

Different ways to see a tree - KLIMT

Simon Urbanek

Department of Computer Oriented Statistics and Data Analysis,
University of Augsburg, Germany

Abstract. Recursive partitioning trees offer a valuable tool to analyze structure in datasets. There are many ways to display various structures contained in a tree. This paper describes different means of visualization of a tree with our prototype software, KLIMT (Klassifikation - Interactive Methods for Trees), for interactive graphical analysis of trees.

Keywords. classification and regression trees, visualization, interactive software, exploratory data analysis

1 Introduction

The value of trees is recognized not only in statistics, but also in many different fields such as machine learning, botany and medical decision-making. Trees provide valuable way of displaying structure in datasets. One of the major advantages of tree-based models is the fact, that their interpretation is quite intuitive and the structure often easy to explain. The models themselves impose hardly any restrictions on the underlying problem and dataset (Breiman [1984]).

Today it is fairly easy to grow many trees using various algorithms, such as the Bayesian approach (Chipman, George and McCulloch [1998]) or greedy algorithms (Ripley [1996]). In order to be able to choose a possibly best model we need to work with the tree, analyze it and explore it. Static tree plots display only one view of a tree, but dependencies or special cases can better be detected with multiple views.

KLIMT is an interactive software for exploratory data analysis of tree models. It is a stand-alone application with interfaces to R/S/S-plus software packages to allow seamless integration. The bidirectional interface is based on the RSJava package from Omegahat and provides similar features as the ggobi/R model described by Temple Lang and Swayne [2001]. For platforms unsupported by Omegahat a flat file interface can be used. Both interfaces allow to start KLIMT within R/S/S-plus by a simple `Klimt(tree, dataset)` command. Technical details about both interfaces and the application can be obtained from the documentation section of the KLIMT project homepage.

Many traditional plots are supported by KLIMT, such as histograms, bar-charts, scatterplots or boxplots. The interactive features include selection, queries, zooming, variation of displays, multiple views, pruning and linked highlighting for all plot types. The software meets most requirements for interactive software as described by Unwin [1999]. We have recently added new approaches to visualization of trees in KLIMT. The next section describes variations of the rather classical, hierarchical interpretation of a tree, whereas different alternative approaches of displaying a tree are presented in section 3.

2 Hierarchical views

Unlike many statistical plots trees don't have a fixed or exactly defined graphical representation. Even if we concentrate on the hierarchical structure the variety of different plotted trees is huge. Three different ways of drawing the same tree are shown in Fig. 1. We want to concentrate on the graphical properties and alternative visualization of trees here, therefore we have chosen an easily interpretable tree generated from the Fisher/Anderson Iris dataset with the *tree* library for R. The dataset is well known and the size of the tree allows us to illustrate the visualization aspects even in this static context without interactivity which is provided in KLIMT to allow analysis of complex tasks. Plot A was drawn with R's native *plot* function, all other plots were created by KLIMT.

Placement of the nodes is one of the most important factors. In some plots the distance between levels of the tree height is constant, in other plots it is proportional to some property of the tree such as the deviance gain in a split (see plot A). It is also possible to display all leaves on the same level to allow better comparison amongst them (plot C). Usually the child nodes are placed below their parents symmetrically, i.e. the centers of parent node and children nodes build an isosceles triangle, but various techniques, such as equidistant partitioning of each level can be used. KLIMT offers various placing algorithms, but also allows the user to freely modify the tree by dragging individual nodes or entire branches.

Not only the node placement varies, but also the means of displaying nodes and connecting them. It is possible to visualize additional information by using different symbols for nodes, e.g rectangles of various sizes, where the size is proportional to the population of a node (plots B and C).

Conventionally trees are plotted in top-bottom orientation, but for deep trees this may cause problems because the screen has usually more room in the horizontal than in the vertical direction. KLIMT alternatively allows the tree to be displayed in left-right orientation (see plot C) to avoid this shortcoming.

Overloading plots with information can offset the benefits of the plot, in particular its ability to provide information at a glance. If there is too much information attached to each particular node it is often not possible to display more than two levels of the tree on a screen or a page. Therefore additional tools are necessary to keep track of the overall structure in order not to get lost. Most of these tools, such as zoom, pan, overview window or toggling of labels are available in interactive context only.

Especially for analysis, visualization of additional information is required. There are basically two possibilities of providing the information: Integration of the information in the tree visualization or use of external linked graphics.

Direct integration is limited by the spatial constraints posed by the fixed dimension of a computer screen or other output medium. Its advantage is the immediate impact on the viewer and therefore easier usage. It is recommended to use this kind of visualization for properties that are directly tied to the tree, such as the node size or the criterion used for the growth of the tree.

External linked graphics are more flexible, because they are not displayed directly in the tree structure for each node separately, but are only logically linked to a specific node. Spatial constraints are less of a problem because one graphic is displayed instead of many for each node. The disadvantage of linked graphics is that they must be interpreted more carefully. The viewer has to bear in mind the logical link used to construct the graphics as it is not visually attached to its source (node in our case).

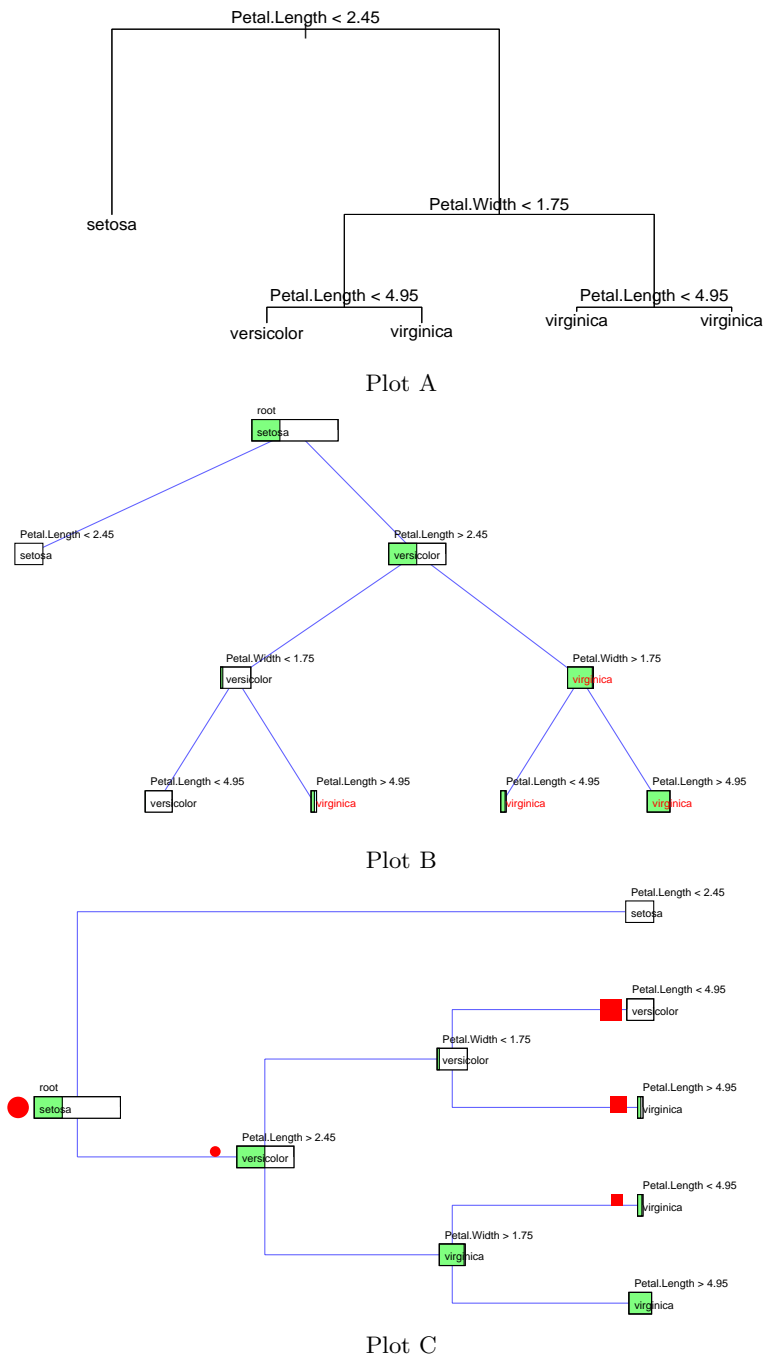


Fig. 1. Three different ways to visualize the same tree.

KLIMIT supports linked highlighting both on case level as well as on node level for linking between trees and plots. Directly related information such as node size or deviance gain are visualized directly in the tree structure.

3 Alternative views

The plot of the hierarchical structure of a tree is not the only way to see a tree. If only two measurement variables are involved in the decisions, a scatterplot of these two along with partitioning lines describes the entire tree. Each rectangular partition corresponds to a leaf and bears either a class name (for classification trees) or the predicted value (for regression trees). An example of such an *enhanced scatterplot* is given in Fig. 2.

If more than two variables are used in the splitting rules of the tree, the enhanced scatterplot represents only a two-dimensional projection of the measurement space, orthogonal to the axes. The interpretation of such a plot must be done more carefully, because splits on variables other than the plotted ones cannot be visualized. Still the scatterplot is helpful when examining individual splits and can be valuable for detection of additional structures in the data.

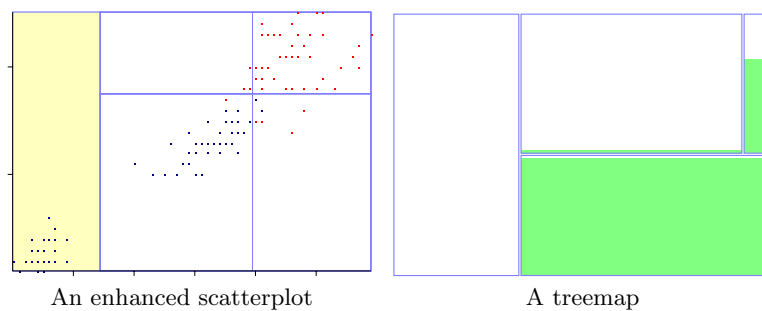


Fig. 2. Alternative ways to visualize a tree.

This approach is especially useful for continuous variables. Unfortunately the principle cannot be generalized for categorical variables, even in the two-dimensional case, because it is not always possible to construct a sequence of categories such that an n -way split of a node results in exactly n continuous partitions of the category space. This is true for any $n > 1$.

Instead of looking at the measurement space it is possible to consider the number of cases in each node. The corresponding graph shown in Fig. 2 is structurally similar to a mosaicplot and is often called a *treemap*. The plot is constructed as follows. The basis is a rectangular region representing all cases thus corresponding to the root. For each child of the root the region is partitioned horizontally into pieces proportional to the number of cases in each node. If the node is not a leaf, its space is now partitioned vertically according to the size of its children. This procedure is repeated recursively until a leaf is reached while the partitioning direction alternates between horizontal and vertical for each level as illustrated in Fig. 3.

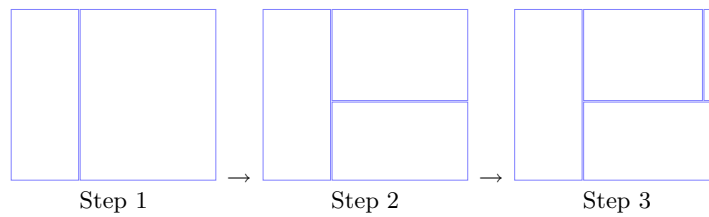


Fig. 3. Construction of a treemap.

The advantage of treemaps compared to scatterplots is that the limitation of two variables does not apply and even splits on categorical variables can be used. In a two-dimensional, continuous case the corresponding partitions in each plot can be mapped in a bijective fashion, but the area used by the same partition in each plot differs. In a scatter plot the size of a partition is given by the scale of the variables, whereas in a treemap the size is proportional to the number of cases in that partition.

When highlighting is applied, selected cases in the scatterplot are represented by points of a different color and/or size. In a treemap the number of selected cases is proportional to the volume of differently colored area within a partition, usually filled from bottom to top as if water was poured into the partitions. The proportion of the height of such highlighting to the total height of a partition is equal to the proportion of selected cases to the total number of cases in the partition. Therefore treemaps are useful for comparing proportions in the dataset, whereas enhanced scatterplots offer a way to recognize individual points, such as outliers or points at the edge of a split.

In order to directly compare leaves it is possible to use special plots that are a combination of treemaps and spineplots. The construction of the plot is done like a treemap where partitioning is not performed in alternate directions, but only in the horizontal direction. The resulting plot resembles a spineplot except that individual spines correspond to leaves of the tree and not classes of a variable. Therefore we refer to this plot as a *spineplot of leaves* as illustrated in Fig. 4.

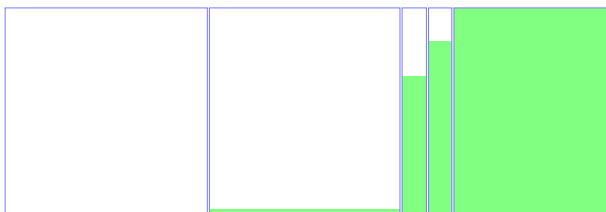


Fig. 4. A spineplot of leaves.

This view is especially helpful in conjunction with linked highlighting. The filled area is proportional to the number of cases highlighted in the corresponding leaf. In Fig. 4 all *virginica* species are selected allowing visual comparison of the absolute proportions in all nodes. Another property of spineplots is that relative proportions inside each spine correspond to the height of the filled area and hence are directly comparable. This means that both absolute and relative comparisons amongst leaves are possible at a glance.

The disadvantage of both spineplots of leaves and especially treemaps is the fact that identification of a certain node within the plot is somewhat difficult. Labeling as proposed for mosaicplots by Hoffman [2001] resulting in doubledecker-plots is not possible, because in general each level of the tree involves different variables in the splits. Direct interactive query remains the most appealing solution in this case.

KLIMT implements all three proposed plots. For spineplots of leaves the identification of individual spines is simplified, because the sequences of leaves in the hierarchical tree plot and the corresponding spineplots for leaves are identical.

3.1 Conclusion

A tree model can be observed from many different angles and each view displays various aspects of the model and underlying dataset. Beside the usual plots emphasizing the hierarchical aspect of a tree, our software KLIMT provides additional views such as enhanced scatterplots, treemaps and spineplots of leaves. All pictures in this paper were generated by KLIMT except for Fig. 1, Plot A. The combination of those plots and the interactive features of KLIMT such as hot linking of all views and plots, queries and immediate manipulations of plots, provides an analyst with a versatile tool for exploratory analysis of classification and regression trees.

References

- Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984) *Classification and regression trees*, Wadsworth
- Chipman, H., George, E. and McCulloch, R. (1998) "Bayesian CART model Search (with discussion)", *Journal of the American Statistical Association*, 93, 935-960
- Hofmann, H. (2001) *Graphical tools for exploration of multivariate categorical data*, BoD, Norderstedt
- Klimt project, <http://www.klimt-project.com>
- Omegahat project, <http://www.omegahat.org>
- Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*, Cambridge University Press
- Temple Lang D., Swayne D. F. (2001) "ggobi meets R: an extensible environment for interactive dynamic data visualization", *Proceedings of the 2nd International Workshop on Distributed Statistical Computing*, TU Vienna
- Unwin, A. R. (1999) "Requirements for interactive graphics software for exploratory data analysis", *Computational Statistics*, 14, 7-22
- Urbanek S. and Unwin, A. R. (2001) "Making trees interactive - KLIMT", *Proceedings of The 33rd Symposium on the Interface of Computing Science and Statistics*