Model-based Recursive Partitioning

Lisa McQuarrie

December 15, 2020

1 Introduction

The journal article "Model-based recursive partitioning" by Zeleis, et al. (2008) [6] details the theory and algorithm of model-based (MOB) recursive partitioning, or MOB trees. MOB trees combine a tree structure with a parametric model to create a prediction algorithm that has many of the benefits of classification and regression trees (CARTs), while reducing some of their disadvantages. In this report, MOB trees will be described and contrasted with the CART algorithm we learned in class, which was developed by Breiman et al. (1984) [1]. I chose to study model-based trees because the algorithm described by Zeleis, et al. forms a foundation from which methods to conduct prediction on many different data types have been built. In particular, I am interested in using prediction in a longitudinal data setting. Generalized linear mixed model (GLMM) trees have been applied to predict patient outcomes in the medical field, using longitudinal data [2], and the algorithm relies heavily on the work of Zeleis, et al. This report will not further consider GLMM trees, however, the usefulness of model-based trees as a building block to more advanced methods is emphasized.

1.1 Review of trees

Trees take a set of observations and recursively split the set into smaller subgroups (called nodes) that are internally similar with respect to the value of the response variable. In CARTs, the observations in the same node are modelled by the mean response in that node. The nodes are identified automatically and do not have to be pre-specified, which is a key advantage of trees over methods like linear regression, where the model formula must be specified a priori. Trees are well-suited to data that contain non-linear relationships between the response and the predictors or interactions between predictors. CARTs approximate the relationship between the response and predictors using a piece-wise constant function. The algorithm for estimating a CART is [4]:

1. For the observations in a node, search over all possible split points of all variables and choose the split point that partitions the range of a variable Z^* into intervals, R_1 and R_2, such that the objective function is minimized.
2. Split the node into two daughter nodes, where one daughter node contains observations with values of $Z^* \in R_1$ and the other contains observations with values of $Z^* \in R_2$.
3. Repeat Steps 1-2 for each node until a stopping criterion is met. Two common stopping criterion restrict the minimum number of observations in a node or the minimum improvement in the objective function. Predicted values for all observations in the same terminal node is the mean response in that node.

The CART algorithm locates the splitting variable and the splitting location in one step by finding the optimal split location over all variables. Variables with many levels have more possible split locations, so it is more likely that the split location will be at one of their levels, compared to variables with fewer levels, simply because there are more opportunities to chase the random errors in variables with many levels. Therefore, there is a bias towards splitting on variables with many levels.

In general, trees are easy to interpret and visualize, and are accessible for practitioners to use. For example, a doctor could intuitively use a tree to determine a diagnosis for a patient using a sequence of binary decisions. MOB trees will aim to maintain these attractive features of decision trees.

MOB trees contrast with CARTs in two key ways: 1) the predicted value in each terminal node is determined by a non-constant function and 2) the tree-fitting algorithm uses a different method of selecting a split location that involves a parametric test and is not biased towards variables with more levels. The remainder of the report will describe the method of MOB trees and provide an example that will be used to illustrate further details and contrast MOB trees with other methods, including CARTs.

1.2 Model-based trees

Model-based trees fit a parametric model in each node of a tree. The motivation behind MOB trees is that the parametric model fit to all observations may not describe the data as well as several local models fit to subgroups of observations, if there are underlying differences between subgroups of observations. The advantage of MOB trees over a global parametric model is that MOB trees can automatically detect subgroups or interactions between the predictors in the model and splitting variables. An equivalent global model would require determining or hypothesizing which subgroups exist before fitting the model and including the subgroup interactions in the model formula. MOB trees can conduct regression or classification by using the appropriate parametric model.

There are two types of predictor variables in MOB trees. A regressor is a predictor included in the parametric model within each node and a partitioning variable is a variable that may be chosen for splitting a node. Unlike with CARTs, the analyst, perhaps with the help of a subject-matter expert, must decide whether a variable should be a regressor or a partitioning variable. For categorical variables, including the variable as a regressor is assuming that the levels of the variable characterise subgroups and including the variable as a partitioning variable allows the tree to adaptively identify subgroups characterised by the variable. For continuous variables, including the variable as a regressor assumes that the relationship between the response and the regressor dictated by the model is correct, at least within subgroups of the sample; if included as a partitioning variable, the tree structure will approximate its relationship with the response.

With the current trend towards personalized medicine, one important use for MOB trees is in the medical field because of their ability to automatically detect interactions [3]. Subgroups of patients that respond better or worse to a treatment can be identified by fitting a MOB tree with the treatment as the regressor and other observed variables as partitioning variables.

2 Algorithm

A parametric model, denoted $M(\mathbf{Y}, \mathbf{X}, \theta)$, is fit in each node, where $\mathbf{Y} = (Y_1, \ldots, Y_n)$ represents the response observations, $\mathbf{X}$ is an $n \times m$ matrix with the rows representing the n observations and columns representing the m model regressors (including a 1 for the intercept), and $\theta = (\theta_1, \ldots, \theta_k)$ is a vector of model parameters. A vector of p partitioning variables, $Z = (Z_1, \ldots, Z_p)$, is also observed. $M(\mathbf{Y}, \mathbf{X}, \theta)$ can be any model that can be estimated by minimizing some objective function, $\Psi(\mathbf{Y}, \mathbf{X}, \theta)$. In a regression problem, a linear model could be used, and in a classification problem, a logistic or multinomial regression could be used. MOB trees aim to split the sample into groups of observations until each group is internally similar enough to be explained by a model with the same estimated parameters, and different enough from other groups that the estimated parameters differ between groups.

The algorithm used to estimate a MOB tree is

1. Let $\psi(Y_i, X_i, \theta) = \frac{\partial \Psi(Y_i, X_i, \theta)}{\partial \theta}$. Fit the model to all observations in a node by finding $\hat{\theta}$ such that $\sum_{i=1}^{n} \psi(Y_i, X_i, \hat{\theta}) = 0$.
2. Assess whether parameter estimates are *stable* for all partitioning variables. If not stable, select the variable with the most instability to split on.
3. Search over all possible split locations within the range of the chosen variable to locate the split that would minimize $\Psi(\mathbf{Y}, \mathbf{X}, \theta)$.
4. Repeat Steps 1-3 until the parameter estimates in all nodes are stable.

Sections 2.1 and 2.2 describe further details regarding Steps 2 and 3, respectively.

2.1 Stability of parameter estimates

To determine whether splitting a node would improve the fit of the tree to the data, MOB trees assess whether the node is *stable*. In a stable node, splitting the node at any point in the range of any partitioning variable would generate two daughter nodes that, when the parametric model is fit in both daughter nodes, would have similar estimates of the model parameters. Stability is tested using independent hypothesis tests for each partitioning variable.

For a particular partitioning variable, Z_j, if the parameter estimates are stable then $\hat{\psi}_i = \psi(Y_i, X_i, \hat{\theta})$ will randomly fluctuate around its mean of 0 over the range of Z_j. If there is systematic deviation from 0 in $\hat{\psi}_i$ over the range of Z_j, this indicates that the range of Z_j should be split into segments. Consequently, the hypothesis test for parameter stability is formulated to test for systematic deviations from 0 in $\hat{\psi}_i$.

The test statistic is based on the magnitude of the score function, scaled by an estimate of the variance of the score function.

Define $W_j(t)$ as [6]

$$W_j(t) = \hat{J}^{-0.5} n^{-0.5} \sum_{i=1}^{t} \psi(Y_{\sigma(Z_{ij})}, \hat{\theta}) \quad t \in 1, \ldots, n, \tag{1}$$

where $\hat{J}$ is an estimate of $\mathrm{Cov}(\psi(Y, X, \hat{\theta}))$, for example $\hat{J} = n^{-1} \sum_{i=1}^{n} \psi(Y_i, X_i, \hat{\theta}) \psi(Y_i, X_i, \hat{\theta})^T$ [6]. Z_{ij} is observation i of variable j and $\sigma(Z_{ij})$ is a permutation that returns the ranking of Z_{ij} when $Z_{1j}, \ldots, Z_{nj}$ are ordered from largest to smallest.

$W_j(t)$ is a partial sum process that sums the score function over a subset of the observations and scales it based on its variance. Specifically, the observations $Y_1, \ldots, Y_n$ are ordered from largest to smallest value of Z_j, then $\hat{\psi}$ is summed over nested subsets of increasing size. t controls the size of the subset. The test statistic is a function of $W_j(t)$ that aggregates $W_j(t)$ over t to capture all the evidence against the null hypothesis of stability. Further discussion of the mathematical formulation of the test statistic is deferred to Appendix A. The test statistic is purposefully formulated so that the model only has to be estimated once in the node and not in the daughter nodes generated by each possible split point. The test statistic has a known asymptotic distribution, so an asymptotic p-value can be calculated. A small p-value indicates instability.

The hypothesis test of stability is repeated, independently, for each of $Z_1, \ldots, Z_p$. Consequently, a multiple testing adjustment should be implemented to prevent inflation of the type I error rate. The Bonferroni adjustment is most common; the significance level used in each hypothesis test is $\frac{\alpha}{p}$ where α is the desired type I error rate for the node. When this adjustment is used, the probability of rejecting the null hypothesis of parameter stability and making a erroneous split at a node is at most α. The authors suggest $\alpha = 0.05$ is often appropriate, however, α can be treated as a tuning parameter.

If one or more of the hypothesis tests returns a p-value less than the significance level, then the variable that produced the test with the smallest p-value is chosen for splitting the node. If none of the p-values are less than the significance level, the node is not split.

The hypothesis test does not provide evidence regarding the location of the split within the range of the chosen variable. The split location is identified in the next step.

2.2 Split location

If, in Step 2, a variable, Z^*, was chosen to split on, then the next step is to find the partition, R_1, R_2, of Z^* such that the objective function is minimized. This is similar to how the split location is found in the CART algorithm, however, the CART algorithm searches all possible splits over all variables for the split location that minimizes the objective function, whereas in this step only the range of Z^* is searched. Zeleis et al. develop the theory for the general case where nodes can be split into more than 2 daughter nodes, but only the binary case will be discussed here to align with the content taught in class.

The split location is the $\text{argmin}_{R_1,R_2} \sum_{i \in R_1} (Y_i - \bar{Y}_1)^2 + \sum_{i \in R_2} (Y_i - \bar{Y}_2)^2$. The observations are grouped into daughter nodes based on their values of Z^* relative to the split location, then the process is repeated in the next node until the null hypothesis is not rejected in all of the p hypothesis tests in each of the nodes.

Generally, pruning is not necessary for MOB trees, as α controls the tree size and erroneous splits are made at a rate $\leq \alpha$.

In the CART algorithm, the splitting variable and split location are identified in one step. In the MOB tree algorithm, the splitting variable is chosen without considering a specific split location within the range of the variable. Consequently, the choice of the splitting variable is not influenced by the number of levels of the variable. Unlike the CART algorithm, selection of the splitting variable in the MOB tree algorithm is unbiased.

3 Example: Air quality data

The air quality data set will be used to exhibit the method of MOB trees and to contrast MOB trees with regression trees and linear regression. The air quality data set contains daily readings from an airport in New York that describe aspects of air quality for the summer of 1973. Variables contained in the data set are

- Ozone: average ozone level, continuous
- Solar.R: solar radiation, continuous
- Wind: average wind speed, continuous
- Temp: maximum daily temperature, continuous

- Month: month of reading, categorical
- Day: day of reading, categorical

The purpose of this example is to conduct regression to predict the variable Ozone. To be consistent with the analysis completed in class, the variables Solar.R and Day will not be included, however, the variable Month will be included as a categorical variable because, based on the marginal plots (Figure 4 in Appendix B), Month appears to have some effect on Ozone.

Figure 4 in Appendix B contains plots of Ozone against each of the predictors. Based on the plot of Ozone and Wind, it may be reasonable to assume that Ozone is a linear function of Wind. However, the plot of Ozone and Temp shows curvature and an assumption of linearity is likely unreasonable. Month is categorical, therefore, including it as a regressor in the in-node linear model would specify a different intercept for each level in Month. Specifying Month as a partitioning variable would mean that the relationship between Month and Ozone would be approximated by the tree structure and interaction effects between Month and the other variables could be detected. The plot of Ozone and Month shows that some levels of Month may have a different distribution of Ozone than others, however, it is difficult to determine what specific relationship should be assumed. Therefore, Month will be specified as a partitioning variable so that the MOB tree can adaptively detect subgroups in the Month variable. To summarise, a MOB tree with a simple linear model is used to predict Ozone. Wind is the regressor in the simple linear model and Temp and Month are partitioning variables.

The MOB tree is compared to a multiple linear regression (fit on the entire sample), where Ozone is predicted using Month, Temp, and Wind, and a regression tree which uses Month, Temp, and Wind as partitioning variables. Regression trees pruned with both the minimum complexity parameter (CP) rule and the 1 standard error CP rule were initially tested, however, pruning with the minimum CP rule produced lower errors so results will only be shown for the regression tree pruned using the minimum CP rule.

The package `partykit` in `R` is used to fit the MOB tree and to provide visualizations of the MOB and regression trees. The function `glmtree` with the default parameter settings of $\alpha = 0.05$ and Bonferroni multiple testing adjustment is used to fit the MOB tree. The `glmtree` function internally uses `glm.fit`, so it can fit an MOB tree with any model that can be fit by `glm.fit`. The `rpart` function in the package `Rpart` is used to fit the regression tree with the argument `cp`=0 so that a large tree is grown and then pruned using the minimum CP rule. The `lm` function in the `stats` package is used to fit the linear regression. Figure 1 shows the fitted MOB tree, Figure 2 shows the fitted regression tree, and Equation (4) (in Appendix B)

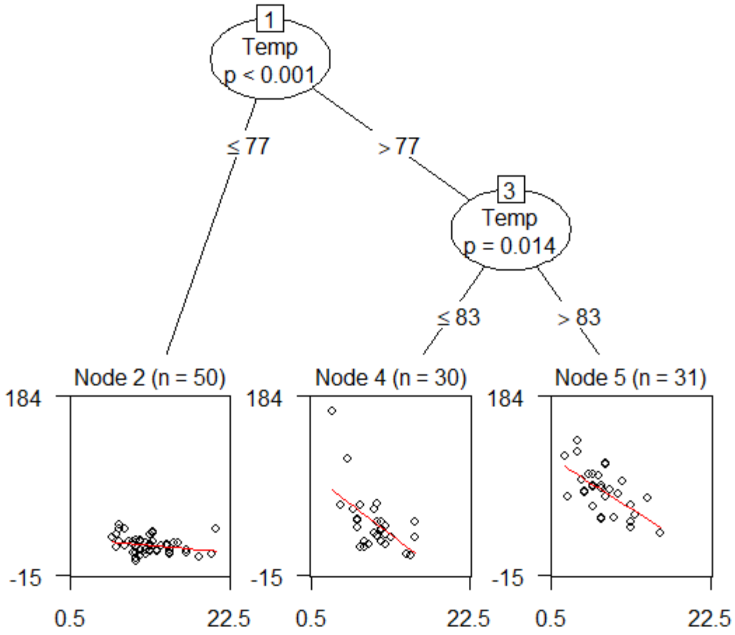

Figure 1: Plot of the fitted MOB tree on the air quality data.

gives the fitted equation of the multiple linear regression.

3.1 Comparison of methods

Figure 1 shows the MOB tree estimated on the air quality data set. The MOB tree splits only twice and is much smaller than the regression tree shown in Figure 2. In general, MOB trees will tend to make fewer splits than regression trees. The observations in the terminal nodes of a MOB tree are not all assigned the same prediction, therefore, the terminal nodes do not have to be as homogeneous as in regression trees.

The MOB tree uses the following piece-wise linear function to predict Ozone.

$$\widehat{\text{Ozone}} = \begin{cases} 26.41 - 0.669\text{Wind} & \text{if Temp} \leq 77 \\ 101.26 - 6.09\text{Wind} & \text{if } 77 < \text{Temp} \leq 83 \\ 117.82 - 5.08\text{Wind} & \text{if Temp} > 83 \end{cases}$$

The MOB tree detected three subgroups, characterised by the value of Temp, between which the estimated simple linear models differ.

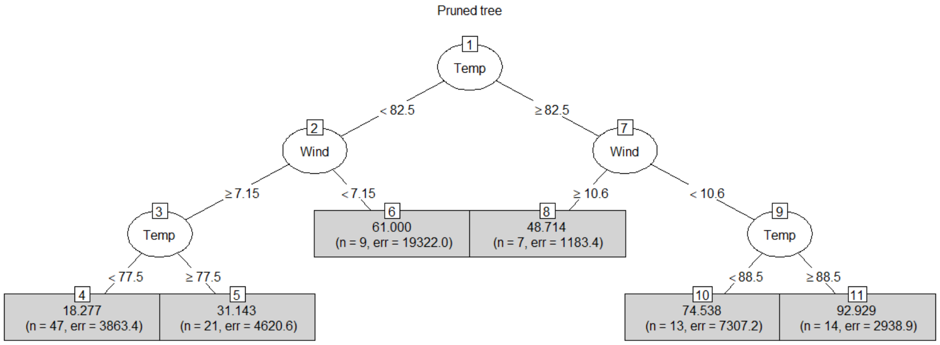

Figure 2: Regression tree fit with `rpart` and pruned with the minimum CP rule.

The fitted MOB tree is small and easy to interpret. A practitioner using this model to guide their decision making would likely be comfortable with common regression models like linear regression and, therefore, would find the piece-wise linear prediction function easy to use. Trees, in general, are easy to interpret, but when large can be cumbersome. Therefore, the tendency of MOB trees to be smaller than regression trees can, in many cases, improve interpretability.

One of the disadvantages of regression trees is that their prediction functions are highly discrete. The prediction function of a regression tree is a step function that can have very large jumps; two observations with very similar values of Temp and Wind can have very different predicted values if they fall on different sides of a split. Because the MOB tree has a piece-wise linear prediction function, this problem is reduced. For example, consider two points that only differ in the value of Temp by 2°: (Wind = 7.5, Temp = 82, Month = 5) and (Wind = 7.5, Temp = 84, Month = 5). These points fall into different terminal nodes in the regression tree and, consequently, the predicted values at these points are very different: 31.14 and 74.53, respectively. In the MOB tree, these observations again fall into two different nodes, therefore, their predictions come from two simple linear models with differing slopes and intercepts. However, their predicted values are more similar than in the regression tree: 55.58 and 79.72, respectively. Although the prediction function of a MOB tree is still discrete, it may have smaller jumps between the function segments than that of a regression tree. Additionally, since MOB trees tend to have fewer splits than regression trees, two points with similar values of the partitioning variables are less likely to fall into different terminal nodes.

The three fitted models exhibit the differences between the MOB tree, regression tree, and global multiple linear regression. The MOB tree fits a simple linear model in 3 groups, where the groups are determined by the value of Temp, creating an interaction between Wind and Temp. The regression tree fits a constant function in 6 groups, where the groups are determined by the values of Temp and Wind. Because the regression tree splits on both Temp and Wind, these variables interact. The interactions in the MOB and regression trees were not specified a priori. The global multiple linear model assumes a linear relationship between the two continuous predictors and Ozone, and allows observations in different months to have a different intercept (since Month was specified as categorical). Any interactions would need to be specified in the equation of the linear model.

Neither the MOB tree (Figure 1), nor the regression tree (Figure 2), used any information from the Month variable. In contrast, for the multiple linear regression, a likelihood ratio test found that Month has a significant effect on Ozone after adjusting for the effects of Wind and Temp. Because a tree structure is a coarse approximation of the relationship between the partitioning variables and the response, both MOB trees and regression trees can often ignore partitioning variables of moderate importance if there are other variables that are more important. This is a disadvantage of regression trees that MOB trees share.

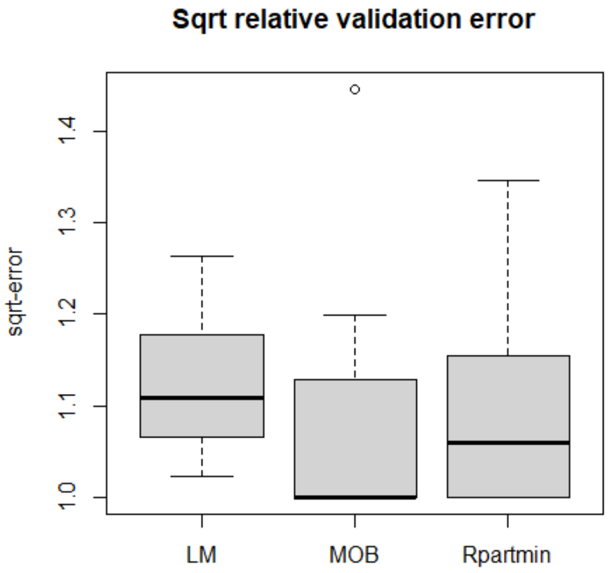

Figure 3: Boxplot of the square-root of relative validation error from 10 bootstrap iterations. Rpartmin is a regression tree fit using `rpart` and pruned with the minimum CP rule. LM is a multiple linear regression and MOB is a model-based tree.

To compare the performance of the three methods, bootstrap resampling is used to estimate the test error. 10 iterations of bootstrap resampling are used, and the three methods are fit on each resample, then are used to predict the observations not included in the bootstrap resample. The distributions of the square-root of relative test error for each method are shown in Figure 3. The relative test error was calculated by dividing the test error from a resample by the minimum test error from the three methods on that resample.

Figure 3 shows that the MOB tree has the smallest median and third quartile relative test error, although it performs poorly on one of the 10 runs. On this data set, in general, the MOB tree produces better predictions than the regression tree and the linear model. The multiple linear regression has a larger median relative test error than both the MOB tree and the regression tree, however, it is more consistent than the other two methods. The variance of the relative bootstrap errors from the multiple linear regression is smaller and it has a lower maximum bootstrap error than the other two methods. The multiple linear regression estimates the fewest parameters of the three methods, and therefore has the lowest variance, so it is unsurprising that the mean squared errors of the linear model are less variable. The regression tree tends to overfit to the bootstrap sample (recall that the regression tree fit to all the available data was much larger than the MOB tree) and does not predict well on observations outside the bootstrap sample. The air quality data set contained both linear and non-linear relationships; the MOB tree is well-suited to such data sets since it can both assume a linear relationship and approximate a non-linear relationship. In this example, a piece-wise linear prediction function performs better than a piece-wise constant function or a global linear function.

4 Conclusion

The MOB tree is a viable predictive method that performes well when applied to the air quality data set compared to the regression tree and multiple linear regression. The MOB tree improves upon some of the disadvantages of the CART while maintaining the benefit of interpretability. Disadvantages of the CART include a biased splitting variable selection method, a piece-wise constant prediction function, and a decrease in interpretability when the tree size increases. In contrast, the MOB tree algorithm separates the selection of a splitting variable and the split location into two steps, thereby achieving unbiased selection of the splitting variable. Additionally, the MOB tree uses a non-constant prediction function in the nodes and, as a consequence, will tend to grow smaller, more interpretable trees.

The MOB tree algorithm also removes the step of pruning that is needed in the CART algorithm to reduce the tree size. The pre-determined significance level provides an internal stopping criterion and unnecessary splits will only occur with at most rate α. If desired, α can be tuned to find the optimal tree size.

Of course, like CARTs, MOB trees are unlikely to be the best predictors when compared with ensemble methods or artificial neural networks, particularly on larger data sets. Despite this, MOB trees have been shown to have acceptable predictive abilities on small to moderate sized data sets and are useful for data exploration and detection of interactions.

Zeleis et al. establish a general algorithm that can be applied to fit MOB trees with a wide range of parametric models and, consequently, can be used to make predictions for many different data types. The MOB tree algorithm can be directly applied with simple or multiple linear regression, generalized linear models, and survival models. It can conduct either regression or classification without requiring any modifications to the algorithm by selecting the appropriate in-node model (for example, a linear model for regression or a logistic regression for classification). The algorithm has also been extended to the context of longitudinal data and time series data. Additionally, MOB trees have been used as weak learners in ensemble methods, such as random forests and boosting. The paper by Zeleis et al. outlines a general and flexible foundation from which many other methods have been developed. MOB trees are an interesting and important topic due to their acceptable predictive abilities, capabilities in data exploration, and capacity to be modified and applied to a wide range of problems.

References

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees.* CRC Press, January 2017.

[2] M. Fokkema, J. Edbrooke-Childs, and M. Wolpert. Generalized linear mixed-model (glmm) trees: A flexible decision-tree method for multilevel and longitudinal data. *Psychotherapy Research*, pages 1–13, June 2020.

[3] M. Fokkema, N. Smits, A. Zeileis, T. Hothorn, and H. Kelderman. Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods*, 50(5):2016–2034, October 2018.

[4] T. Loughin. Stat 852 lecture 8: Regression trees, September 2020.

[5] A. Zeileis and K. Hornik. Generalized M-fluctuation tests for parameter instability. *Statistica Neerlandica*, 61(4):488–508, November 2007.

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

5 Appendix A: Algorithm details

5.1 Identifying split variable: test statistic

Continuous variables

For continuous variables, the test statistic for the hypothesis test of stability is [6, 5]

$$\lambda(W_j) = \max_{i=1,\ldots,n}(\frac{i}{n} \times \frac{n-i}{n})^{-1} \|W_j(i)\|_2^2 \tag{2}$$

This is the maximum of the squared L_2 norm of $W_j(t)$, scaled by its variance function [6]. The limiting distribution of this test statistic is related to a Bessel process [6], therefore, an asymptotic p-value can be calculated.

Categorical variables

The test statistic for categorical variables must account for the possibility of observations with the same value, unlike in the continuous variable case. Let C be the number of categories, n_c be the number of observations in category c, and I_c be the index of category c in $1, \ldots, C$. The test statistic is [6]

$$\lambda(W_j) = \sum_{c=1}^{C} \frac{I_c^{-1}}{n} \| \sum_{i=1}^{n_c} W_j(i)\|_2^2 \tag{3}$$

The test statistic is the squared L_2 norm of the sum of the scores in category c, weighted and summed over all categories. It has an asymptotic χ^2 distribution with $k(C-1)$ degrees of freedom [6], where k is the number of parameters in θ, so an asymptotic p-value can be calculated.

6 Appendix B: Figures

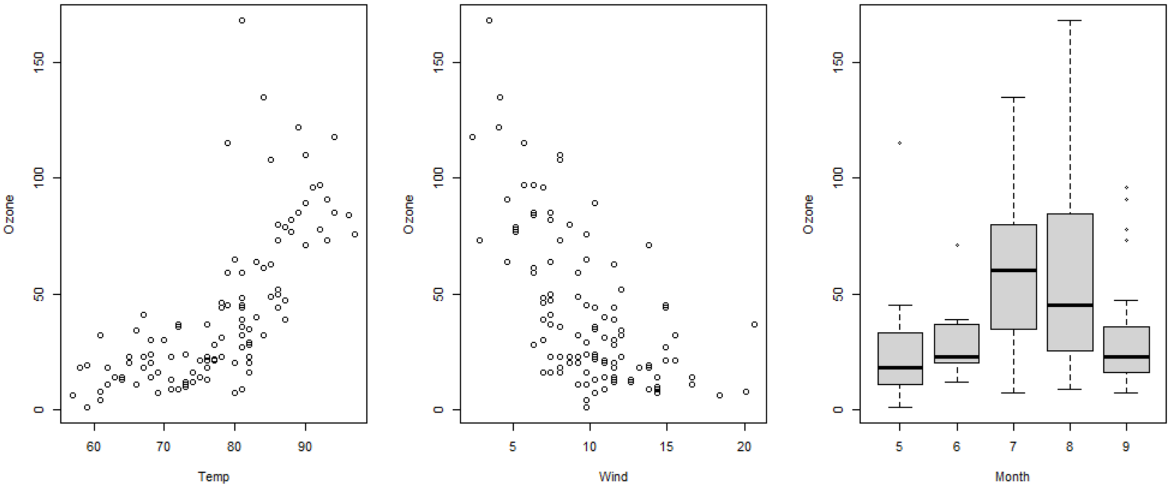

Figure 4: Relationships between the predictors and response in the air quality data.

$$
\begin{aligned}
\widehat{\text{Ozone}} = & -82.55 - 3.03 \times \text{Wind} + 2.13 \times \text{Temp} - 17.70 \times \text{I}(\text{Month} = 6) - 11.18 \times \text{I}(\text{Month} = 7) \\
& - 8.87 \times \text{I}(\text{Month} = 8) - 19.25 \times \text{I}(\text{Month} = 9) \qquad (4)
\end{aligned}
$$

Equation for the multiple linear regression fit to the air quality data. I(Month$= i$) $= 1$ if the observation is in Month i and 0 otherwise.