Generalized linear tree: a flexible algorithm for predicting continuous variables

Alberto Rodrigues Ferreira [1] **Alex Akira Okuno** [1]

Abstract

Tree-based models are popular among regression methods to predict continuous variables. Also, Generalized Linear Models (GLMs) are pretty standard in many statistical applications and provide a generalization to many of the most commonly applied statistical procedures. However, in most regression tree methods, there is only one theoretical model associated for prediction in the final nodes, like multiple linear regression, logistic regressions, polynomial models, Poisson models, among others. We, therefore, propose a new tree method in which we estimate a GLM in each leaf node of the estimated tree including variable selection, new hyperparameters optimization, and tree pruning. Our method, called Generalized linear tree (GLT), has shown to be competitive compared to other well-known regression methods in real datasets, with the advantages and estimation flexibility provided by GLMs.

1. Introduction

A regression tree model is a nonparametric estimate of a regression function constructed by recursively partitioning a data set with the values of its predictor $\mathbf{X}$ variables (Loh & Zheng, 2012). This way, the tree algorithm partitions the explanatory variable domain into rectangles through a series of decision rules, in such a way to obtain sets that are similar between themselves. In a regression tree, it is common to use the mean value of the regions as a prediction for the instances that fall under this region.

Tree-based models have been extensively used in the statistical practice and have been object of great attention in the statistics literature partly due to its easy interpretability and good performance in classification and regression tasks. Despite these benefits, trees are not usually as accurate as other methods, such as Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs) (Elith et al., 2008). They have difficulty in modelling smooth functions, even ones as simple as a straight-line response at 45 degrees to two input axes.

In this paper, we propose a new methodology that incorporates the flexibility and predictive power of GLMs into regression tree models. The Generalized Linear Tree (GLT) is a supervised machine learning algorithm in which the main idea is to partition a dataset in order to minimize the variability of the partitions, measured by the standard deviation and adjust a more sophisticated and flexible parametric model (GLM) to each of these partitions. This way, if we have evidence to believe that a single model adjusted to the entire dataset is not adequate, we can generally improve the prediction of the response variable. We use the backward method to do the variable selection in the GLM models and we also do a hyperparameter optimization and tree pruning in order to avoid overfitting.

2. Related Work

Some of the most known decision/regression trees implementations are: CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993), where specifically in the regression case, the trees predict averages or medians, that is, for a given partition, the prediction is always the same, which is an undesirable property for some cases.

Another tree-based algorithm is the model tree, which is an algorithm that, for regression, combines a regression tree and a multiple linear regression (MLR), what might yield a more effective prediction, although there is only one theoretical model associated with this prediction (MLR). Some implementations of model trees are M5 (Quinlan et al., 1992) and M5' (Wang & Witten, 1996). Rtree model algorithms with another types of regression also exist, for example, there is a bayesian approach of GLM (Chipman et al., 2002), non-parametric models, Poisson model (Chaudhuri et al., 1995), multiple linear model (Alexander & Grimshaw, 1996) and polynomial models (Chaudhuri et al., 1994).

There are classification trees with a similar idea. One example of such algorithm is the Logistic Models Tree (Landwehr et al., 2005), which combines a decision tree and a logistic

[1]Department of Statistics, University of São Paulo, São Paulo, Brazil. Correspondence to: Alberto Rodrigues Ferreira <albertor@ime.usp.br>, Alex Akira Okuno <akira.okuno@ime.usp.br>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

regression to predict the probability of success of the classes involved in the problem. (Zeileis et al., 2008) also incorporates a similar idea, in the sense that it is possible to use any given parametric model in a recursive tree model. The parameters of this model are obtained via M-type estimators.

3. How the method works

The GLT goes through several steps during the training stage, where some of these steps are similar to existent tree-based models and some are novel to our methodology, which is where we propose to improve the predictive performance.

The main contribution that differentiates our paper from previous works is the greater number of models that can be estimated, while avoiding a prohibiting computational cost to estimate such models and doing a tree pruning procedure and optimizing new hyperparameters to avoid overfitting.

3.1. Dataset Partitioning

The first step in the GLT model is the partitioning of the explanatory variables space. We are going to denote this space by S_1 and the partitions by $\{S_2, \cdots, S_k\}$ (supposing $\lfloor k/2 \rfloor$ partitions), and the partition sets follow the rules below:

$S_i = S_{2i} \cup S_{2i+1}, \forall i = 1, \cdots, \lfloor k/2 \rfloor$
$S_{2i} \cap S_{2i+1} = \emptyset$

For each internal node of the tree, an exhaustive search is done in order to find the best partitions, i.e. the partitions that maximize the $V(S_i)$ metric:

$$V(S_i) = \sigma_i - \frac{|S_{i1}|}{|S_i|}\sigma_{i1} - \frac{|S_{i2}|}{|S_i|}\sigma_{i2},\ i = 1, \cdots, \lfloor k/2 \rfloor$$

where $|S_{i1}|$, $|S_{i2}|$ and $|S_i|$ are the number of observations that belong to the the subsets S_{i1}, S_{i2} e S_i, respectively, σ_i, σ_{i1} and σ_{i2} are the standard deviation of the response variable in the subsets S_i, S_{i1} and S_{i2} respectively.
The exact process to determine the subsets S_{i1} and S_{i2} given S_i depends on the type of explanatory variables, and is given as follows.

Continuous Variables

1. For each continuous explanatory variable x, we observe its unique values and sort them. We are going to denote by A_x the set of sorted unique values of the variable x.
2. For each $a_j \in A_x, j \in 1, \cdots, \#A_x - 1$, we calculate $R_j = \frac{a_j + a_{j+1}}{2}$
3. The sets S_{i1} and S_{i2} are constructed by finding the best R_j such that the $V(S_i)$ metric is optimized. Then:
   $S_{i1} = \{S_i : x \leq R_j\}$
   $S_{i2} = \{S_i : x > R_j\}$

Categorical Variables

1. For each categorical explanatory variable x, let A_x be the set of all of its factors observed.
2. For each $R_j \in A_x, j \in 1, \cdots, \#A_x$ the sets S_{i1} and S_{i2} are formed such that:
   $S_{i1} = \{S_i : x = R_j\}$
   $S_{i2} = \{S_i : x \neq R_j\}$

This process is summarized in the following pseudo-code:

Algorithm 1 Tree building

Input: training data X, y,number of subsets of the data k
for i=1 **to** $\lfloor k/2 \rfloor$ **do**
  Set best separator $V(S_i)$
  if $S_i = \emptyset$ **or** At least one hyperparameter condition (e.g. max. depth) is not satisfied **then**
    $S_{2i} = \emptyset$ and $S_{2i+1} = \emptyset$
  else
    $S_{2i} = S_{i1}$ and $S_{2i+1} = S_{i2}$
  end if
end for

3.2. Recursive model selection

Once the tree structure is build, GLMs are adjusted to all the leaf nodes. The distribution of the response variable $\mathbf{Y}$ to be adjusted has to belong to the exponential family:

$$f(y; \phi, \theta) = exp\{a(\phi)^{-1}[y\theta - b(\theta)] + c(y, \phi)\} \forall \phi, \theta \in \Theta$$

Then the expectation of $\mathbf{Y}$ is related to the linear predictor through $g(\mu) = x^T\beta$. This way, we can predict the response variable through the inverse function of the GLM link function: $\hat{y} = g^{-1}(x^T\hat{\beta})$.

We also added new hyperparameters that are not standard in the known tree algorithms, for example: (1) minimum percentage decrease in the response variable's standard deviation in each partition and (2) minimum quantity of models to be adjusted in each node. These hyperparameters aim to avoid overfitting.

Due to a high computational cost of adjusting several models in each node, we do a model selection procedure. This way, we choose the best predictive models using a hyperparameter $\alpha \in (0, 1)$, that defines the proportion of models that will be chosen in the next nodes so that only a subset of models that predict the response variable well enough are chosen.

Algorithm 2 illustrates the data partitioning process and the recursive model selection, considering the hyperparameters that are used as a stop criterion.

Algorithm 2 Recursive selection models

Input: Training data X, y, α: percentage of models to be chosen in the next node, k: number of subsets of the data, m: number of total MLGs, $\tau = \{1, \cdots, m\}$, MLG models used $\mathcal{M}$
for i=1 **to** k **do**
  if $S_i \neq \emptyset$ **then**
    initialize ϵ as an empty list of size m
    for j $\in \tau$ **do**
      Estimate $\mathcal{M}_j(X(S_i), y(S_i))$
      $\epsilon[j] = error(y, \hat{y} := \hat{\mathcal{M}}_j(X(S_i)))$
    end for
    $\tau = 100 * \alpha\%$ first indices of $sort(\epsilon)$
  end if
end for

Figure 1. Generalized linear models adjusted in each of the 3 subsets of a dataset

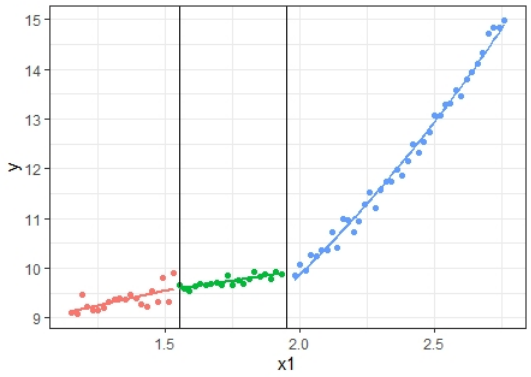
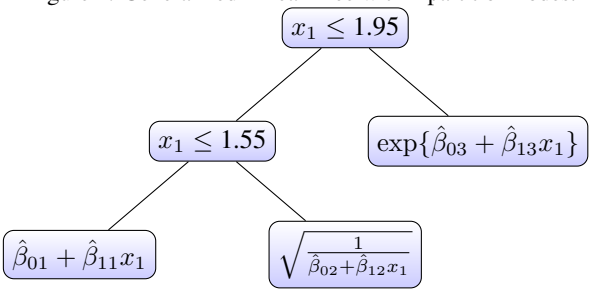


Figure 1 displays some simulated data and the prediction associated to it, where the estimated tree (shown in Figure 2) exemplifies the exact decision nodes and the GLMs adjusted to each of the 3 subsets rendered by the algorithm.

Figure 2. Generalized Linear Tree with 2 partition nodes.



Also, $\hat{\beta}_{\mathbf{0j}}$ and $\hat{\beta}_{\mathbf{1j}}$, $j = 1, 2, 3$ are the maximum-likelihood estimated parameters of the regression models in each subset.

3.3. Variable Selection

The variable selection in the GLMs is based off of a minimization of AIC criterion (Akaike, 1974), that is a negative log-likelihood based function that is penalized for the quantity of parameters in the parametric model. Obtaining the best set of explanatory variables via AIC is done, in practice, with a backwards stepwise regression in each subset.

This way, decision-makers can easily analyze the selected variables in the model and verify which path in tree led to the obtained prediction.

3.4. Tree pruning

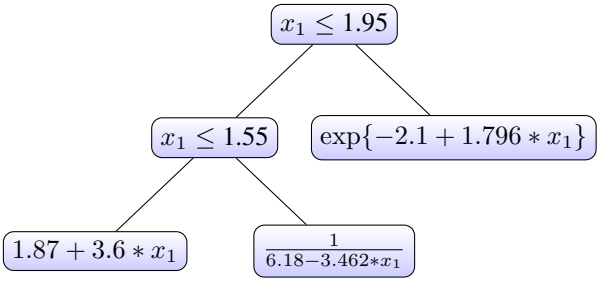
The process of tree pruning is the main method to avoid overfitting. This procedure happens after each subset has a final model associated with it. This simplifies the tree and make it generally more efficient for prediction and interpretability purposes.

The pruning is done from the bottom to the top, just as it is in a standard decision tree. There are 2 pruning types that can be used: (1) applies a correction in the training error(Quinlan et al., 1992) and (2) utilizes a pruning set to be obtained through a partition in the training set.

4. Predicting new observations

In order to do the prediction, we have to verify in which terminal node the specific observation to be predicted falls into. Based on the regression model in each terminal node, the new instances can be predicted.

Figure 3. Example of an Generalized Linear Tree with estimated coefficients



We show below examples of predictions for some values of x_1 according to the estimated tree shown in Figure 3.

x_1	Prediction
2.4	$\exp\{-2.1 + 1.796 * 2.4\} = 9.12$
1.75	$(6.18\text{-}3.462*1.75)^{-1} = 8.23$
3.31	$\exp\{-2.1 + 1.796 * 3.31\} = 46.75$
1.55	1.87+3.6*1.55=7.45

5. Experiments

The main goal of this paper is to evaluate the performance of GLT against other known algorithms in the literature. For this performance comparison, we gathered 7 regression datasets from `https://archive.ics.uci.edu/ml/datasets.php`.

We set our training set as 70% of the datasets and 30% for the test set. For hyperparameter optimization, we adopted a repeated cross-validation procedure consisting of 30 iterations of random search optimization, where each one has 5 cross validations with 3 folds for all tested algorithms. In all cases, we used the mean absolute error as the evaluation metric.

We tested the following algorithms: Generalized Linear Tree, Ridge Regression, Lasso Regression, Random Forest, Regression Tree, Gradient Tree Boosting, GLM with Normal distribution, GLM with Gamma Distribution and GLM with Inverse Gaussian Distribution.

The performance results of each algorithm is displayed in the table below [1], where the best performance is indicated in bold.

Table 1. Performance results in terms of absolute mean error

Algorithms	Iris	Auto	CST	RSV	Parkison	Qsar	Energy
GLT	**0.260**	**1.181**	10.133	6.753	5.560	0.248	7.260
GLM Normal	0.325	1.378	2.105	6.210	2.040	0.300	3.215
GLM Gamma	0.322	2.174	2.805	10.406	2.694	0.300	8.741
GLM Inverse Normal	0.460	2.174	5.753	10.406	2.686	0.300	8.741
Ridge	0.272	1.223	2.172	7.251	1.930	0.266	2.323
Lasso	0.394	1.223	2.164	7.322	1.932	0.300	2.500
Random Forest	0.300	1.263	2.698	5.943	1.351	0.242	1.377
Regression Tree	0.330	1.373	3.283	6.619	1.505	0.257	1.434
Gradient Tree	0.287	1.229	**2.041**	**5.913**	**0.312**	**0.227**	**1.014**

6. Conclusion

In this paper, we proposed a flexible alternative algorithm, the GLT, to solve continuous variable prediction problem. Our algorithm mixes the idea of regression trees with Generalized Linear Models. We observed that GLT outperformed every other algorithm tested in 2 out of 7 datasets. It is important to note that GLT outperformed the regularized regressions (LASSO and Ridge) and the plain GLMs in most datasets. Future research on this topic could include testing other families of models other than Generalized Linear Models, such as other types of parametric regression, non-linear methods and non-parametric methods.

[1] Some of the dataset names were abbreviated. Specifically: Auto for Auto MPG, CST for Concrete slump test, RSV for Real State Valuation, Qsar for QSAR fish toxicity and Energy for Energy efficiency.

References

Akaike, H. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

Alexander, W. P. and Grimshaw, S. D. Treed regression. *Journal of Computational and Graphical Statistics*, 5(2): 156–175, 1996.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. Classification and regression trees. 1984.

Chaudhuri, P., Huang, M.-C., Loh, W.-Y., and Yao, R. Piecewise-polynomial regression trees. *Statistica Sinica*, pp. 143–167, 1994.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y., and Yang, C.-C. Generalized regression trees. *Statistica Sinica*, pp. 641–666, 1995.

Chipman, H. A., George, E. I., and McCulloch, R. E. Bayesian treed models. *Machine Learning*, 48(1-3):299–320, 2002.

Elith, J., Leathwick, J. R., and Hastie, T. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008. doi: https://doi.org/10.1111/j.1365-2656.2008.01390.x.

Landwehr, N., Hall, M., and Frank, E. Logistic model trees. *Machine learning*, 59(1-2):161–205, 2005.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Loh, W.-Y. and Zheng, W. Regression trees for longitudinal and multiresponse data. 2012. doi: 10.1214/12-AOAS596.

Quinlan, J. R. C4. 5: programs for machine learning. 1993.

Quinlan, J. R. et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pp. 343–348. World Scientific, 1992.

Wang, Y. and Witten, I. H. Induction of model trees for predicting continuous classes. 1996.

Zeileis, A., Hothorn, T., and Hornik, K. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008.