

# EXPLORING STATISTICAL FORESTS

Simon Urbanek, Department of Computer Oriented Statistics and Data Analysis,  
University of Augsburg, Germany (Simon.Urbanek@math.uni-augsburg.de)

**Key Words:** Classification and regression trees, model assessment, visualization

## Abstract:

Trees are a valuable way of displaying structure in datasets, especially for classification problems. Improved classification results can be achieved using forests of trees. Adding various visualization methods and interactive tools for analysis of individual trees and of whole forests gives complementary insight into the data. This paper describes different views and methods to analyze tree forests as implemented in our prototype software, KLIMT (KLassification - Interactive Methods for Trees).

## 1. Introduction

Tree structured models have been used in many different fields, such as botany or medicine, even before their value was discovered by statisticians, especially for classification problems. In the first major work about trees in statistics Breiman et al.[1] concentrate on building one, possibly optimal model. Since then different methods have been developed to improve the prediction accuracy of tree models. A combination of results of several trees in a final classifier is often used, for example in bagging[2] or boosting[3]. The drawback of such methods is the loss of interpretability, one of the biggest advantages of trees. Therefore attempts have been made to analyze entire forests of trees and retrieve some information about the underlying data. This can lead to selection of one model, but also to multiple distinct groups of trees explaining dependencies. In practice such information is often more valuable than just a hard-to-interpret classifier.

Purely numerical approaches to forest analysis has not yielded good results. Several metrics have been defined to measure the similarity between trees, such as the fit metric, the partition metric, the matching coefficient and the tree metric. Each of them captures only one of many properties of a tree model and the comparison of many trees leads to a high-dimensional space. The graphical approach of using MDS delivers coarse results, since the mapping of the distances of the tree distance matrix to inter-point Euclidean distances is in most cases poor[4].

Our approach is to use interactive graphics to explore the forest and extract information about the models under consideration and the underlying data. We present various plots that can be helpful for model comparison and detecting clusters of similar trees.

## 2. Visualization of Forests

In the following we want to illustrate analysis of a forest based on the Wisconsin cancer data. The tree models were generated using bagging. The methods presented here can be equally applied to trees generated by other methods, such as boosting or random forests, but they are not explicitly described here.

For a single classification tree the class prediction is the information directly obtained by applying the tree model to a dataset. The misclassification rate is therefore the most natural and commonly used measure of fit for a particular tree. In practice it is only a coarse measure since a tree forest many contain many trees with the same misclassification rate but very different structure. Furthermore since it is a global property of the tree it cannot be used for any conclusion concerning individual nodes or variables.

One method to avoid this shortcoming is to consider the deviance gain for each node. The CART tree growing algorithm is based on maximizing the decrease of impurity for each node. In our case we used the entropy measure of impurity which corresponds to maximizing deviance gain. Although tree deviance is a global measure, the definition of deviance gain allows us to estimate the quality of each split individually. Provided the same test set is used, it is possible to compare deviances of different trees and different nodes. This allows us to analyze an entire forest of trees.

In KLIMT it is possible to load several trees and apply them to a single dataset. The results can be stored as *forest data*. Each case corresponds to an inner node and contains several variables including the tree which the node is part of, the population of the node, the deviance gain, the tree deviance and the split variable. Extended forest data also contain additional variables such as the deviances and populations of the child nodes (left/right) and splitting

values. Forest data can be generated for both classification and regression trees. The emphasis here is on the quality of each split and the relations between variables, not on the cases of the test dataset.

First application of this forest data is to assess variable importance. The primary interest is which variables contribute most to the model and which can be considered less important. The most naïve approach to estimate variable importance would be to count the occurrences of each variable in all models. This can be graphically displayed in a barchart as in the left plot of fig. 1. According to this plot **B.Nuclei** is the most frequently used variable. Obviously this is not a very good approximation to the variable importance, since variables used early in the tree have much more weight than variables used locally in lower nodes.

Deviance gain was the method of choice to measure contribution of a split and since deviance is additive we can use it as a weight. An overview of the global variable importance can be obtained by plotting a barchart of variables weighted by the deviance gain as shown in the right plot of fig. 1.

The first two layers of the trees, that is first two splits when classifying a case, are highlighted in both plots. Note that both charts are sorted by value and the order is different in each of them. **B.Nuclei** is obviously the most frequently used variable, but only half of its occurrences contribute significantly to the deviance reduction. The weighted barchart points out **Unif.Cell.Size** as the most important variable.

This global view does not tell us if there are more groups of trees or whether any variables show a local behavior. It is possible that all trees use **Unif.Cell.Size** as the first split with large deviance gain, but it is also possible that all of the important variables are used only with different frequencies although they have almost the same gain. We need a more refined approach.

One idea is to use a weighted fluctuation diagram of variables vs. trees with deviance gain as weight. The resulting plot is shown in fig. 2. Trees are plotted on the  $x$ -axis and variables on the  $y$ -axis. Each rectangle corresponds to a variable in a single tree. The area of the rectangle corresponds to the sum of deviance gains of the splits which use the variable. Large boxes represent variables with most deviance gain in the tree.

Selecting different nodes in an interactive environment allows us to see the corresponding gains in the fluctuation diagram. In fig. 2 green highlighting corresponds to the first two splits.

Clearly **Unif.Cell.Size** and **B.Nuclei** are the key variables, both are always used, **Unif.Cell.Size** and

**B.Nuclei** mostly in first two splits. **Clump.Th** is used in lower splits only, but still contributes consistently to the explanation. Interestingly whenever **Unif.Cell.Shape** was used as first split, **Unif.Cell.Size** had very little gain, which is a sign of association of those two variables since masking occurs.

The conclusion of this analysis would be that **Unif.Cell.Shape** is masked by **Unif.Cell.Size**, which can be verified by removing each variable and growing a new tree. It turns out that both trees have very similar structure and misclassification rate. **Unif.Cell.Size** and **B.Nuclei** cooperate well and are used mostly in the first two splits. **B.Chromatin** seems to draw most of its overall deviance gain from the single occurrence in the first split as can be verified by deselecting that single tree. Therefore the third most important variable is **Clump Th.** since it is used more often although only in lower parts of the trees.

The advantage of fluctuation diagrams is that linked highlighting at case level is possible, which is very valuable for interactive analysis. They are not designed for exact comparison of values, but merely for finding structures. If several hundreds of trees are involved fluctuation diagrams can no longer be used. More detailed analysis of lower tree layers with fluctuation diagrams is more difficult since only visually big changes in size are detectable. Zooming would partially solve this drawback but the danger of “getting lost” in the diagram would increase.

An alternative approach to display similar information in a different way is to use parallel coordinates plots (PCP). They are normally used to display highly multivariate data. In our case trees are the dimensions, variables are the cases and deviance gains are the values. This requires aggregation on tree level since we considered data at node level before and the aggregation was performed implicitly by weighted plots.

Based on the same data as the plots before, the PCP is shown in fig. 3. Each line corresponds to the total deviance gain of a single variable, trees are represented by the parallel axes. Additionally the top line represents the sum of deviance gains of all inner nodes of the tree, which is equal to the total deviance of the tree minus the remaining deviance in the terminal nodes. Therefore higher total gain corresponds to smaller remaining deviance and hence better separation. All coordinates are plotted using a common scale. The three most important variables mentioned in previous paragraphs are selected and denoted by different colors.

The cooperation of **Unif.Cell.Size** and **B.Nuclei** is clearly visible. In most cases they appear in tandem.

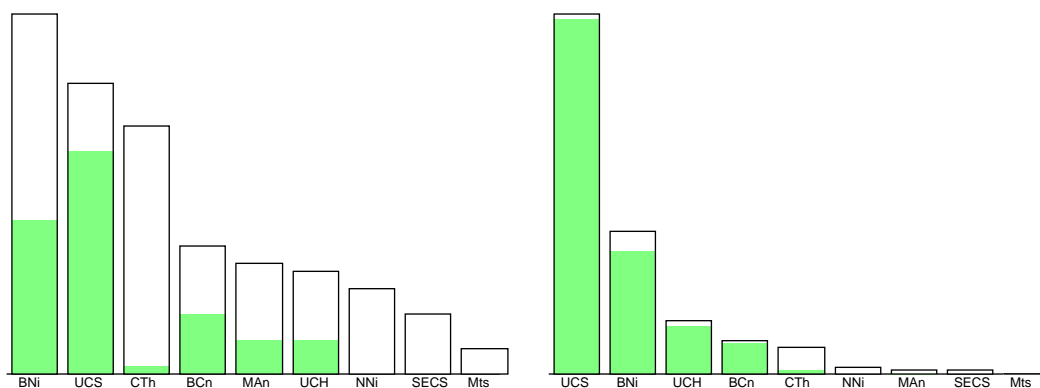


Figure 1: Barchart and weighted barchart of variable with deviance gain as weight. First two splits of all trees are selected.

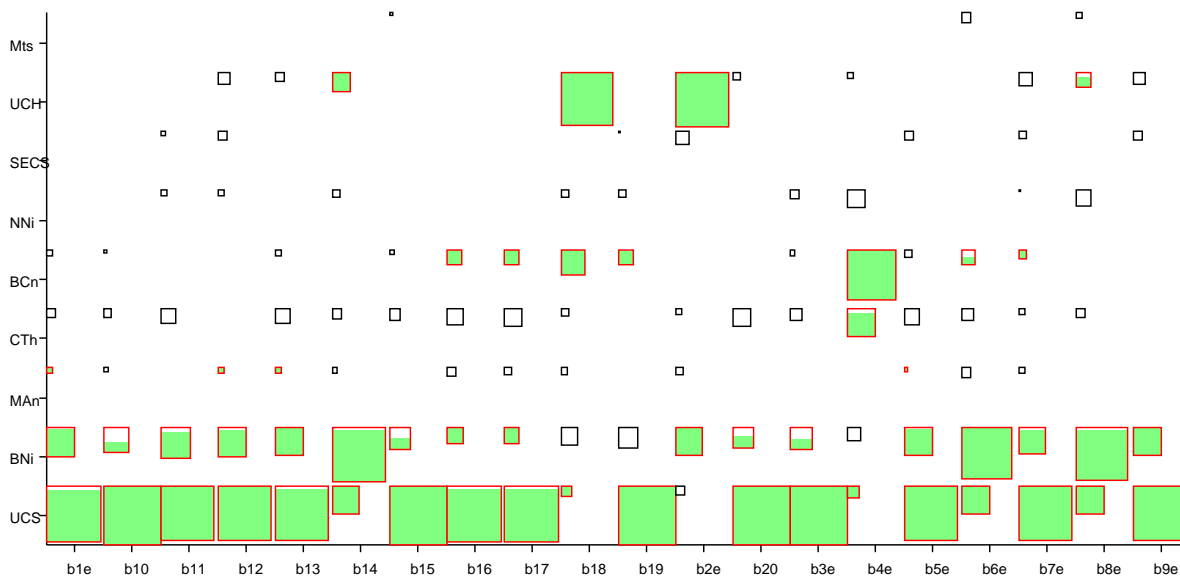


Figure 2: Weighted fluctuation diagram of variable vs. tree with deviance gain as weight. First two splits in each tree are selected.

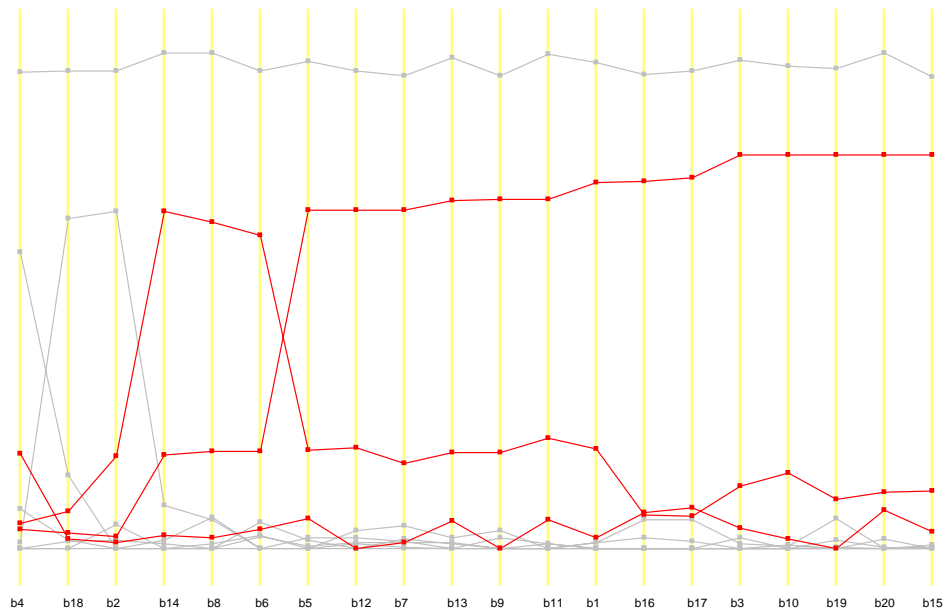


Figure 3: Parallel coordinates plot of deviance gain and total gain. Selected are variables `Unif.Cell.Size`, `B.Nuclei` and `Clump.Th`.

Together they have almost always the same amount of deviance gain. It is noticeable that a higher gain of a single variable results in suppression of other variables and in this case also in slightly lower total deviance. A variable used in the same position in more trees, especially the first one, has usually always the same deviance gain. Additional gain of the same variable can be therefore only explained by additional splits lower down the tree. In fact all trees on the right side use `Unif.Cell.Size` more than once. In many cases this leads to less powerful trees since the information content of a variable decreases by each split. Sequential splits on the same variable perform well only in cases with very clear multiple structures.

Although `Unif.Cell.Size` is used in the first split in most trees, it appears that trees with `B.Nuclei` split followed by `Unif.Cell.Size` have the highest deviance, see trees `b14` and `b8`. This may be also due to the fact that `Unif.Cell.Shape` is exceptionally used in addition to `Unif.Cell.Size` and is not masked completely. It turns out that each of these variables that otherwise mask each other is used in a separate branch split by `B.Nuclei`. This way information on each variable is used where it helps more. This may improve the model, but whether it is desirable depends on the application. In practice it is sometimes very ex-

pensive to measure one additional variable and since they are highly correlated it is preferred to use only one of them.

PCPs allow us to add the total deviance gain to the plot, which was not possible in the fluctuation diagrams. It is also easier to distinguish subtle differences in values, especially when the values of the compared variables are near each other. Another advantage of PCPs is the ability to handle large forests. Fig. 4 displays a deviance gain PCP of 200 trees generated by bootstrapping from the *cancer* dataset with `unif.Cell.Size` selected in red and `B.Nuclei` in blue.

As in the previous figure the three distinct groups of trees are clearly visible. The largest group on the right side features `Unif.Cell.Size` as the variable with greatest deviance gain. The left group, almost a third of the trees, is dominated by `Unif.Cell.Shape` which confirms the masking hypothesis. Finally a small central group is formed by trees with primary split on `B.Nuclei`. Just right beside this group there is a whole bunch of trees which are clearly dominated by the two variables `B.Nuclei` and `Unif.Cell.Shape`.

A very important factor in PCPs is the order of trees. Fig. 5 illustrates that a poorly chosen order can make the plot unusable. The order of trees is given by their total deviance in that plot. At a first

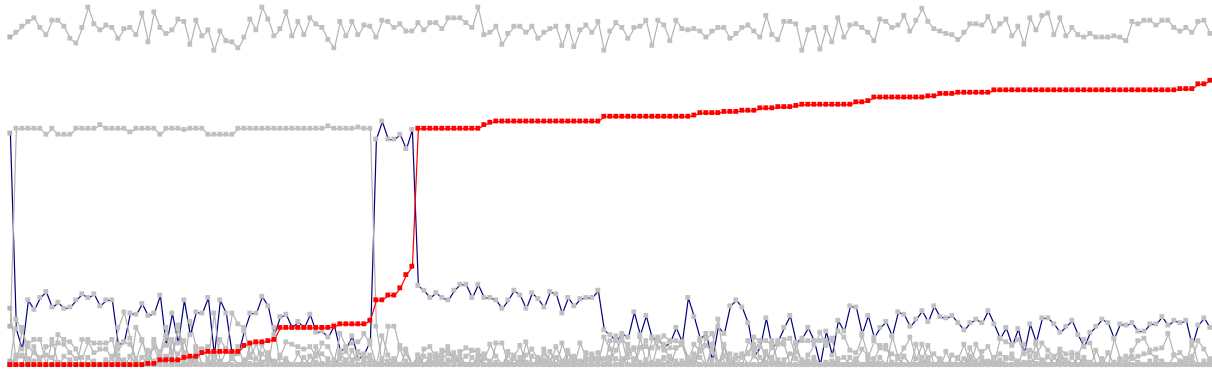


Figure 4: Parallel coordinates plot of deviance gain and total deviance of 200 trees. Selected are variables Unif.Cell.Size (red) and B.Nuclei (blue).

glance this might be a natural choice for the order of the trees, but it is basically impossible to detect any structure from a plot with such an order. There may exist special cases where some pattern is visible, but they will be rather rare. In most cases the variability of the total deviance is fairly low and therefore the order is partially random.

There are several approaches to determine an useful order of the trees. In figures 3 and 4 the primary order is given by the deviance gain of the dominant variable. For smaller forests it is possible to re-sort the trees manually in order to highlight patterns. For larger forests where simple sorting doesn't yield the desired results more sophisticated methods like clustering may be used.

### 3. Discussion

We have illustrated the usage of interactive plots to analyze a forest of trees. The individual trees were created by bootstrapping, but any other method of generating trees will work as well. These trees can be loaded in KLIMT for individual analysis and afterwards the forest data consisting of information about each node and tree can be generated. Using weighted barchart and fluctuation diagrams together with linked highlighting allows a thorough analysis of the forest down to per-case level. For larger forests parallel coordinate plots can be used, with trees as dimensions and deviance gain per variable as cases. PCP plots in this paper were generated by CAS-SATT, all other plots by KLIMT.

Interactive exploration of forests can provide additional information about the underlying problem.

Such information enhances the interpretability of the models by finding logical relations between various groups of trees. If desired it is also possible to select an optimal model manually. An extension to this method can consist of providing misclassification information for the interactive framework and linking to the already present visualization of individual trees. Interactive misclassification plots, where all misclassified cases for a tree are plotted, and misclassification frequency plots would provide helpful tools for this purpose. This would allow to concentrate the analysis on cases that tend to be misclassified, showing more information about this group.

Another enhancement especially in the assessment of variable importance would be to consider measures of importance even for variables not used in a split. The CART algorithms pick an optimal variable, but the deviance gain for other variables can be still high, only just not high enough to “beat” the chosen variable. Depending on the software used for tree generation such information may be available as well and could be aggregated to produce methods of importance assessment that are not affected by masking. Such analysis is not very easy, since choosing a different variable for a split also changes the structure of the following branch, rendering aggregation a non-trivial task.

Other common methods such as randomization or interactive removal of variables can be used to obtain different trees to be analyzed with the described methods. This would make it possible to concentrate on variables of interest and check the stability of the tree modes. Such analysis would fit well into the in-

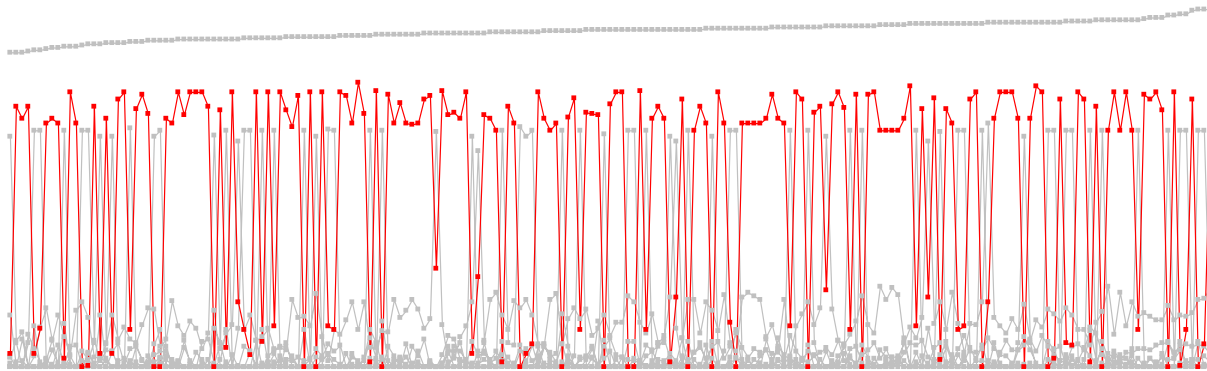


Figure 5: Same plot as in fig. 4 but with axes ordered by total deviance gain.

teractive framework, since the user can concentrate on trees and variables of interest to perform more detailed analysis.

## References

- [1] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Wadsworth, 1984.
- [2] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, p. 123/140, 1996.
- [3] Y. Freund and R. E. Schapire, “Experiments with a new booting algorithm,” in *13th International Conference on Machine Learning*, 1996.
- [4] H. Chipman, E. George, and R. McCulloch, “Extracting representative tree models from a forest,” working paper 98-07, Department of Statistics and Actual Science, University of Waterloo, 1998.