from 2009 up to 2012 for an environment station located at coastal-environment area somewhere in northern of Malaysia. Thus, step by step flowchart from the beginning till the objective been achieve, were provided, and created. Sensitive testing of model been carried out with the three main parameters. Only the number of trees (nt) is to be determine by using the method of estimating the optimal number of iterations; an independent test set (*test*), out-of-bag estimation (*OOB*), and five-fold CV. While the learning rate (lr) and interaction depth (tc) been fixed at 0.001 and 5 respectively. Results indicated that the BRT analysis algorithm best modelled with the best combination of parameters nt of 10000 together with lr and tc that achieves minimum predictive error (minimum error for predictions). Besides, with the boosting output of relative influence plot, and partial dependency plot, the variables significantly influenced Ozone are humidity, ambient temperature, NO, and wind speed with 61.72%, 18.17%, 10.27% and 4.5% respectively. The algorithm model for BRT produced by using the simulated data is best guidance to be used in the field of air pollution specifically. As a matter of fact, the BRT Algorithm can be modelled in varies field with big dataset.

Introduction

De'ath (2000) introduced new and powerful BRT techniques by introducing classification and regression trees to analyse complex ecological data. Nonlinear relationships, high-order interactions, and missing values of data require more flexible and robust methods. De'ath and Fabricius (2000) used classification and regression trees to analyse a survey from the Australian central Great Barrier Reef. The study used S-Plus statistical software along with a library of tree routines which used the recursive partitioning (RPART) for all classification and regression tree analysis. The analysis focused on the relationships between each of the physical variables (sediment, visibility, wave action and slope angle) and spatial variables (cross-shelf position, reef type, within-reef location, depth zone, reef identity) and the response variables, which is the presence–absence of the ratings of abundance of Asterospicularialaurae. It was found that classification and regression trees are powerful tools for the analysis of complex ecological data. According to De'ath (2000) the advantages of the tree and its features include: the flexibility to handle a wide range of response types, rank statistics, which result in invariance of the tree to any monotonic transformations of the explanatory variables, ease and robustness of construction, ease of interpretation of complex results involving interactions, and the ability to handle missing values in both response and explanatory variables.

In his recent work, De'ath (2007) developed a new form of BRT, namely, 'aggregated boosted trees' (ABT), which aim to reduce prediction error relative to boosted trees. ABT comprise a collection of BRTs, each of which is grown on a cross-validation subset of the data. A single regression tree (SRT) and a series of ABTs were fitted to the data.

The results show that the performance of BRTs can be improved using ABTs. Another enhancement is that BRTs typically assume the data are independent; however, BRTs and ABTs can also be adapted to deal with multilevel errors, as are frequently encountered when subsampling of stratified sampling is used. In conclusion, the number of trees per component is consistently predicted more accurately than in the case of a SRT. These approaches were marginally, but consistently better than BRT and also computationally found to be more efficient compared to BRT when applied to the biological field. The most comprehensive and updated application of BRTs was conducted and reported by Elith et al. (2008), who provided a workingguide to BRT. The ensemble method presented for fitting statistical models differs fundamentally from conventional techniques that aim to fit a single ungenerous model. The report explains and provides a working guide to BRT. The work also demonstrates the practicalities and advantages of using BRT as an analytical tool to analyse the distribution of the short-finned eel from native freshwater fish data from New Zealand. From the above discussion, compared to other applications such as GAM, classification and regression trees (CART) and other tree-based models, it is clear that the BRT technique has been proven to be the most effective

Tele:
E-mail addresses:imanasri90@gmail.com

technique to analyse data. Higher accuracies obtained from BRT analysis such as those in the Lawrence (2004) study provides an opportunity to researchers in other areas such as the atmospheric environment to choose BRT as a tool to analyse air pollution data. It is clear from the above that the highly effective learning algorithm in BRT can be applied to other situations and studies.

Methodology

The analysis of boosted regression trees would comply with the dataset that has been prepared. The data is very big as they would be refer as big dataset. The simulation of BRT in R statistical software will be using a one-hour time range of data from a station installed in a coastal-environment area somewhere in Penang. Data used is the secondary data acquired from the Department of Environment (DOE) Ministry of Natural Resources and Environment NRE Malaysia. The analysis and definitely the algorithm modelling is using R statistical software – a programming statistical open-source tool. According to Chambers (2007), the open-source approach can help lead to trustworthy software that is fully accountable and readily accessible, and this is an important component of the aim of this study.

The model was fitted in R 2.15.1 software (R Development Core Team, 2012) using the gbm package version 1.6-3.1 (Ridgeway, 2010).The gbm package was combined with other packages, namely, sp, rJava, raster, dismo, survival, reshape and lattice, gplot and ggplots. For all settings other than those mentioned, we used the defaults in the gbm. Three-model fitting typically needs the specification of three main parameters, namely, the number of trees (nt), the learning rate (lr) and the interaction depth or tree complexity (tc). The use of cross-validation (CV) to select optimal settings is becoming increasingly common (Hastie, 2001), and is led by machine learning, which focuses on predictive success.

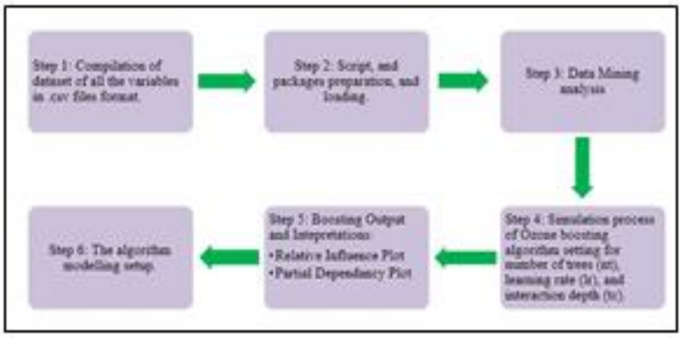

Step 1: Compilation of dataset of all the variables in .csv files format.
Step 2: Script, and packages preparation, and loading.
Step 3: Data Mining analysis
Step 4: Simulation process of Ozone boosting algorithm setting for number of trees (nt), learning rate (lr), and interaction depth (tc).
Step 5: Boosting Output and Interpretations
•Relative Influence Plot
•Partial Dependency Plot
Step 6: The algorithm modelling setup.

Figure 1.1. Flowchart of step-by-step Process for the BRT Algorithm Modelling

Figure 1.1 shows the steps to be taken thoroughly from the beginning of collecting data method up till the analysis and algorithm modelling. The flowchart is adapted and reconstructed to suit author research.

The development of the BRT model involves determining the model algorithm settings of the main model input parameters (learning rate, number of trees and interaction depth) that are fitted using the R software by choosing a 10-fold cross-validation approach, as proposed by Hastie et al. (2001). Three important terminologies in BRT are: learning rate or shrinkage– a shrinkage parameter applied to each tree in the expansion; tree size or number of trees – the total number of trees to fit, which is equivalent to the number of iteration sets for each setting and the number of basic functions in the additive expansion; and interaction depth – the maximum depth of the interactions of variables, where 1 implies an additive model and 2 implies a model with up to two-way interactions.

The process of obtaining the BRT algorithm and the analyses there on is divided into three main stages, which are discussed in which are data compilation and preparation, second the analysis and development of the Ozone boosting algorithm for the dataset and finally the BRT output and interpretations, and algorithm setting.

The gbm offers three methods for estimating the optimal number of iterations or nt after the gbm has been fitted: an independent test set (test), out-of-bag estimation (OOB), and five-fold CV. The optimal number of iterations based on the independent test set method uses a single holdout base dataset, which is similar to Friedman's MART software (Ridgeway, 2007).

According to De'ath (2007), the AdaBoost was examined from a statistical perspective by Friedman et al. (2000) and Hastie et al. (2001). This led to a series of theoretical and practical advances and work on understanding boosting which realised the power and potential of boosting as a general method for approximation based on additive models.

Explanation of Boosted Regression Trees

There are several techniques that aim to improve the performance of a single model by fitting many models and combining them for prediction. A 'decision tree learning' or 'decision trees' is a tool in machine learning and/or data mining which maps observations about a certain item to conclusions about a certain item's target value (Kriegler, 2007). According to Friedman (2001), no matter how dimensionally large the predictor variable space is, or how many variables are used for the prediction, the model subcomponents can be represented by a two-dimensional graphical representation which can be easily plot and interpreted.

Modern decision trees are described statistically by Breidman (1984) and Hastie et al. (2001). The tree-based models partition the predictor space into rectangles by using a series of rules to identify regions having the most homogeneous responses to predictors (Elith et al. 2008). The partitions can be described by a series of if-then statements or they can be visualised by a graph that looks like a tree (Schonlau, 2005). They fit a constant to each region, with classification trees fitting the most probable class as a constant, and regression trees fitting the mean response for observations in that region, assuming normally distributed errors. For example, in Figure 2.2a and 2.2b the two predictor variables, X1 and X2, could be, for example in this study, wind speed and direction, and the response, Y, could be the mean Ozone of the station. Region Y1 and Y2 etc. are the terminal nodes or leaves, and t1 and t2 are split points which are chosen to minimise prediction errors. An effective strategy for fitting a single decision tree is to grow a large tree, then prune it by collapsing the weakest links identified through cross-validation (Hastie, 2001).

It is explained that a regression tree is a piecewise constant or piecewise linear estimate of a regression function, which is constructed by recursively partitioning the data and sampling the data space. The ease of interpretation from two-dimensional plots as shown is a powerful tool for practitioners and has proven to be an appropriate methodology to study air pollution.

In related literature on the theory of boosting, many boosting algorithms have been coded in packages such as the open-source R software, proprietary software called STATISTICA developed by StatSoft, and many more (Carty, 2011). Although the approaches are different, the typical structure of the boosting algorithms is similar.

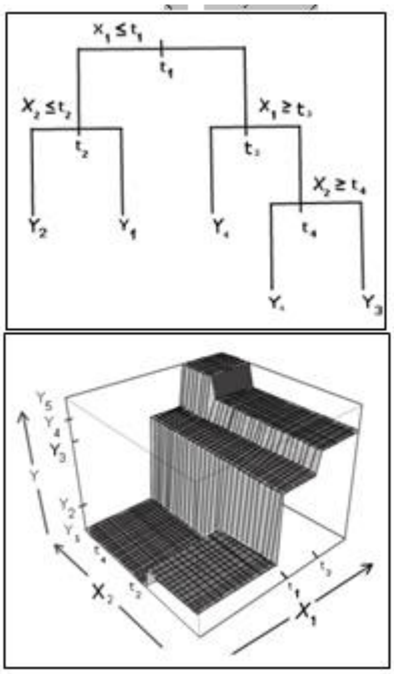


Figure 2.2. A single decision tree, (a) with a response Y, two predictor variables, X_1 and X_2, and split points, t_1, t_2; (b) shows the prediction surface (Hastie, 2001)

Friedman et al. (2000) comments that the bounds and the theory associated with AdaBoost algorithms are interesting, but tend to be too loose to be of practical importance. Later, Friedman_(2001) developed the gradient boosting algorithm, which called for all of the training data observations to be included in the function estimation process at each iteration. The basic structure of the basic boosting algorithm developed by Friedman (2001) can be explained as follows:

1. For all observations, initialise the fitted values to a scalar.
2. For a given loss function, compute the 'error' – the difference between the fitted and actual values of y – for each observation.
3. Fit a model of the error against the predictors.
4. Compute the fitted values of the errors.
5. Compute the 'boosted' fitted values of training data observations by adding the predicted values from the various iterations to the fitted errors from step 4.
6. Repeat steps 2 to 5 a large number of times, each time producing a new vector of predictions.

The Mechanics of Stochastic Gradient Boosting

In 2002, Friedman added a stochastic element to the above boosting algorithm by proposing to take a random sample of observations in each iterations. The performance of gradient boosting was improved by adding an element of randomness to the algorithm and creating the stochastic gradient boosting machine (GBM) or stochastic BRT (Friedman, 2002).This involves taking subsamples of training data; between typically 40–60% in each iteration. This can be achieved by indicating the percentage of the training data in the algorithm. The term 'stochastic gradient boosting' is also simplified to 'gradient boosting' or more simply 'boosted trees'. Hereinafter, the term BRT, will denote 'stochastic gradient boosting' (SGB) using least squares regression trees. Friedman (2002, p. 367) states that "gradient boosting constructs additive regression models by sequentially fitting a simple parameterised function to current 'pseudo'- residuals by least squares at each iteration. In each training data point, the 'pseudo'-residuals are the gradient of the loss function being minimised." Although many approaches exist, the original structure of a boosting algorithm is generally the same. The discussion here will focus on the function approximation, which comes directly from Friedman's SGB algorithm (Friedman, 2002) as follows:

Let $\{\boldsymbol{y}_{i\,,}\boldsymbol{x}_i\}_{\boldsymbol{i}}^{\boldsymbol{N}}$ of known (y, x) values be the entire training data sample and$\{\boldsymbol{\pi}(\boldsymbol{i})\,{}_{,\boldsymbol{i}}^{\boldsymbol{N}}$ be a random permutation of the integers {1,.......*N*}. According to Friedman (2002) the function estimation in BRT has a system of random response variables, *y*, and a set of data input or explanatory variables, *x*. The used of given 'training' data $\{\mathbf{y}_{\mathbf{i}\,,}\mathbf{x}_{\mathbf{i}}\}_{\mathbf{1}}^{\boldsymbol{N}}$ of known (*y*, *x*), is to find a function $\hat{\boldsymbol{f}}$(x) that maps x to y, over the join distribution of to expect the value of some loss function $\boldsymbol{\psi}(\boldsymbol{y},\boldsymbol{f}(\boldsymbol{x}))$ is minimised. Initialise$\hat{\boldsymbol{f}}$(x) to the same constant value across all observations,

$$\hat{\boldsymbol{f}}_{\mathbf{0}}(\boldsymbol{x}) = \left(\begin{smallmatrix}\mathbf{arg}\,\boldsymbol{min}\\ \boldsymbol{F}(\boldsymbol{x})\end{smallmatrix}\right)\boldsymbol{E}_{\boldsymbol{y},\boldsymbol{x}}\,\boldsymbol{\psi}(\boldsymbol{y},\boldsymbol{f}(\boldsymbol{x})). \tag{2.1}$$

Boosting approximates $\hat{\boldsymbol{f}}$(x) by an additive expansion of the form

$$\Phi(\xi) = \textstyle\sum_{\boldsymbol{m}=\mathbf{0}}^{\boldsymbol{M}} \boldsymbol{\beta}_{\boldsymbol{m}}\,\eta(\xi;\boldsymbol{a}_{\boldsymbol{m}}). \tag{2.2}$$

where function h(x ; a) are simple functions of x with parameters a = {a_1, a_2, ...}. In a forward stage-wise manner, the expansion coefficients $\boldsymbol{\beta}_{\boldsymbol{m}}$ and the parameters $\boldsymbol{a}_{\boldsymbol{m}}$ are jointly fit to the training data sets (Carty, 2011 p 22).

First, one starts with a preliminary guess, F_o(x), and then for *m* = 1, 2, ...,*M*to the following is done:

$$(\boldsymbol{\beta}_{\boldsymbol{m}}, \boldsymbol{a}_{\boldsymbol{m}}) = \overset{\mathbf{arg}\,\boldsymbol{min}}{\boldsymbol{\beta},\boldsymbol{a}} \textstyle\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{\psi}\left(\boldsymbol{y}_{\boldsymbol{i},}\boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}\boldsymbol{x}_{\boldsymbol{i}}\right) + \boldsymbol{\beta}\boldsymbol{h}(\mathbf{x};\mathbf{a})). \tag{2.3}$$

and

$$\boldsymbol{F}_{\boldsymbol{m}}(\xi) = \boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}\,(\xi) + \boldsymbol{\beta}_{\boldsymbol{m}}\,\eta(\xi;\boldsymbol{a}_{\boldsymbol{m}}). \tag{2.4}$$

According to Friedman (2001), gradient boosting approximation solves (4.3) for arbitrary differentiable loss function $\boldsymbol{\psi}(\boldsymbol{y},\boldsymbol{f}(\boldsymbol{x}))$ with two-step procedure as follows:

First, by fitting the function of h(x ; a) using least squares

$$\boldsymbol{a}_{\boldsymbol{m}} = \overset{\mathbf{arg}\,\boldsymbol{min}}{\boldsymbol{a},\boldsymbol{\rho}} \textstyle\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}}[\tilde{\boldsymbol{y}}_{\boldsymbol{im}}\square\,\boldsymbol{\rho}\boldsymbol{h}(\boldsymbol{x}_{\boldsymbol{i}};\alpha]^2. \tag{2.5}$$

and to the current 'pseudo'-residuals

$$\tilde{\boldsymbol{y}}_{\boldsymbol{im}} = -\left[\frac{\boldsymbol{\partial}\boldsymbol{\varphi}(\boldsymbol{y}_{\boldsymbol{i},}\boldsymbol{F}(\boldsymbol{x}_{\boldsymbol{i}}))}{\boldsymbol{\partial}\boldsymbol{F}(\boldsymbol{x}_{\boldsymbol{i}})}\right]_{\boldsymbol{F}(\boldsymbol{x})\infty\,\boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}(\boldsymbol{x})} . \tag{2.6}$$

Second, to calculate the optimal value for the coefficient $\boldsymbol{\beta}_{\boldsymbol{m}}$ from given *h*(x;$\boldsymbol{a}_{\boldsymbol{m}}$)

$$\boldsymbol{\beta}_{\boldsymbol{m}} = \overset{\mathbf{arg}\,\boldsymbol{min}}{\boldsymbol{\beta}} \textstyle\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{\psi}\,(\boldsymbol{y}_{\boldsymbol{i}}, \boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}\,(\boldsymbol{x}_{\boldsymbol{i}}) + \boldsymbol{\beta}\boldsymbol{h}(\boldsymbol{x};\boldsymbol{a}_{\boldsymbol{m}})). \tag{2.7}$$

This approach has been applied to the case where the base learner h(x ; a) is an L terminal node regression tree. Friedman (2002) also states that at each iteration, *m*, a regression tree partitions the *x* space into *L*- disjoint regions,$\{\boldsymbol{R}_{\boldsymbol{lm}}\}_{\boldsymbol{l}=\mathbf{1}}^{\boldsymbol{L}}$, and predicts a separate constant value in each one as

$$\mathrm{H}(\xi; \{\boldsymbol{R}_{\boldsymbol{lm}}\}_{\boldsymbol{l}=\mathbf{1}}^{\boldsymbol{L}}) = \textstyle\sum_{\boldsymbol{l}=\mathbf{1}}^{\boldsymbol{L}}[\bar{\boldsymbol{y}}_{\boldsymbol{lm}}\ 1(\xi \in\ \boldsymbol{R}_{\boldsymbol{lm}})]. \tag{2.8}$$

where$\bar{\boldsymbol{y}}_{\boldsymbol{lm}}$ = mean $_{\mathrm{xi}\epsilon Rlm}$($\bar{\boldsymbol{y}}_{\boldsymbol{lm}}$) is the mean of (4.18) in each separate region R_{lm}.

The tree as calculated in (4.20) predicts a separate constant value of $\bar{\boldsymbol{y}}_{\boldsymbol{lm}}$ within each region $R_{lm,}$ and the solution to (4.19) diminishes to a simple 'location' estimate based on $\boldsymbol{\psi}$:

$$\boldsymbol{\gamma}_{\boldsymbol{lm}} = \overset{\mathbf{arg}\,\boldsymbol{min}}{\gamma} \textstyle\sum_{\boldsymbol{x}_{\boldsymbol{i}}\,\in\boldsymbol{R}_{\boldsymbol{lm}}} \boldsymbol{\psi}\,(\boldsymbol{y}_{\boldsymbol{i}}, \boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}\,(\boldsymbol{x}_{\boldsymbol{i}}) + \boldsymbol{\gamma}\,). \tag{2.9}$$

Friedman (2002) also clarifies that the existing approximation,$\boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}$(x), is separated and updates in each region:

$$\boldsymbol{F}_{\boldsymbol{m}}(\xi) = \boldsymbol{F}_{\boldsymbol{m}-\mathbf{1}}(\xi) + \boldsymbol{\nu} * \boldsymbol{\gamma}_{\boldsymbol{lm}}1(\xi\boldsymbol{\epsilon}\boldsymbol{R}_{\boldsymbol{lm}}). \tag{2.10}$$

where $\boldsymbol{\nu}$ is the 'shrinkage' parameter that ranges between 0 < $\boldsymbol{\nu}$ ≤1. The role of shrinkage in the context of boosting is to scale the contribution of each tree by a factor of $\boldsymbol{\nu}$ when it is added to the current approximation (Hastie, 2009).

Shrinkage or regularisation is an important feature that has an impact on each additional tree and is used in an effort to avoid over fitting. Therefore Friedman's boosting algorithm includes *shrinkage*, λ in the iterations to control the gradient's steps, so that a more optimal solution is not overshot. In other words, λ tends to mitigate over fitting, in which the only trade-off is computational time. It is important to note that smaller values of shrinkage always give improved predictive performance. The shrinkage parameter 0 < *v*< 1 controls the learning rate of the procedure. Smaller shrinkage values require a larger number of iterations. Friedman (2000) found that small values (≤ 0.1) lead to much better generalisation error. For a reasonable computational time, the number of iterations is usually 3,000–10,000 with a shrinkage rate between 0.01 and 0.001 (Ridgeway, 2007).

Friedman (2002) made a minor modification to the gradient boosting algorithm to include randomness as an essential part of the boosting procedure, which led to the introduction of the SGB algorithm. The difference in SGB algorithm is that in each iteration a subsample of the training data is selected at random and without replacement from the full training data set.

The stochastic procedure as applied to the gradient boosting algorithm is described as follows: Let $\{\boldsymbol{y}_{i\,,}\boldsymbol{x}_i\}_i^{N}$ of known (y,x) values be the entire training data sample and$\{\boldsymbol{\pi}(\boldsymbol{i})\,_{,i}^{N}$be a random permutation of the integers {1,.......N}. Then a random subsample of size $\widetilde{\boldsymbol{N}}$<*N* is given by $\{\boldsymbol{y}_{\pi(i)\,,}\boldsymbol{x}_{\pi(i)}\}_i^{\widetilde{N}}$. The SGB algorithm is then:

1. $\hat{f}_0(x) = \arg min_\gamma \sum_{1=1}^{N} \psi\,(y_i,\gamma).$ (2.11)
2. $Form = 1\ toMdo$:

$\{\pi(i)\}_1^N$ = rand_perm $\{i\}_1^N$. (2.12)

$\tilde{y}_{im}$ = - $[\frac{\partial\varphi(y_{i,}F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$, l = 1, $\widetilde{N}$. (2.13)

$\{R_{lm}\}_1^L$) = L – terminal node tree ($\{\tilde{y}_{\pi(i)m},x_{\pi(i)m}\}_1^{\widetilde{N}}$ (2.14)

$\boldsymbol{\gamma_{lm}}$ **= arg min**$_\gamma \sum_{x_{\pi(i)\in R_{lm}}} \boldsymbol{\psi}\left(\boldsymbol{y}_{\pi(i)}, \boldsymbol{F}_{m-1}\,(\boldsymbol{x}_{\pi(i)}) + \boldsymbol{\gamma}\right)$ (2.15)

F_m (x) = F_{m-1}(x) + $\upsilon.\gamma_{lm}$ 1(x∈ R_{lm}). (2.16)

3. End For
4. End Algorithm

According to Friedman (2002), the smaller the fraction *f* = $\widetilde{\boldsymbol{N}}$/*N*, the more the random samples used in successive iterations will differ. Making the value of *f* smaller reduces the amount of data available to train the base learner at each iteration. This condition causes the variance associated with the individual base learner estimates to increase.

The BRT is controlled through a 'bag fraction' that specifies the proportion of data to be selected at each step. The default bag fraction is 0.5, meaning that, at each iteration, 50% of the data are drawn at random, without replacement, from the full training set. Optimal bag fractions can be established by comparing predictive performance and model-to-model variability under different bag fractions. Study by Elith et al. (2008) shows that stochasticity improves model performance, and fractions in the range 0.5–0.75 have given the best results for the responses. This additional feature of the algorithm results in the marked improvement of predictions from the BRT model. There are five tuning parameters that need to be controlled (in addition to the distribution): the training sample size relative to the training population (*bag.fraction*), the number of iterations (*nr*), the learning rate (*lr*), the maximum tree depth (*interaction depth*), and the number of observations in each terminal node. For each iteration, only the random subset of the residuals is used to build the tree. The in-bag fraction signifies the fraction of observations in the training data to sample for each iteration. The fraction of the training set observations is randomly selected to propose the next tree in the expansion.

Partial Dependencies Function and Plots

Friedman (2001) indicates that visualisation is one of the most powerful interpretational tools for graphical renderings of the value of the derived approximation ($\hat{\boldsymbol{F}}$(*x*)). Considering the predictors, *x*, the variables that are used in BRT model fitting are represented as '*x*' while the response variables are represented as '*y*'. The interpretation of BRT output can be graphically presented in three different ways, namely, relative influence variables, partial dependence plots between variables and two-way variables interactions, all of which will be discussed in the next sections.

Discussion

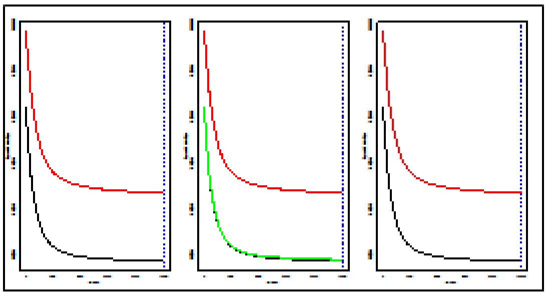


Figure 3.1. The methods for estimating the optimal number of iterations with (a) OOB, (b) CV, and (c) Test.

Author has selected the method of "CV" in gbm to obtain an estimate of the optimal number of boosting iterations for a gbm object and to plot performance measures (Ridgeway, 2010). These can be determined by the number of iterations using the test set, which can be found when the iterations achieve minimum predictive error (minimum error for predictions for independent samples), as indicated by Elith et al.(2008). First, the prediction performance of the model was assessed by estimating the optimal number of boosting iterations for a gbm object by obtaining the boosting model with minimum predictor error using the "CV" method (De'ath 2007; Elith et al., 2008; Ridgeway, 2010) and proven graphically by an error plot, which shows the minimum error. The best number of iterations for "CV" with a minimum square error from a five-fold CV method was performed by using the gbm package in R software to indicate the best iteration for the number of tree. The number of trees is 1000.

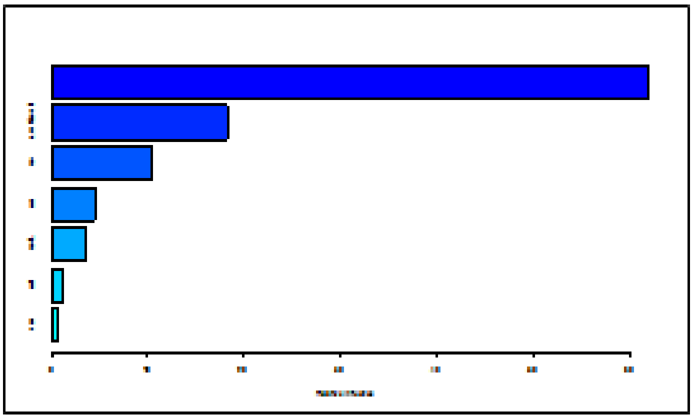


Figure 3.2. Relative Influence of Significant Variables

As illustrated in Figure 3.2, the most influence variable to Ozone is humidity with 61.72%. While the second most influence would be ambient or temperature with 18.17%. The least significant influence to Ozone is NOX with at most of 0.73% only

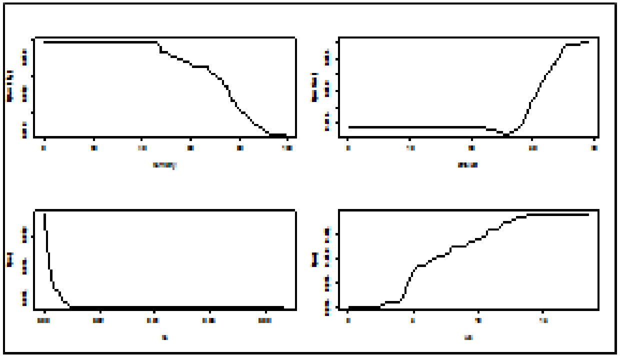
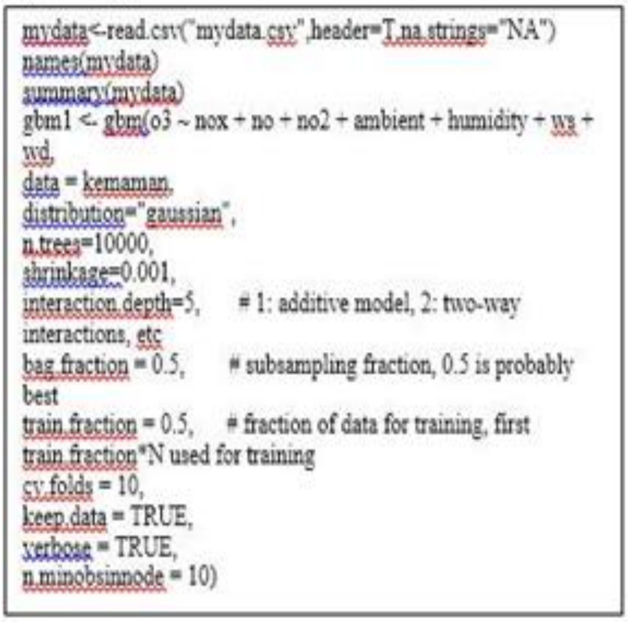


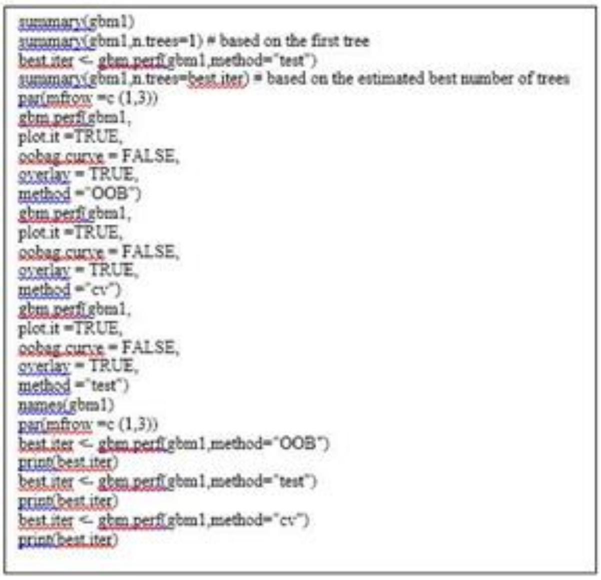
Figure 3.3. Partial Dependence Plot of Most Influenced Variables.

Figure 3.displays the partial dependence function performed on four fitted variables from the sample of data. This is a single-variable partial dependence plots on a few of the important predictor variables for the first randomly generated function used in the simulation study. As we can see, unlike most other approximation methods, there is no explicit smoothness constraint imposed upon the boosted model.



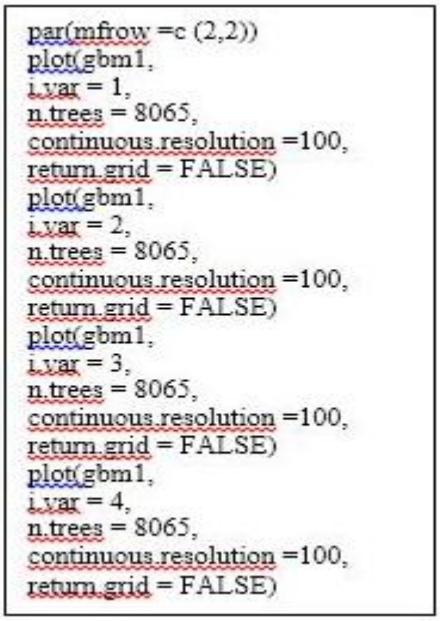
```
mydata<-read.csv("mydata.csv",header=T,na.strings="NA")
names(mydata)
summary(mydata)
gbm1 <- gbm(o3 ~ nox + no + no2 + ambient + humidity + ws +
wd,
data = kemaman,
distribution="gaussian",
n.trees=10000,
shrinkage=0.001,
interaction.depth=5,      # 1: additive model, 2: two-way
interactions, etc
bag.fraction = 0.5,       # subsampling fraction, 0.5 is probably
best
train.fraction = 0.5,     # fraction of data for training, first
train.fraction*N used for training
cv.folds = 10,
keep.data = TRUE,
verbose = TRUE,
n.minobsinnode = 10)
```

Figure 3.4. The Algorithm Setting for BRT



```
summary(gbm1)
summary(gbm1,n.trees=1) # based on the first tree
best.iter <- gbm.perf(gbm1,method="test")
summary(gbm1,n.trees=best.iter) # based on the estimated best number of trees
par(mfrow =c (1,3))
gbm.perf(gbm1,
plot.it =TRUE,
oobag.curve = FALSE,
overlay = TRUE,
method ="OOB")
gbm.perf(gbm1,
plot.it =TRUE,
oobag.curve = FALSE,
overlay = TRUE,
method ="cv")
gbm.perf(gbm1,
plot.it =TRUE,
oobag.curve = FALSE,
overlay = TRUE,
method ="test")
names(gbm1)
par(mfrow =c (1,3))
best.iter <- gbm.perf(gbm1,method="OOB")
print(best.iter)
best.iter <- gbm.perf(gbm1,method="test")
print(best.iter)
best.iter <- gbm.perf(gbm1,method="cv")
print(best.iter)
```

Figure 3.5. The Algorithm Setting for BRT (Determination of Number of Tree)



```
par(mfrow =c (2,2))
plot(gbm1,
i.var = 1,
n.trees = 8065,
continuous.resolution =100,
return.grid = FALSE)
plot(gbm1,
i.var = 2,
n.trees = 8065,
continuous.resolution =100,
return.grid = FALSE)
plot(gbm1,
i.var = 3,
n.trees = 8065,
continuous.resolution =100,
return.grid = FALSE)
plot(gbm1,
i.var = 4,
n.trees = 8065,
continuous.resolution =100,
return.grid = FALSE)
```

Figure 3.6. The Algorithm for Partial Dependency Plot

Conclusion and Acknowledgement

We have been fortunate to work with and learn from each other. So grateful to so many people who have discussed the theme of big data, as the term even have been popular long ago. As well as the Boosted Regression Trees, an approach of methodology in focus to analyse the cross-sectional study to handle big data. Author would like to express special gratitude to Assoc. Prof. Dr. Sabri Bin Ahmad for his tutoring, from the beginning until this recent time. Not to mention, millions of thanks would go to Dr. Noor Zaitun Binti Yahaya (AHEA, UK) for her determination and guidance to author in order to complete the boosted regression trees methodology understanding. This would be the reason for her to be the author's co-supervisor. This short paper, not yet to complete is the fundamental step in setting of boosted regression trees algorithmically.

Reference

1. BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A, & STONE. 1984. Classification and Regression Trees. *Wadsworth International Group*. Belmore, California, USA.
2. BREIMAN, L. 1999. Using adaptive bagging to de bias regressions, Technical Report 547, Statistics Dept. UCB.
3. CARSLAW, D. C. & TAYLOR, P. J. 2009. Analysis of air pollution data at a mixed source location using boosted regression trees. *Atmospheric Environment*, 43**,** 3563-3570.
4. CARSLAW, D.C AND ROPKINS. 2012. Open-air R-package for air quality data analysis. *Env. Modelling & Software*. Vol. 27-28.
5. CARTY, D, M. 2011. An analysis of boosted regression trees to predict the strength properties of wood composites. Master's thesis, University of Tennessee, http://trace.tennessee.edu/utk_gradthes/954
6. CHAMBERS 2007. Software for Data Analysis: Programming with R. New York: Springer.
7. DE'ATH, G. & FABRICIUS, K. E. 2000. Classification and Regression Trees: A Powerful Yet Simple Technique for Ecological Data Analysis. *Ecology*, 81**,** 3178-3192.
8. DE'ATH, G. 2002. Multivariate Regression Trees: A New Technique for Modeling Species-Environment Relationships. *Ecology*, 83**,** 1105-1117.

9. DE'ATH, G. 2007. Boosted Trees for Ecological Modeling and Prediction. *Ecology*, 88**,** 243-251.

10. DE'ATH, G. & FABRICIUS, K. E. 2000. Classification and Regression Trees: A Powerful yet Simple Technique for Ecological Data Analysis. *Ecology,* 81**,** 3178-3192.

11. ELITH, J., LEATHWICK, J. R. & HASTIE, T. 2008. A working guide to boosted regression trees. *Journal of Animal Ecology,* 77**,** 802-813.

12. FRIEDMAN, J.H. 1999. Greedy Function Approximation: A Gradient Boosting Machine. Technical Report, Department of Statistics, Stanford University, USA.

13. FRIEDMAN, J.H., HASTIE, T., TIBSHIRANI, R. 2000. Additive logistic regression a statistical view of boosting. *Annals. Of Statistics*. 28(2): 337-374.

14. FRIEDMAN, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics,* 29**,** 1189-1232.

15. FRIEDMAN, J. H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis,* 38**,** 367-378.

16. FRIEDMAN, J.H, POPESCU, B.E. 2005. Predicting learning via rule ensembles. Technical Report. Stanford University. Online: http://.stat.stanford.edu .California, USA.

17. FRIEDMAN, J. H. & POPESCU, B. E. 2008. Predictive learning via rule ensemble. *Annals of Applied Statistics,* 2**,** 916.

18. FREUND, Y. & SCHAPIRE, R.E. 1996. Experiments with a new boosting algorithm. In machine learning: Proceedings of the 13th International Conference 148-156. San Francisco, USA.

19. GARDNER, M. W. & DORLING, S. R. 2000. Statistical surface ozone models: an improved methodology to account for non-linear behaviour. *Atmospheric Environment,* 34**,** 21-34.

20. GHAZALI, N.A., RAMLI, N.A., YAHAYA, A.S. 2009. A Study to investigate and Model the Transformation of Nitrogen Dioxide into Ozone Using Time Series Plot. *European Journal of scientific Research* 37(2), p. 192-205.

21. GHAZALI N. A., RAMLI N. A., YAHAYA A. S., YUSOF N. F., SANSUDDIN N., MADHOUN W. A. 2010. Transformation of nitrogen dioxide into ozone and prediction of ozone concentrations using multiple linear regression techniques. *Environmental Monitoring & Assessment 165(1)* p. 475-489. *Doi: 10.1007/s10661-009-0960-3*.

22. HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. 2001. *The Elements of Statistical Learning: Data mining, Inference, and Prediction. 2nd edition.* Springer-Verlag, New York.

23. HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. 2009. *The Elements of Statistical Learning: Data mining, Inference, and Prediction. 2nd edition.* Springer-Verlag, New York.

24. KRIEGLER, B. 2007 Cost-sensitive Stochastic Gradient Boosting within a Quantitative Regression Framework. *PhD Thesis.* University of California, LA.

25. KRIEGLER, B. AND BERK, R. 2007. Estimating the homeless population in Los Angeles: An Application of cost-sensitive Stochastic Gradient Boosting. *Department of Statistic series.* University of California.

26. LAWRENCE, R., BUNN, A., POWELL, S. & ZAMBON, M. 2004. Classification of remotely sensed imagery using stochastic gradient boosting as a refinement of classification tree analysis. *Remote Sensing of Environment,* 90**,** 331-336.

27. LEATHWICK, J. R., ELITH, J., FRANCIS, M. P., HASTIE, T. & TAYLOR, P. 2006. Variation in demersal fish species richness in the oceans surrounding New Zealand: an analysis using boosted regression trees. *Marine Ecology Progress Series,* 321**,** 267-281.

28. MOISEN, G. G., FREEMAN, E. A., BLACKARD, J. A., FRESCINO, T. S., ZIMMERMANN, N. E. & EDWARDS JR, T. C. 2006. Predicting tree species presence and basal area in Utah: A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods. *Ecological Modelling,* 199**,** 176-187.

29. R DEVELOPMENT GROUP. 2008. R: A Language and environment for Statistical Computing. *In:* COMPUTING., R. F. F. S. (ed.). Vienna, Austria.

30. RIDGEWAY, G. 1999. The State of Boosting. *Journal of Computing Sciences and Statistics,* 31**,** 171-181.

31. RIDGEWAY, G. 2007. Generalized Boosted Models: A guide to the *gbm* package.

32. RIDGEWAY, G. 2010. GBM: Generalized Boosted Regression Models. . *R Package Version 1.6-3.1.*

33. SCHAPIRE, R.E. 1990. The strength of weak learnability. *Machine Learning*, 5(2):197–227.

34. SCHONLAU, M. 2005. Boosted Regression (Boosting): An Introductory tutorial and a stata plugin. *The Stata Journal,* 5**,** 330-354.

35. SCHONLAU, M. 2005. Boosted Regression (Boosting): An Introductory tutorial and a stata plugin. *The Stata Journal,* 5**,** 330-354.

36. WESTMORELAND, E. J., CARSLAW, N., CARSLAW, D. C., GILLAH, A. & BATES, E. 2007. Analysis of air quality within a street canyon using statistical and dispersion modelling techniques. *Atmospheric Environment,* 41**,** 9195-9205.

37. WORLD HEALTH ORGANIZATION (WHO). 2005. WHO air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide. *World Health Organization 2006*.

38. YAHAYA, N. Z., TATE, J. E. & TIGHT, M. R. 2010a. The Spatial and temporal Variation of PNC around urban traffic junction. *Annual Aerosol Science Conference,The Aerosol Society UK.* University of Southampton, Southampton, United Kingdom.

39. YAHAYA, N. Z., TATE, J. E. & TIGHT, M. R. 2010b. The Influence of Traffic Flow, Synoptic and In-street Winds on Particle Number Concentrations (PNC) around an Urban Intersection. *In: The 15th Int. Union of Air Pollution Prevention and Env. Protection Assoc.* World Clean Air Congress, 9 - 16 September 2010. Vancouver, British Columbia, Canada.

40. YAHAYA, N. Z., TATE, J. E. & TIGHT, M. R. 2011. Analysing Roadside Particle Number Concentrations using Boosted Regression Trees (BRT). *In: The European Aerosol Conference,* EAC 2011, 5 9 September 2011. The University of Manchester, Manchester, United Kingdom. 22.

41. YOUNG, T, M. 2007. Parametric and non-parametric regression tree models of the strength properties of engineered wood panels using real-time industrial data. Doctor of Philosophy degree, The University of Tennessee, Knoxville, US.