Research & Reviews: Journal of Statistics and Mathematical Sciences

Application of Gradient Boosting Algorithm In Statistical Modelling

Hamid A Adamu[1]*, Murtala Muhammad[2] and Abdullahi Mohammed Jingi[2]

[1]School of Computing Science and Engineering, University of Salford, Manchester, UK

[2]Department of Computer Science Adamawa State University (ADSU), Mubi Mubi, Adamawa State Nigeria

Research Article

Received date: 17/12/2019

Accepted date: 01/01/2019

Published date: 08/01/2019

***For Correspondence**

School of Computing Science and Engineering, University of Salford, Manchester, UK.

E-mail: modibbokomsiri45@gmail.com

Keywords: GLM, GAM, Gradient boosting and Cross-validation

ABSTRACT

Gradient Boosting (GB) is a machine learning technique for regression, which produces more accurate prediction models in form of ensemble weak prediction models. It is an iterative algorithm that combines simple parameterized functions with weak performance (high prediction error) in order to produce a highly accurate prediction by minimizing the errors [1]. Therefore, this paper investigates the application of gradient boosting algorithm in Generalised Linear Model (GLM) and Generalised Additive Models (GAM) to produce better prediction using Munich rental data. More interestingly, to compare the performance of classical GLM and GAM and their corresponding boosted packages in prediction. However, in boosting algorithm, optimum-boosting iterations are highly recommended to avoid over fitting. It plays an important role when the fitting model, we, therefore, employ the use of the Akaike Information Criterion (AIC) based technique to determine the appropriate boosting iteration that gives the optimum prediction. We applied the AIC and Cross-validation (CV) techniques to determine the optimum boosting iterations. The results obtained are then compared to investigate the algorithm that is more accurate. It is noticed that by default, the gamboost (boosted GAM) fits models using smooth base-learners (bbs). Similarly, it also noted that the coefficients of the fitted model will be in matrix form if smooth base-learners are used while they are just linear if linear base-learners are used.

INTRODUCTION

Statistical modeling is a process of constructing parsimonious statistical models to aid in understanding the matter in question and Regression analysis is regarded as the most popular and recommended statistical techniques for modeling the relationship between a response variable and explanatory variables [2]. It is basic and well known that every statistical model deals with uncertainties and therefore contains a random part that aims at describing those uncertainties that exist in the matter.

Therefore, to model these uncertainties, probability distributions are widely used and highly recommended. Statistical modeling was introduced in the early 1970s by Nelder and Wedderburn when they initiated the unifying concept of the generalized linear model which models only the conditional mean of the response variable while treating other distributional parameters as constants. Meanwhile, Boosting has its origin from a theoretical framework for studying machine learning called the PAC (Probably Approximately Correct) learning technique by (Valiant, 1984) which was later explained clearly by Kearns J. M and Vazirani V. U in 1985. They are the first in the history to raise an observation of whether a “Weak” learning algorithm which performs slightly better than random guessing can be “Boosted” in order to produce strong learning algorithm [3]. The PAC was improved by Schapire [4] where he came up with the first boosting algorithm in 1989.

However, Freud [5] then published a paper where he improved the efficiency of the boosting algorithm. The most popular boosting algorithm is called Adaboost or adaptive boosting and was introduced by Freud and Schapire [6]. The algorithm can be applied effectively in prediction/estimation. Other applications of the ad boost include; face detection.

Meanwhile, boosting which was originally developed for classification problems had been recently extended to regression models by [7]. It is a machine language technique for regression that produces a proper prediction model in the form of an ensemble of many weak prediction models. Thus, boosting is a technique for improving the accuracy of predictive functions by applying the function repeatedly in series and combining the result of each function with weighting in order to minimize the prediction error. It is a committee-based approach and other related techniques include; bagging, stacking, and model averaging [8]. But boosting is unique because it builds the model in sequence (Forward stage-wise procedure) and outstands all these techniques. Moreover, the algorithm combines many weak classifiers in order to produce a strong committee-based technique that gives the best result than the individual weak classifiers.

This paper applies a gradient boosting algorithm to GLM and GAM to enhance the prediction of the net rent of rooms (per square meter). The algorithm is applied in order to minimize the errors which might be encountered during the prediction to provide a more accurate and reliable estimation.

GENERALIZED LINEAR MODELS (GLMs)

Generalized Linear Models (GLMs) are the generalization of the linear regression which allows not only response variables that are normally distributed but it also manage with the responses that are not normal. GLM is a large class of statistical models for relating responses to linear combinations of predictor variables.

GENERALIZED ADDITIVE MODELS (GAMs)

Generalized Additive Models (GAMs) were introduced by Hastie and Tibshirani as an extension and improvement of the generalized linear models (GLMs). In GLM, the data (y_1,...,i_n) is assumed to be generated independently from an exponential family with the expectation (E[y]=μ). But for the case of GAMs, instead of the linear relationship, the predictor γ_i is the additive function of the covariates (x_{i1},...,x_{ip}).

$$\gamma_i = f_i\left(x_{i1}\right)+,.....,+f_p\left(x_{ip}\right)$$

Where f_1,...,f_p are unknown functions, usually selected to be smoothers.

PROBLEM STATEMENT

Boosting can be explained from an optimization viewpoint, as a forward stage-wise optimization classification/regression method for generating a strong ensemble of weak base learners to obtain a strong one which gives an optimal prediction. It proved to be an effective algorithm both theoretically and empirically in improving the performance of the base classifiers. The outcomes of predictions are mostly faced with shortcomings (especially for high dimensional data) due to the increase in errors during the prediction which leads to poor performance of the prediction. Therefore, this paper tends to investigate and implement the gradient boosting algorithm in R software using GLM and GAM packages to reduce the error in the prediction in order to produce an optimal estimation.

LITERATURE REVIEW

A considerable amount of literature has been published recently by various researchers on Boosting because of its effectiveness, reliability, accuracy, importance and diverse applications. It attracts the attention of different researchers. Friedman [9] highlighted that a general gradient descent (boosting) algorithm is developed for additive expansions based on any kind of fitting criterion. Meanwhile, specific algorithms have also been presented for least squares, least absolute deviation and Hubber loss function for regression, whereas, the multi-class logistic likelihood for classification. In addition, Ye [10] investigated the Stochastic Gradient Boosted Decision Trees (GBDT) where they identified that it's one of the most widely used learning algorithms in machine language in recent years. They further revealed that the algorithm is easy to interpret, adaptable, effective, and reliable and produces highly accurate models. However, most implementations today are computationally expensive and require all training data to be in main memory.

According to Culp [11], stochastic gradient boosting provides an improvement, which incorporates a random mechanism at each boosting step showing development and high quality in performance with speed in generating the ensemble. They further verified that boosting has proved to be an effective algorithm both theoretically and empirically in improving the performance of the base classifiers. Moreover, they showed how a package implements the AdaBoost algorithm, using exponential and logistic loss functions for classification problems. In addition, the package also allows the implementation of regularized versions of the method by using the learning rate as a tuning parameter, which leads to improved computational performance. A contemporary study by Zhang, [12] proves the flexibility of the gradient boosting algorithm where they investigated its use in two applications, first they showed the advantage of loss functions that were developed particularly for optimizing application objectives and then extend the original gradient boosting algorithm with Newton-Raphson method to speed up learning. They further presented that the algorithm is useful in determining regions of high Word Error Rate (WER) and yields up to 20% relative improvement depending on the selected operating point. Moreover, Mayer [13] published a paper that described a boosting algorithm for high dimensional data on GAMLSSs to overcome the limitations of current fitting procedures which include infeasibility for high dimensional data. It is specifically designed to allow the simultaneous estimation of predictor effects and variable selection. To verify the algorithm in a real-world problem, they applied it to Munich rental guide data, which are used by landlords and tenants as a reference for the average rent of a flat depending on its characteristics and spatial features.

Consequently, Ridgway [14] investigated the applications of boosting where he identified that it is one of the algorithms that has shown great promise. He further reported that empirical studies have shown that combining models using the boosting algorithm produces more accurate classification and regression models. In addition, this algorithm can also be extended to the exponential family as well as proportional hazards regression models.

However, Schapire [15] showed that this procedure is proved to improve the accuracy of the weak learner alone. The term 'boosting' refers to the act of boosting the accuracy of a classification hypothesis. Meanwhile, Friedman et al., verified that AdaBoost is approximately equivalent to fitting GAMs for the case of a two-state classifier with covariates. Niu [16] Identified that AdaBoost algorithm can be fitted in a gradient descent optimization framework- which is highly important for analyzing the algorithms' procedure. Meanwhile, from the optimization viewpoint, boosting is a forward stage-wise optimization classification margin. They further verified that different Cost-Sensitive Boosting (CSB) algorithms-which are mostly performed by directly modifying the original AdaBoost can also be regarded as gradient descent for minimizing a unified objective function and their results also proved the effectiveness of the proposed algorithm.

GRADIENT BOOSTING

Gradient Boosting (GB) is a machine learning technique for regression/classification which produces more accurate prediction models in form of ensemble weak prediction models. It is an iterative algorithm that combines simple parameterized functions with weak performance (high prediction error) in order to produce a highly accurate prediction by minimizing the errors [1]. It builds models in a forward stage-wise procedure and generalizes these models by optimizing an arbitrary loss function. The method is highly robust and can be applied to classification or regression problems from different response distributions such as Poisson, Gaussian, Bernoulli, and Laplace.

The technique was perfectly applied in modeling insurance loss cost [1] to predict auto "at-fault" accident loss cost using data from a major Canadian insurer. The predictive accuracy of the model was compared against the conventional Generalized Linear Model (GLM) and proved to be more accurate. In addition, it had also been successfully applied to robust training of hidden Markov models for automatic speech recognition.

RESEARCH METHODOLOGY

The main idea of boosting is to provide an optimization technique which can manage and deal with a large number of covariates that are not informative and at the same time avoiding overfitting. However, Freidman showed that building additive models could be done in a step-wise manner such that at each stage a new weak learner is built by fitting gradients. The glmboost algorithm is generally used to fit linear regression via component-wise linear least squares models using L2 boosting. By default, glmboost fits a linear models with initial boosting iterations of 100 (mstop=100) and 10% shrinkage parameter (v=0.1). The function also uses Gaussian distribution by default to minimize squared error. On determining the optimum number of boosting iterations, it was observed that Cross-Validation (CV) which is one of the resampling techniques used in finding the optimum boosting iterations performs better than AIC based technique when fitting models using gamboost algorithm. Thus, the CV produced a lesser number of boosting iterations than the AIC based technique. Different distribution will be fitted the rent data where the distribution that best fits the data set will be selected via AIC technique. In addition, the data set will also be fitted using the classical GLM and GLMBOOST.

GLMBOOST FUNCTION (GRADIENT BOOSTING WITH LINEAR COMPONENTS)

Fitting generalized linear models using gradient boosting algorithm is termed glmboost. The function glmboost can be used to fit linear models via component-wise boosting and each column of the design matrix is fitted and selected separately using a simple linear mode. Thus, the algorithm is a gradient boosting for optimizing specific loss functions in which the component-wise linear models are used. He further explained that its main concept is to minimize empirical risk (e.g negative log likelihood) via stage-wise technique through Functional Gradient Descent (FGD). However, by default, glmboost function fits linear models with Gaussian family distribution by minimising square errors where the shrinkage parameter υ=0.1 and with initial mstop=100. When glmboost is used to fit any linear model, the results are more accurate and reliable than its corresponding glm function as we will soon verify.

FITTING THE RENT DATA USING GLMBOOST AND THE CLASSICAL GLM

Fitting linear regression models by means of gradient boosting with component-wise linear base learners is of interest of different researchers in Mathematics and Statistics today. To do this, we employ the glmboost function from glmboost package on R in order to obtain more accurate and efficient results. The model will then be compared with the corresponding glm function to prove the accuracy of the boosting algorithm. Therefore, fitting the glmboost models to the rent data set using the two main variables-the floor and the area of the rooms are fitted and the results are as follow

```
Generalized Linear Models Fitted via Gradient Boosting

Call:

glmboost.formula(formula=R ~ (Fl+A), data=rent, control=ctrl1)

Squared Error (Regression)

Loss function: (y - f)^2

Number of boosting iterations: mstop=500

Step size: 0.1

Offset: 811.8803

Coefficients:

(Intercept) Fl A

-6129.154236 9.214360 2.825308

attr(,"offset")

[1] 811.8803
```

By default, the function uses 100 number of boosting iterations. But without knowing the optimum number of boosting iterations, many covariates that might not be significant might also be included in the model. Therefore, it is highly important to determine how many boosting iterations are appropriate (optimum number) whereby only the covariates that are significant can be included in the model. To compute the optimum number of boosting iterations using AIC technique, the command below is employed.

```
mstop(aic<- AIC(model1))

[1] 131
```

Thus, the Akaike information criterion (AIC) was employed for determining the optimal number of boosting iterations and indicates that the optimum number of the boosting iterations is 131. Similarly, whatever the number of boosting iterations used in this model, the same number of optimal boosting iteration (131) is obtained which proved that 131 is the optimum boosting iterations as far as this model is a concern as shown in **Figure 1.**

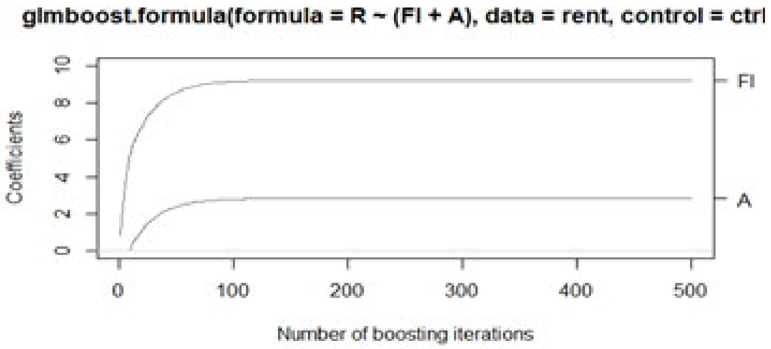

Figure 1. Graph of the coefficients of the fitted gamboost model.

Therefore the best model is the one with the optimum number of boosting iterations (131) which has the following coefficients. The plot of the optimum model with the corresponding optimum boosting iterations (131) using AIC based selection is shown in **Figure 2** below.

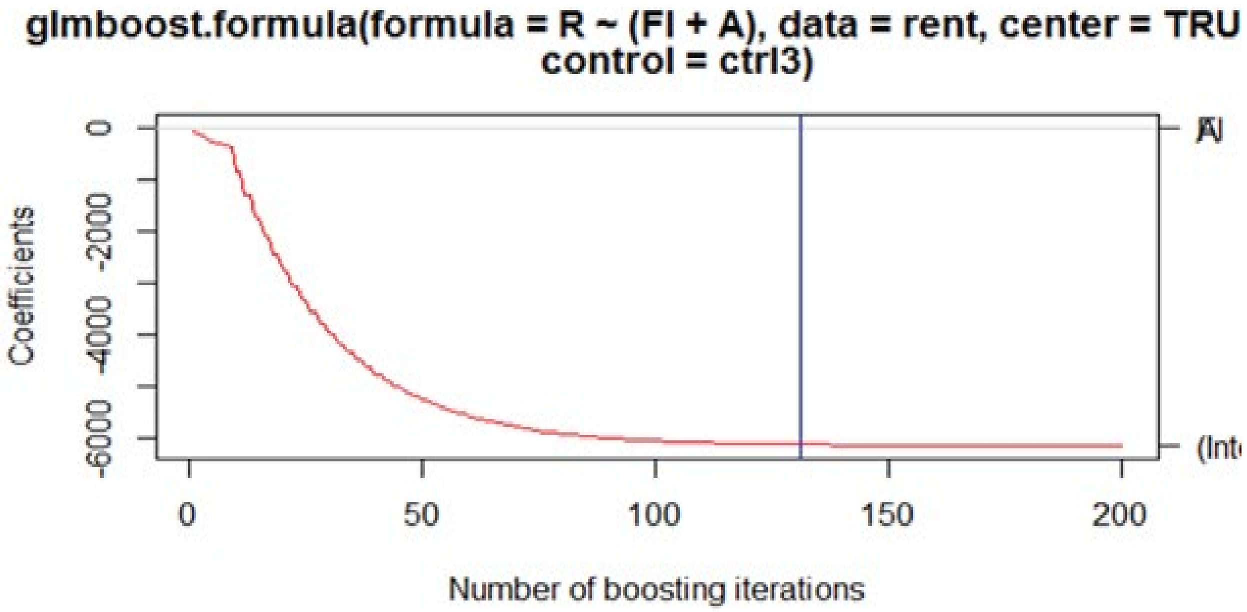

Figure 2. The plot of the model showing the optimum boosting iterations using AIC.

It is noticed that there is a slight decrease in the coefficients in the optimum model. The coefficients and the AIC was given by the optimum model are the best results as far as glmboost is a concern. More interestingly, it is highly recommended to fit the corresponding glm to the rent data and compare the individual AICs for both glm and glmboost. To fit the GLM to the data, the command glm is employed and the output is shown below:

```
Call: glm(formula=R ~ (FI+A), data=rent)

Coefficients:

(Intercept) FI A

 -5317.274 9.214 2.825

Degrees of Freedom: 1968 Total (i.e. Null); 1966 Residual

Null Deviance:      282700000

Residual Deviance: 205700000    AIC: 28350

>AIC(model2)

[1] 28351.14
```

It is clearly indicated that there is a massive increase in the AIC compared to the corresponding glmboost function. The optimum model in the glmboost [17] has a smaller AIC than the corresponding glm function. Therefore, the glmboost proved to be more accurate than the ordinary glm here.

CROSS-VALIDATION

The optimal stopping point is a very important parameter and needs to be considered. Thus when building models tuning is applied to prevent overfitting. Different techniques exist to accurately determine the optimal stopping iteration, but for the sake of accuracy, the cross-validation technique is employed in this project to estimate the empirical risk for choosing the appropriate number of boosting iterations. Cross-validation (CV) is one of the resampling techniques used in model selection, it is mostly used in selecting the number of boosting steps if the relationship between the covariates is linear and the covariates have a linear effect on each other. Other resampling techniques include bootstrap and AIC based selection as explained in the previous sections. However, CV is divided into three (3) types namely; random sub-sampling, K-fold cross-validation, and leave-one-out cross-validation. However, K-fold cross-validation will be employed here because of its ability to include all the test examples in the dataset for both training and testing [18].

We now fit a linear model to the rent data as above and employ the CV technique in selecting the optimum number of boosting iterations. Therefore, using the CV, we obtained 72 as the optimum number of iterations compared to 100 by default and 131 using AIC based-selection. The optimum model corresponding to the optimum number of the boosting iteration is shown below:

```
model1[mstop(cvm)]
Generalized Linear Models Fitted via Gradient Boosting
Call:
glmboost.formula(formula=R ~ (Fl+A), data=rent, center=TRUE, control=boost_control(trace=TRUE))
Squared Error (Regression)
Loss function: (y - f)^2
Number of boosting iterations: mstop=72
Step size: 0.1
Offset: 811.8803
Coefficients:
 (Intercept) Fl A
-5801.492376 8.977825 2.665367
attr(,"offset")
[1] 811.8803
```

STOPPING CRITERIA

The most important parameter of the algorithm is the number of boosting steps (M) and the penalty is supposed to be large enough otherwise the minimum AIC will occur at an early boosting step thereby yielding a non-optimal result. The outstanding advantage of the algorithm over the conventional methods for fitting generalized additive models such as GAM or GAMLSS will be enjoyed for a large number of covariates compared to the number of observations such as 10 covariates with only 100 observations [19].

GAMBOOST (GRADIENT BOOSTING WITH SMOOTH COMPONENTS)

Boosting Generalised Additive Models (gamboost) was introduced by Tutz and Binder [19] and is used for fitting generalized additive models using likelihood-based boosting. The main idea of gamboost is the Gradient Boosting (GB) for optimizing a certain loss function, where component-wise smoothing techniques are used as base-learners. The algorithm yields a predictor which is additive in the covariates. However, in gamboost, a large number of boosting steps are employed to obtain an additive model and in each step smoothers like B-splines are fitted to each covariate [20] and the response is the residuals from the previous step.

In order to update the covariate, some variable selection techniques such as deviance or boosting (by some particular model selection criterion like cross-validation) are used to select the covariate to be updated [21-25]. Therefore, for each step, the B-spline coefficient estimates are fitted and a smooth function estimate is obtained.

Gamboost employs the use of an approach that avoids fitting any uninformative covariates thereby fitting only the significant variables, therefore, an implicit technique of variable selection is silently carried out. **Table 1** below shows the summary of boosting iterations produced by the two (2) resampling techniques on both glmboost and gamboost [26,27].

S/N	Function	AIC based method	Cross-Validation
1	Glmboost()	131	72
2.	Gamboost()	111	71

Table 1. Comparison of the optimum number of iterations using AIC and CV.

CONCLUSION

This paper applies a gradient boosting algorithm to the GLM, and GAM to investigate the effect of boosting the packages. It has been noticed that boosting the packages yield better estimation than the classical ones. The algorithm is applied in order to minimize the errors which might be encountered during the prediction to provide a more accurate and reliable estimation. However, in boosting algorithm, optimum-boosting iterations are highly recommended to avoid over fitting [28]. It plays an important role when fitting model and hence the project employs the use of AIC based technique (in glmboost and gamboost) to determine the appropriate boosting iteration that gives the optimum prediction. We applied both the AIC and Cross-validation (CV) techniques to determine the optimum boosting iterations [29].

It is noticed that by default, the gamboost fits models using smooth base-learners (bbs). Thus, the same result is obtained if the model is fitted with or without bbs-which is a clear indication that the algorithm fits smoothers by default. Similarly, it also noted that the coefficients of the fitted model will be in matrix form if smooth base-learners are used while they are just linear if linear base-learners are used [30,31]. However, when the concept of the Cross-Validation is applied to the gamboost function, the optimum number of iteration reduced from 111 to 71 showing that CV produced a lesser number of boosting iterations than the AIC technique and therefore provides better results.

REFERENCES

1. Guelman L. Gradient boosting trees for auto insurance loss cost modeling and prediction. Expert Syst Appl. 2012;39:3659-3667.
2. Rigby RA, et al. A flexible regression approach using GAMLSS in R. London Metropolitan University, London. 2009.
3. Freund Y, et al. A short introduction to boosting. Journal-Japanese Society for Artificial Intelligence. 1999;14:1612.
4. Schapire RE. The strength of weak learnability. J Mach Learn Res. 1990;5:197-227.
5. Freud Y. Booting a weak learning algorithm by the majority. AT and T Laboratories-New Jessy. 1995.
6. Freund Y, et al. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 1997;55:119-139.
7. Hastie T Tibshirani, et al. The elements of statistical learning. Data Mining, Inference, and Prediction. 2009.
8. Elith J, et al. Boosted regression trees for ecological modeling. R documentation. Software. 2017.
9. Friedman J. et al. Discussion of boosting papers. Statistics Department-University of Stanford. 2003.
10. Ye J, et al. Stochastic gradient boosted distributed decision trees. In Proceedings of the 18th ACM conference on Information and knowledge management. ACM. 2009:2061-2064.
11. Culp M, et al. Ada: An r package for stochastic boosting. J Stat Softw. 2006;17:9.
12. Zhang B, et al. Application of specific loss function using gradient boosting. University of Washington, New York-USA. 2011.
13. Mayr A. et al. Generalised additive models for location scale and shape for high dimensional data-a flexible approach on boosting. J. Royal Stat. Soc. 61 Series. 2012;3:345-514.
14. Rigway G. The state of boosting, University of Washington-USA. 1999.
15. Schapire RE. The boosting approach to machine learning: An overview. In Nonlinear estimation and classification Springer, New York, NY. 2003:149-171.
16. Niu B, et al. Using Ada Boost for the prediction of subcellular location of prokaryotic and eukaryotic proteins. Mol. Divers. 2008;12:41.
17. Chu WS, et al. Identifying gender from unaligned facial images by set classification. In Pattern Recognition (ICPR), 2010 20th International Conference IEEE. 2010:2636.
18. Gutierrez-Osuna, R. Cross Validation, s.l. Wright State University. 2006.
19. Tutz G, et al. Generalized additive modeling with implicit variable selection by likelihood-based boosting. Biometrics. 2006;62:961-971.
20. Eilers PH, et al. Flexible smoothing with B-splines and penalties. Stat. Sci. 1996:89-102.
21. Voudouris V, et al. Modelling skewness and kurtosis with the BCPE density in GAMLSS. J. Appl. Stat. 2012;39:1279-1293.
22. Brown CD, et al. Body mass index and the prevalence of hypertension and dyslipidemia. Obes. Res. 2000;8:605-619.
23. R Forge G. R graphic manual: Families o9f gamlss models. In: R Statistical Software. 2012.
24. Rosset S, Zhu J. Piecewise linear regularized solution paths. Ann. Stat. 2007;35:1012-1030.
25. Stasinopoulos DM, et al. Modelling rental guide data using mean and dispersion additive models. J. Royal Stat. Soc. Series D (The Statistician). 2000;49:479-493.

26. Hofner B, et al. gamboostLSS: boosting methods for GAMLSS models. URL https://r-forge. r-project. org/projects/gamboostlss. R package version. 2012.

27. Huang H, et al. Faster gradient descent and the efficient recovery of images. Vietnam Journal of Mathematics. 2014;42:115-131.

28. Leathwick JR, et al. Variation in demersal fish species richness in the oceans surrounding New Zealand: an analysis using boosted regression trees. Mar Ecol Prog Ser. 2006;321:267-281.

29. Rigby B, et al. The distribution toolbox of GAMLSS. The GAMLSS Team. 2014.

30. Rigby RA, et al. Generalized additive models for location, scale and shape. J. Royal Stat. Soc. Series C (Applied Statistics). 2005;54:507-554.

31. Valiant LG. A theory of the learnable. Communications of the ACM. 1984;27:1134-1142.