

# A toolkit for stability assessment of tree-based learners

Michel Philipp, *University of Zurich*, [Michel.Philipp@psychologie.uzh.ch](mailto:Michel.Philipp@psychologie.uzh.ch)  
Achim Zeileis, *Universität Innsbruck*, [Achim.Zeileis@R-project.org](mailto:Achim.Zeileis@R-project.org)  
Carolin Strobl, *University of Zurich*, [Carolin.Strobl@psychologie.uzh.ch](mailto:Carolin.Strobl@psychologie.uzh.ch)

**Abstract.** Recursive partitioning techniques are established and frequently applied for exploring unknown structures in complex and possibly high-dimensional data sets. The methods can be used to detect interactions and nonlinear structures in a data-driven way by recursively splitting the predictor space to form homogeneous groups of observations. However, while the resulting trees are easy to interpret, they are also known to be potentially unstable. Altering the data slightly can change either the variables and/or the cutpoints selected for splitting. Moreover, the methods do not provide measures of confidence for the selected splits and therefore users cannot assess the uncertainty of a given fitted tree. We present a toolkit of descriptive measures and graphical illustrations based on resampling, that can be used to assess the stability of the variable and cutpoint selection in recursive partitioning. The summary measures and graphics available in the toolkit are illustrated using a real world data set and implemented in the R package **stablelearner**.

**Keywords.** Stability, Recursive partitioning, Variable selection, Cutpoint selection, Decision trees

## 1 Introduction

Recursive partitioning approaches, such as classification and regression trees (CART, [2]), conditional inference trees [5] or model-based recursive partitioning [7], are widely used for modelling complex and possibly high-dimensional data sets [11]. The methods are able to detect high-degree interactions and nonlinear structures in a data-driven way. Therefore, these methods have been frequently applied in many scientific disciplines, as well as in many industries for predictive modeling purposes [8].

Nowadays, more complex and more flexible methods exist for predictive learning, that often achieve a better prediction accuracy (e.g., random forests, boosting, support vector machines, neural networks). Recursive partitioning, however, is still a popular method in situations where the aim is to infer and interpret the structure of the underlying process that has generated the data. For this purpose, recursive partitioning is often favoured over other methods, since the results can be illustrated in the form of decision trees, which are relatively easy to interpret. Therefore tree-based methods are widely used as exploratory modeling techniques in many fields, such as social and behavioral sciences (see e.g., [7]).

Recursive partitioning algorithms recursively split the predictor space  $\mathcal{X} \in \mathbb{R}_p$  to form homogenous groups of observations. The various algorithms, that have been proposed in the literature, mainly differ with respect to the criteria for selecting the split variable, choosing the cutpoint and stopping the recursion

(see [5]). CART, for example, selects the variable and the cutpoint that best unmixes the classes in case of a classification problem, or that most reduces the squared error loss in case of a regression problem. Conditional inference trees, on the other hand, perform splitting and stopping based on a statistical inference procedure.

Despite their popularity, a major drawback of recursive partitioning methods is their instability. By studying the predictive loss of different regularization techniques, Breiman [3] identified recursive partitioning (among others) as unstable. It is well known that small changes in the training data can affect the selection of the split variable and the choice of the cutpoint at any stage in the recursive procedure, such that the resulting tree can take a very different form [8, 11, 12]. Moreover, recursive partitioning methods do not provide measures of confidence for the results. Therefore, users cannot assess the degree of certainty for selected variables and cutpoints. Hence, the question remains to what extent one can rely on the splits in a single tree to draw conclusions.

Previous research has already focussed on assessing the stability of trees from different perspectives and with different goals (see e.g., [1, 4, 9]). Their methods are commonly based on a measure that is used to compare the distance (or similarity) between pairs of trees. In [4], for example, a measure of similarity between trees is proposed to stabilize the selection of the splits in a specific tree algorithm. And more recently, measures and theory were proposed to detect observations that influence the prediction or the pruning in CART [1]. While in these approaches the prediction, partitioning and the structure of the trees are considered separately, they may also be combined in one measure [9]. Thus, while previous research has focussed on reducing instability, measuring the influence of individual observations or assessing the distance between trees, we focus on assessing and visualizing two important aspects that reveal the stability of a tree resulting for a given data set: the variable and the cutpoint selection.

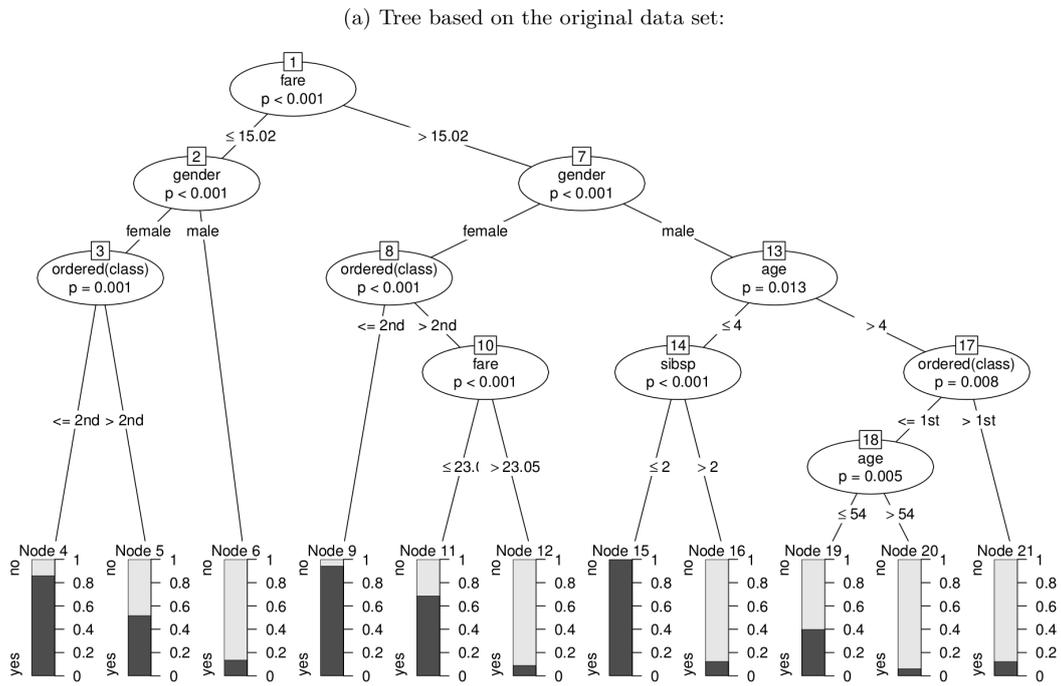
In this paper, we first discuss instability of results from recursive partitioning using a practical example. In the second part, we present a computational procedure and a number of graphical tools that support users for assessing the stability of the variable and the cutpoint selection. The proposed methods are implemented in the software package **stablelearner** (currently available from <https://R-Forge.R-project.org/projects/stablelearner/>) for the free R system for statistical computing [10]. By using a real world data set, the package will be used throughout the article for illustrating the proposed methods.

## 2 Instability of trees

To illustrate the instability of trees we have used recursive partitioning to predict the survival of the passengers during the sinking of the RMS Titanic in 1912 by several passenger characteristics. A complete passenger list is available online on <http://www.encyclopedia-titanica.org/> (accessed on 2016-04-05). According to the list, 1317 passengers (excluding crew members) were aboard from which 500 survived the sinking. The passenger information, that was used for training the tree, was gender, age, fare, class (1st, 2nd or 3rd), place of embarkment (B = Belfast, C = Cherbourg, Q = Queenstown, S = Southampton), number of siblings/spouses aboard (abbreviated as sibsp) and number of parents/children aboard (abbreviated as parch). The last two features were obtained from an overlapping data set available on

<http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets>. The tree was generated using the function `ctree` from the **partykit** package [6] that performs recursive partitioning in a conditional inference framework in R and is illustrated in the form of a tree in the upper panel of Figure 1. In the following, we will refer to this result as the original tree, since the partitioning was performed on the original passenger data (as opposed to random samples drawn from the original data set employed subsequently).

Based on a bootstrap sample taken from the original passenger data, we generated a second tree, which is illustrated in the lower panel of Figure 1. The structures of the trees look quite different at first sight, which suggests a large instability of the tree. However, when looking closer one can identify variables that were selected in both trees and split at the same or a similar cutpoint. For example, the numerical variable `age` was split at 4 and 54 in the original tree and at the values 4 and 36 in the bootstrap



(b) Tree based on a bootstrap sample drawn from the original data set:

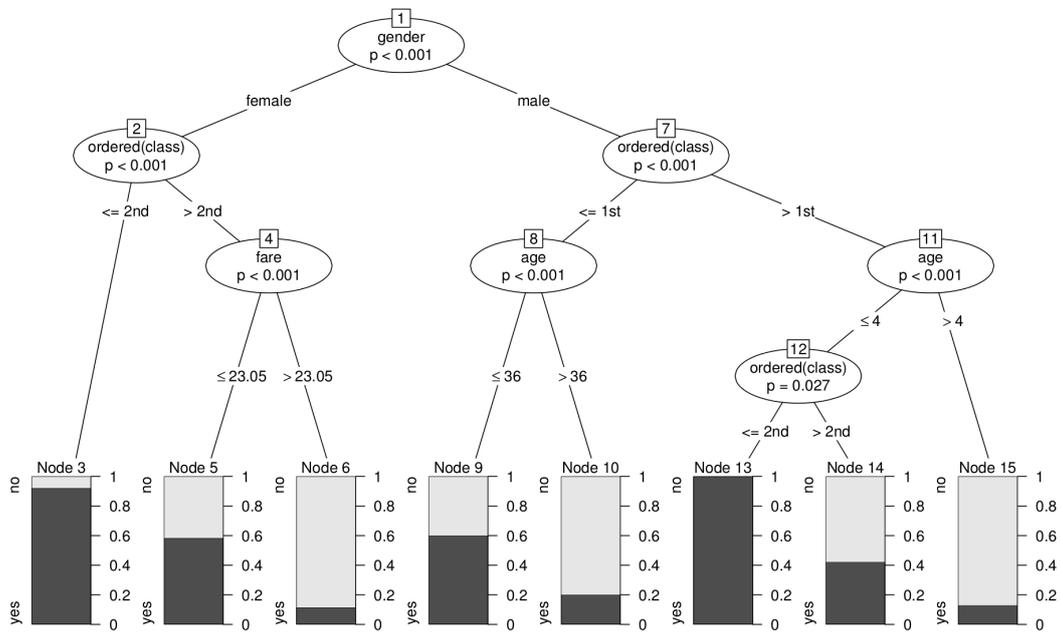


Figure 1. Tree representation of results from recursive partitioning for the RMS Titanic passenger data.

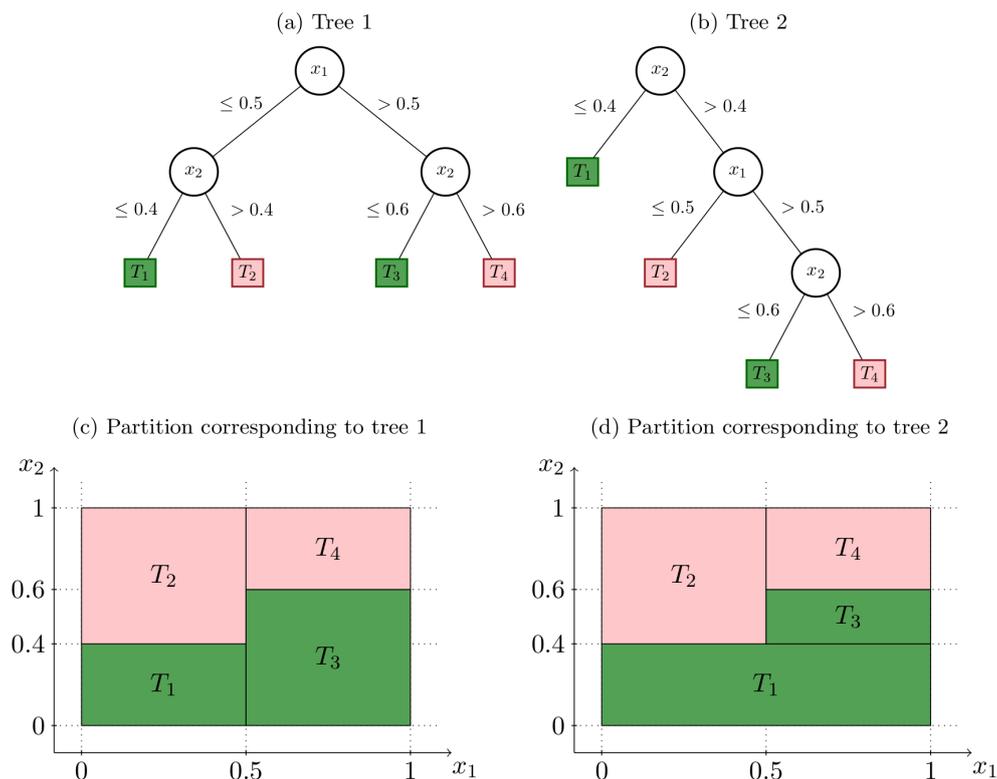


Figure 2. Examples of different tree structures, but equivalent partitions and interpretations.

tree, or the numerical variable `fare` was split twice in the original tree at 15.02 and 23.05 and at 23.05 in the bootstrap tree. Thus, many splits appeared in both trees, only the order and the cutpoints for numerical variables were slightly different.

As Turney [12] elaborates in his work, two trees that are structurally different can be logically equivalent. This means that two trees can lead to very similar or even the same interpretation although their structures (in particular the order of the splits) look very different. To illustrate this principle, we consider two hypothetical trees for a simple classification problem with two classes and a two-dimensional predictor space. Note however, that the statement also holds for any type of response variable and also for predictor spaces with more dimensions. Figure 2 shows two trees (upper row) and representations of the corresponding partitioning of the feature space (bottom row). In the illustration the predicted class in each terminal node is indicated by the colors red and green. According to the figures in panel (a) and (b), the tree structures differ by the split variable in their root node, their path structure and the sequence of split variables in the paths between the root node and the leafs. Yet, though the two trees are structurally different, the predictions are equivalent for any point in the predictor space. By mentally merging the two partitioning representations, it becomes evident that the two trees have identical splits, that only appear in a different order in the tree representation.

To assess whether a tree is stable or not, it is therefore principally important to investigate the stability of the splits, rather than the stability of the entire tree structure. From a single tree representation it is not possible to identify which splits are stable. It is possible, however from an ensemble of trees, e.g., generated by resampling from the original data. From the ensemble, the stability of the splits can be assessed by investigating the variable selection frequency and the cutpoint variability.

### 3 Measuring variable selection and cutpoint stability

In the following we will outline what steps are necessary from a conceptual point of view to assess the stability of variable and cutpoint selection in trees. Subsequently, these steps will be illustrated for a binary classification tree modeling survival vs. non-survival on the RMS Titanic.

The first step to assess stability is to draw several samples from the original data. The second step is to compute the descriptive measures and graphics provided in our toolkit over all samples. The options implemented in the package for generating samples in the first step are bootstrap sampling (sampling with replacement), subsampling (sampling without replacement),  $k$ -fold sample splitting (partitioning the original data into  $k$  equally sized samples), leave- $k$ -out jackknife sampling, or further user-defined strategies. Since each option has its specific peculiarities, they will likely generate different results. For the further illustration we will focus on bootstrap sampling, which is most widely used and was chosen as the default option in the function `stabetree()` that performs the resampling and extracts the required information from the ensemble for further analysis:

```
R> library("stablelearner")
R> data("titanic", package = "stablelearner")
R> m <- ctree(survived ~ gender + age + fare + ordered(class) + embarked +
+   sibsp + parch, data = subset(titanic, class %in% c("1st", "2nd", "3rd")))
R> s <- stabetree(m, B = 500)
```

The function `stabetree()` requires a tree-based model object that either inherits from class `party` (like, e.g., the result of `ctree()` or `glmtree()`) or can be coerced to it (like, e.g., the results of `rpart()` or `J48()`). Additionally, parallelization can easily be utilized with a convenience option for multicore computation based on `parallel` (for platforms that support this).

In the remaining part of this section, descriptive measures and graphical illustrations are introduced for investigating the stability of the splits, specifically for the variable and the cutpoint selection. First, the measures will be briefly discussed and then illustrated for the Titanic example.

#### Variable selection analysis

The aim of the variable selection analysis is to investigate *whether* variables that are selected for splitting in the original tree are also consistently selected for splitting in the resampled data sets. Furthermore, it can be compared *how often* (on average) a variable is selected within the original tree and the repetitions, respectively.

The first descriptive measure is simply the relative frequency of selecting variable  $x_j$  for splitting, computed over all repetitions in the procedure. Let  $b = 1, \dots, B$  denote the index for the repetitions and  $j = 1, \dots, p$  the index of the variables considered for partitioning. Further, let  $\mathbf{S} = \{s_{bj}\}$  be a binary matrix, where  $s_{bj} = 1$  if variable  $x_j$  was selected for splitting in repetition  $b$  and 0 otherwise. Then, the relative *variable selection frequency* is computed by  $100 \cdot \frac{1}{B} \sum_{b=1}^B s_{bj}$  and is expected to be large (i.e., close to 100%) for those variables selected in the original tree, if the result is stable. The variable selection frequency can be illustrated graphically using a `barplot()` method that generates the barplot depicted in the left panel of Figure 3. The variables depicted on the  $x$ -axis are sorted in decreasing order with respect to their variable selection frequencies (here and in all the following graphical tools). The bars of variables selected in the original tree are colored in dark gray and the corresponding labels are underlined. Thus, from the plot we can infer that the variables gender, class, age, fare and sibsp were selected for splitting in the original tree. The height of the bars corresponds to the variable selection frequency depicted on the  $y$ -axis. The first two bars reach the upper limit of 100%, which means that the variables gender and class were selected for splitting in each repetition. The variable age, represented by the third bar, was selected slightly less than 100% (but still very often) over the repetitions. The variables fare and sibsp, represented by the fourth and the fifth bar, were selected in the original tree, but not as frequently over all repetitions. This indicates that the splits in those variables in the original tree must

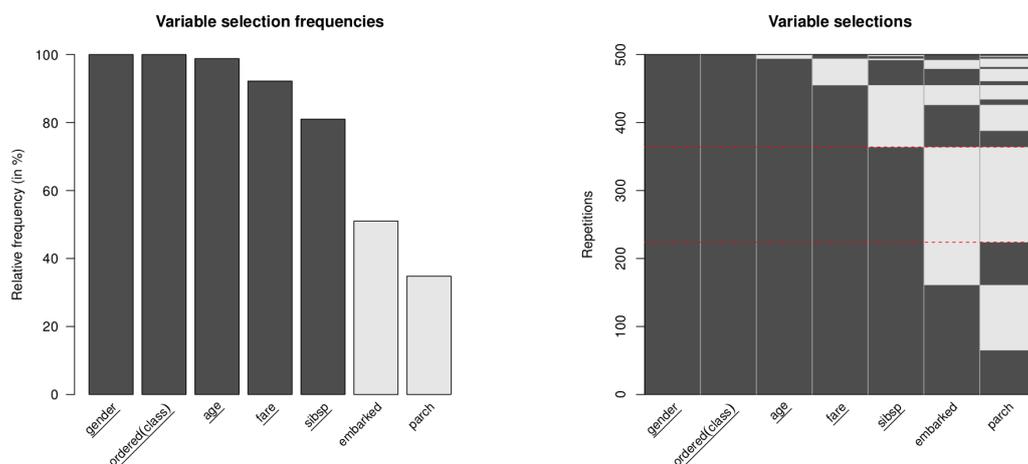


Figure 3. Graphical variable selection analysis.

be considered less reliable compared to the splits of the variable `gender`, `class` and `age`. The last two bars represent the variables `embarked` and `parch`, which were not selected in the original tree. They were selected for splitting in less than 50% of the repetitions. This indicates that although those variables seem to carry some information that is useful for predicting survival, they are not predominant. From a content perspective one may assume for this example that, over the repetitions, the variables `embarked` and `parch` occasionally acted as a proxy for the other variables in the data set.

The `summary()` method prints the corresponding table with the variable selection frequency (termed `freq`) in the first column for each variable. The second column (headed by an asterisk) indicates whether the variable was selected for splitting in the original tree:

```

          freq * mean *
gender      1.000 1 1.644 2
ordered(class) 1.000 1 2.578 3
age         0.988 1 2.316 2
fare       0.922 1 1.676 2
sibsp     0.810 1 1.132 1
embarked  0.510 0 0.638 0
parch     0.348 0 0.444 0
(* = original tree)

```

The third column in the table (termed `mean`) contains the values of another descriptive measure and denotes the average count splitting in variable  $x_j$  per tree. Let  $\mathbf{C} = \{c_{bj}\}$  be an integer matrix, where  $c_{bj}$  equals the number of times  $x_j$  was used for splitting in the tree for repetition  $b$ . Note that this number can be greater than one, because the same variable may be used for splitting in different parts of the tree. The *average variable split count* is computed by  $\frac{1}{B} \sum_{b=1}^B c_{bj}$  and is expected to be close to the count of splitting in variable  $x_j$  in the original tree. The last column in the table (also headed by an asterisk) indicates how many times the variable was selected for splitting in the original tree. For example, the variable `gender`, was used on average 1.644 times over all repetitions and twice in the original tree. It is possible, that the variable `gender` was often split on a higher level (and thus less often used for splitting) in the repetitions, as compared to the original tree. The reverse may be assumed for the variable `age`, which was on average more often used for splitting over the repetitions than it was used for splitting in the original tree. Similar interpretations follow from the information for the other variables.

Furthermore, we can investigate the combinations of variables selected in the various trees over the repetitions. This can be illustrated using the function `image()`. The resulting plot, that is illustrated in

the right panel of Figure 3, is a graphical illustration of the binary matrix  $\mathbf{S}$  that contains the variable selections over the repetitions. A fine grid of rectangles is drawn for each element in  $\mathbf{S}$ , which are colored dark gray if  $s_{bj} = 1$  and light gray if  $s_{bj} = 0$ . The repetitions (illustrated in the  $y$  direction) are ordered such that similar combinations of selected variables are grouped together. The combination of variables used for splitting in the original tree is marked on the right side of the plot using a thin solid red line. The area representing the combination is additionally enclosed by two dashed red lines. Notice that this is also the most frequent combination of variables selected over all repetitions. Repetitions that included additional variables beyond the combination in the original tree are illustrated below the marked area. Hence, we can deduce from the illustration that the variables **embarked** and **parch** were sometimes additionally used for splitting. In the replications above the marked area some splitting variables from the original tree were substituted with other variables.

## Cutpoint analysis

The variable selection analysis showed that there are some variables which are consistently used for splitting, indicating that those variables are more relevant in predicting survival than others. However, even when the same variables are selected, the splits may still vary with respect to the cutpoints chosen for splitting. Therefore a further important step in assessing the stability of the splits is the analysis of the cutpoints, which provides more detailed information about the variability of the splits.

We suggest different graphical illustrations for analyzing the variability of the cutpoints for numerical, unordered categorical and ordered categorical variables. Using the function `plot()` these illustrations can be generated for all variables specified in the model. According to the type of variable the correct illustration is chosen automatically and the variable names are underlined if the variable was selected for splitting in the original tree. Figure 4 illustrates these plots for the variables in the Titanic passenger data set.

To analyze the cutpoints for ordered categorical variables, we suggest to use a barplot that shows the frequency of all possible cutpoints. Those are sorted on the  $x$ -axis by their natural order that arises from the ordering of the categories of the variables. Examples are given for the variables **class**, **sibsp** and **parch** in Figure 4. Additionally, the cutpoints chosen in the original tree are marked using a vertical dashed red line. The number above each line indicates at which level the split occurred in the original tree. For example, the cutpoint between the first and the second class is selected more than 500 times (the number of repetitions in this example). This means that for some repetitions the split appeared several times in different positions in the same tree (for example in parallel branches). However, the passengers were split even more often between the second and the third class. The illustration indicates that the observations were consistently split by their class affiliation over the repetitions to predict survival of the passengers. The cutpoint in the variable **sibsp**, on the other hand, was less stable. Although the variable was quite frequently selected for splitting, the variable was often split between lower categories over the repetitions as compared to the original tree. The variable **parch**, which was not used in the original tree, was split only few times between the lower categories and can thus be considered as not very relevant.

To analyze the partition for unordered categorical variables (avoiding ambiguities by using the term “partition” rather than “cutpoint” here), we suggest to use image plots, as illustrated for the variables **gender** and **embarked** in Figure 4. When using binary splits, observations with the same categories are partitioned into the left or the right daughter node. Thus, the categories are assigned to the left or to the right branch of the split, respectively. For visualizing the partitions over the repetitions, categories that are directed to the same daughter node are illustrated by the same color. For the variable **gender**, there is only one possible split between the two categories **Female** and **Male**. The plot illustrates, however, that this split occurs many times (more than 500) over all repetitions, which underscores the relevance of the split. The combination of categories that represent a partition as it occurred in the original tree, is marked on the right side of the plot using a thin solid red line. The area representing the partition is additionally enclosed by two dashed lines (this is a little hard to see here, because the binary variable **gender** only offers one possible partition). Furthermore, the number(s) on the right side of the marking also represent(s) the level(s) of the corresponding split(s) in the structure of the original tree. The two

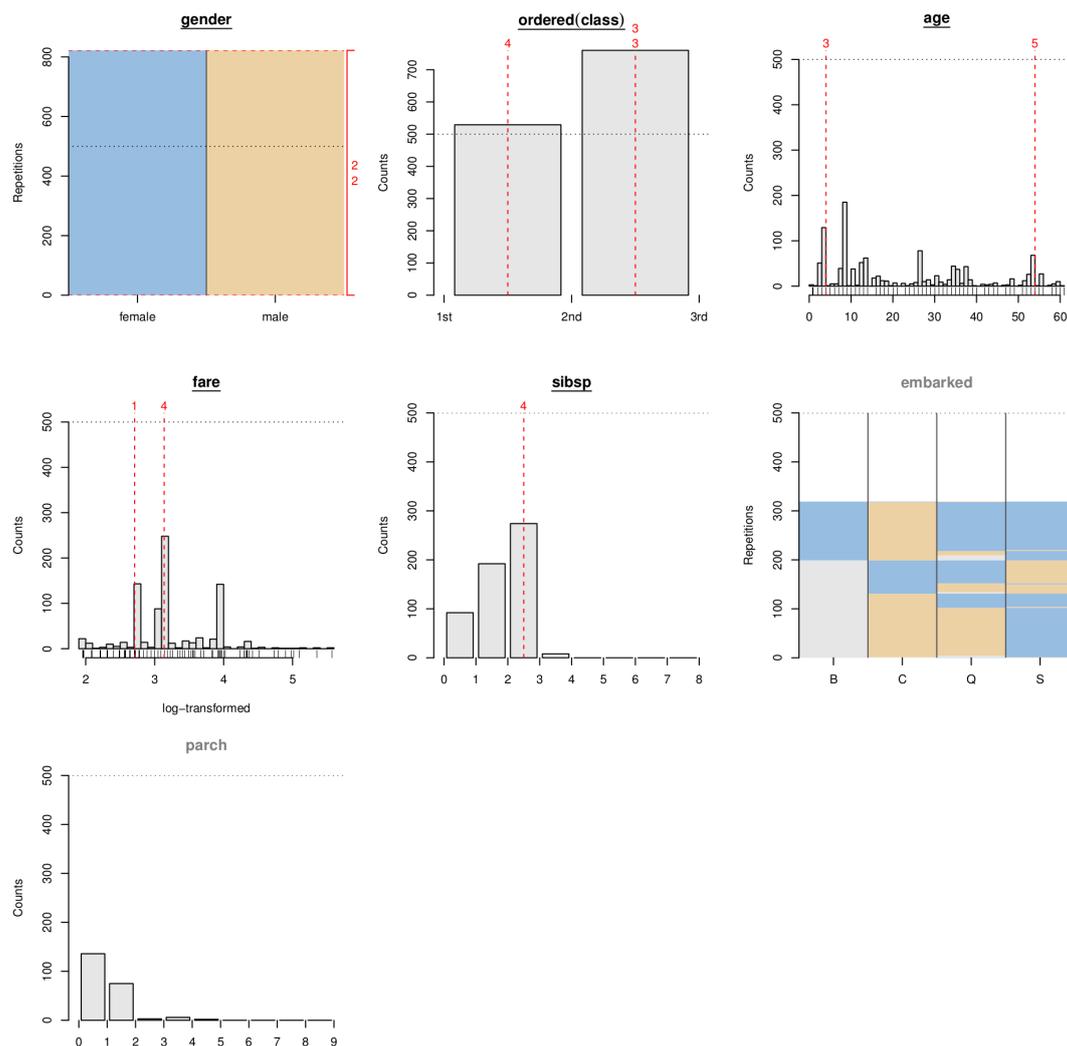


Figure 4. Graphical cutpoint analysis.

numbers on the right side of the illustration for the variable **gender** in Figure 4 indicate that **gender** was split twice on the second level in the original tree.

The plot becomes more detailed for variables with more than two categories such as the variable **embarked**. This variable, however, was not used for splitting in the original tree. Nevertheless it was used relatively often for splitting over all repetitions. In this illustration the additional color light gray is used when a category was no more represented by the observations left for partitioning in the particular node. The partitions over all repetitions are ordered such that equal partitions are grouped together. The most frequent partitions are  $[C, Q]$  versus  $[S]$  and  $[C]$  versus  $[B, Q, S]$ . Since passengers from the different classes tended to embark in different cities (e.g., most third class passengers embarked in Southampton), the variable **embarked** may in some repetitions (but not in the original tree) have been used as a proxy for the variable **class** in parts of the tree.

To analyze the cutpoints for numerical variables, we suggest to use a histogram, as illustrated for the variables **age** and **fare**. According to the distribution illustrated for the variable **age**, the cutpoints selected over the repetitions spread over the complete range of possible cutpoints. Although some cutpoints were selected more frequently than others, there were no unique cutpoints that were selected over

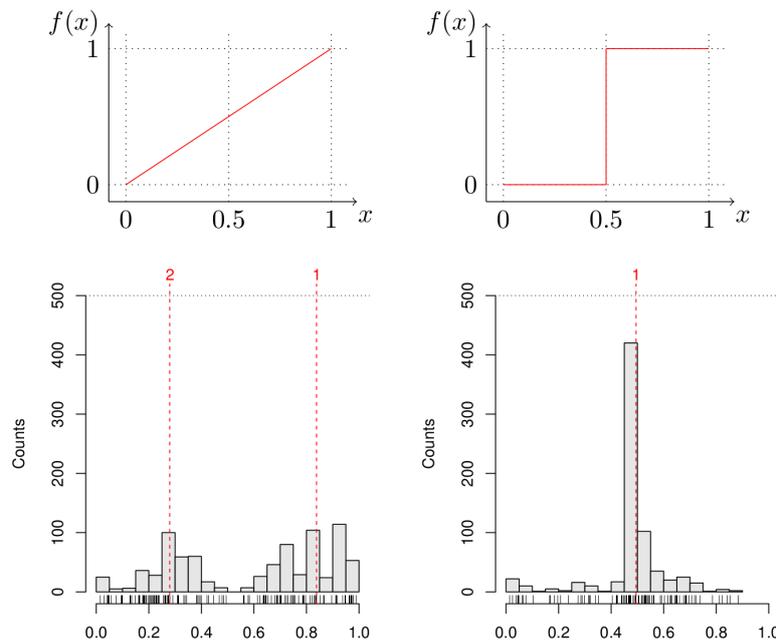


Figure 5. Cutpoint analysis for artificial regression problem.

most repetitions. The selected cutpoints of the variable **fare** are illustrated on a logarithmic scale in Figure 4, as it makes the picture easier to read. Again, the cutpoints selected over the repetitions spread over the complete range of possible cutpoints. However, the cutpoints selected in the original tree match two of three distinct peaks in the histogram and can be considered slightly more stable as compared to the cutpoints within the variable **age**.

From a conceptual point of view, the cutpoint pattern reflects the underlying functional shape. Due to the recursive nature of trees, smooth functions need to be approximated by several splits while piecewise constant step functions can be described straightforwardly by individual splits (see also [11]). This is illustrated in Figure 5. The upper left panel illustrates a linear relationship. To approximate this functional form, a tree algorithm will split the variable several times at different cutpoints. Altering the data would thus very likely lead to a different cutpoint. For a piecewise constant function like the one illustrated in the upper right panel of Figure 5, on the other hand, the functional form is captured by a single split, that is relatively easy to detect and will not change much if the data are altered.

To further demonstrate how the cutpoint stability plots can reflect the underlying functional form, we have simulated 500 observations from the model  $y = f(x) + \varepsilon$  for each of the two functions displayed in the top row of Figure 5. The variable  $x$  was sampled from a uniform distribution  $\in [0, 1]$  and  $\varepsilon$  was sampled from a standard normal distribution. In the bottom row of Figure 5 the stability of the cutpoints for the variable  $x$  is illustrated for the two artificial examples. As expected, the identification of a stable cutpoint failed for the example with the linear relationship (see lower-left panel). In the example with the piecewise constant relationship, however, the cutpoint at 0.5 was correctly recovered over most repetitions (see lower-right panel). For the Titanic example illustrated in Figure 4 this means that the cutpoints selected in the original tree for the variable **age** are rather unstable and should not be overinterpreted because the underlying functions seems to be smooth rather than piecewise constant. The cutpoints selected for the variable **fare** are slightly more stable.

To sum up, the stability analysis of the binary classification tree fitted for the Titanic data revealed that many splits in the original tree illustrated in Figure 1 were rather stable, but some parts were quite variable. First, the splits of the variables **gender** and **class** can be considered as important and stable. Second, the splits of the variable **age** are ambiguous, although the variable is definitely relevant

for predicting survival of the passengers. Furthermore, the splits of the variable `fare` are fairly stable, but the variable was a few times not selected for splitting over the repetitions. Thus, if the data were altered slightly, the variable might also have been omitted for splitting in the original tree. And finally, the split of the variable `sibsp` is least stable and should not be overinterpreted.

## 4 Discussion

In this paper we have presented a toolkit of descriptive measures and graphical illustrations that can be used to investigate the stability of the variable and cutpoint selection in models resulting from recursive partitioning. It was demonstrated how the tools are used and illustrated how intuitive they are by a real world data set. The analysis revealed that many aspects of the fitted tree were rather stable, but some parts were quite variable. Notice that the toolkit is not limited to classification trees, but can also be used to investigate the stability of regression trees or model-based trees. It was further illustrated that clear cutpoints from piecewise constant functions in the underlying data generating process, can be identified using the proposed graphics for the cutpoint analysis.

To acknowledge some limitations associated with the tools it should be mentioned, that they produce less meaningful results for very large trees with many splits. If the structure of the underlying data generating process is complex, the sample size or the number of predictors is large, it can become tedious to interpret a tree. Assessing the variable selection and cutpoint stability of such trees is computationally very intensive and the result might be unclear. However, the complexity of a tree can be reduced by modifying the settings (i.e., the pruning rule or the stopping criteria) of the recursive partitioning algorithm. Furthermore one should always be aware that any resampling scheme can only mimic what would happen if a new sample could be drawn from the population. And finally, the proposed tools do not assess the predictive stability of trees, which is another important aspect for their interpretation, as we briefly saw in Section 2. This aspect will be addressed in future research.

# Bibliography

- [1] Bar-hen, A., Gey, S., and Poggi, J.-M. (2015). Influence Measures for CART Classification Trees. *Journal of Classification*, **32(1)**, 21–45.
- [2] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. Belmont: Wadsworth.
- [3] Breiman, L. (1996). *Heuristics of Instability and Stabilization in Model Selection*. *The Annals of Statistics*, **24(6)**, 2350–2383.
- [4] Briand, B., Ducharme, G. R., Parache, V., and Mercat-Rommens, C. (2009). A Similarity Measure to Assess the Stability of Classification Trees. *Computational Statistics & Data Analysis*, **53(4)**, 12081217.
- [5] Hothorn, T., Hornik, K., and Zeileis, A. (2006). *Unbiased Recursive Partitioning: A Conditional Inference Framework*. *Journal of Computational and Graphical Statistics*, **15(3)**, 651–674.
- [6] Hothorn, T. and Zeileis, A. (2015). *partykit: A Modular Toolkit for Recursive Partytioning in R*. *Journal of Machine Learning Research*, **16**, 3905–3909.
- [7] Kopf, J., Augustin, T., and Strobl, C. (2013). *The Potential of Model-Based Recursive Partitioning in the Social Sciences – Revisiting Ockham’s Razor*. In McArdle, J. J. and Ritschard, G. (Ed.), *Contemporary Issues in Exploratory Data Mining in the Behavioral Sciences* (pp. 75–95). New York: Routledge.
- [8] Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. New York: Springer-Verlag.
- [9] Miglio, R. and Soffritti, G. (2004). The Comparison Between Classification Trees Through Proximity Measures. *Computational Statistics & Data Analysis*, **45(3)**, 577–593.
- [10] R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- [11] Strobl, C., Malley, J., and Tutz, G. (2009). *An Introduction to Recursive Partitioning: Rationale, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random Forests*. *Psychological Methods*, **14(4)**, 323–348.
- [12] Turney, P. (1995). *Technical Note: Bias and the Quantification of Stability*. *Machine Learning*, **20(1–2)**, 23–33.