

Making Trees Interactive - KLIMT

Simon Urbanek

su@uni-augsburg.net

Antony R. Unwin

antony.unwin@math.uni-augsburg.de

Department of Computeroriented Statistics and Data Analysis,
Augsburg University

Interface2001

Abstract

Trees are a valuable way of displaying structure in data sets. Adding interactive tools would make them even more valuable. This paper describes our prototype software, KLIMT (Klassifikation - Interactive Methods for Trees), for interactive graphical analysis of trees. The research is work in progress and there are many different possible options. What features do analysts want?

1 Introduction

The use of tree-based models is becoming more popular in statistics for various reasons, in particular the intuitive interpretation and good structure representation they offer. Trees have been used in many application areas, for example botany and medical decision-making. Today many popular software packages allow us to construct classification and regression trees. An example of a classification tree is given in Fig. 1. The tree has been built with the *tree* library in R with default parameters. The underlying dataset is a medical study about meningitis disease treatment¹. Several medical parameters of patients were recorded and the patients were classified by the success of treatment (*yes* = treatment was successful, *no* = treatment was not successful). The goal is to find a good explanatory model by analyzing the data with a tree. The visualization in Fig. 1 does not show much information about the data itself, such as the number of cases in a node, the distribution of classes or the deviance. Therefore any further analysis of the model is not possible without additional tools. There have been numerous attempts to include additional information in the tree, notably *mobiles* [10] or various enhanced trees by adding colors,

¹Total number of cases is 138. The dataset consists of four different variables (ALTER, FIE, ZZLQ, GRA) plus an outcome (classification) variable. There are 28 negative cases and 110 positive cases.

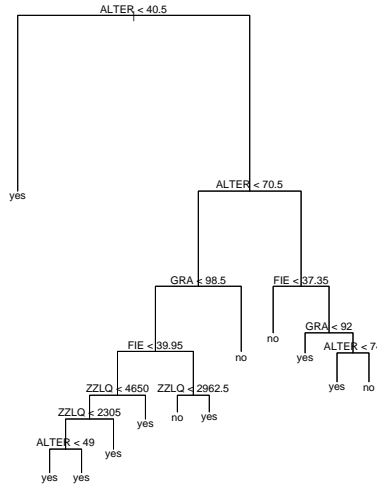


Figure 1: Classification tree as plotted by R.

histograms and descriptive information [2]. All these approaches have their limits, by adding too much information to a node the tree becomes cluttered with different symbols and loses its clarity. Some of these techniques are also feasible only for very small datasets because of screen space constraints. All these static representations of a tree show only one view of the data - dependencies or special cases are hard to detect without plotting multiple views.

The graphical display of tree analyses is a big advantage and it can be made an even bigger advantage by adding interaction. Unwin [8] describes the basic features which any interactive graphics software should offer: querying, zooming, rescaling, selection with linking and the use of multiple views. They are implemented to a greater or lesser extent in packages such as *Data Desk*, *JMP*, *MANET* [5] and *CASSATT* [11]. All of the features could be extended considerably: zooming should include panning and an orientation view to aid navigation; rescaling may include sorting as well as transforming; and querying should offer far more than a simple default. In the case of tree diagrams in *KLIMT*, there are already two levels of querying of nodes, a terse summary and a more extensive description, but ideally it should be possible to query the model behind the diagram and carry out sensitivity analyses. The next section describes how interactive features have been adapted and implemented in *KLIMT*.

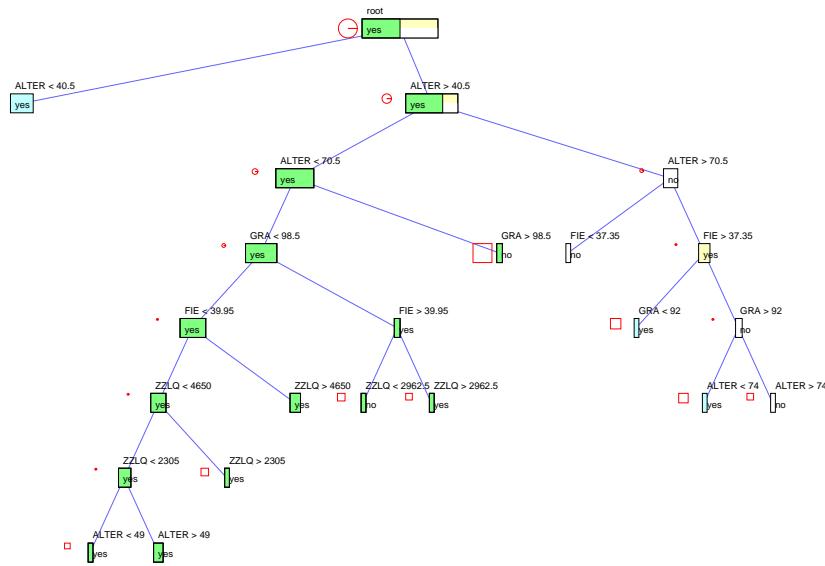


Figure 2: Tree displayed by KLIMT.

2 Features

With the requirements for an interactive software in mind we tried to include all the features in KLIMT that are helpful for an analysis of a tree model. Let's start with some basics. KLIMT arranges the tree so it fits in the working window. You can see a snapshot of the initial state in Fig. 2. Now working in that window you can re-arrange the tree at any time by simply dragging the nodes - both entire subtrees or single nodes can be moved. This is especially helpful if you want to sort nodes in some logical order or increase the readability of the tree. As a computer screen has the opposite orientation of a sheet of paper it is handy for bigger trees that you can rotate the tree by 90 degrees. KLIMT uses an algorithm to place the nodes so that the links won't overlap, but you are still free to rearrange the tree as you please.

Once the tree has the desired shape you can query the nodes for basic statistical information. There are two different query types: basic and extended query. Basic query displays the total number of cases in the selected node as well as percentages and numbers of cases for each class. Extended query adds the splitting criterion, the node classification, the deviance and the deviance gain. If a criterion other than deviance was used for growing the tree then that criterion's value is displayed instead.

KLIMT has an intuitive command for manual pruning. Simply select a de-

sired node, issue the “Prune” command and the entire branch will be pruned. A small plus-sign will remind you that the branch has been pruned. By clicking on the symbol you can expand the branch to its full extent again. You have also the capability of copying a pruned tree or a branch into a new window for further analysis.

While working with the tree you will certainly be interested in the dataset itself. That is why KLIMT offers you a variety of basic plots like histograms, barcharts and scatterplots as can be seen in Fig. 3. By selecting the desired variables from a list you can create any of the available plots instantly. But the real value of the plots is the ability to use linked highlighting in both plots and the tree itself. Selecting cases in a plot means that the same cases will be highlighted in corresponding nodes of the tree and in all other plots. For example by creating a barchart of the classification variable and selecting a bar for a specific class in the plot you can see where the cases of that class are distributed in the tree and of course in other plots. You can choose from various selection methods such as addition, intersection or exclusion, to select exactly the cases of interest.

For visualization of different aspects of the data in nodes there are different views available. You can choose between proportional and fixed view. In the proportional mode the size of a displayed node is proportional to the number of cases in that node. This enables the visual comparison of node sizes at a glance. Once highlighting is used you can directly compare the number of selected cases in each node visually. In fixed mode all nodes have the same size and the highlighting is displayed proportionally, so you can compare the relative proportions of classes in nodes. Besides changing the node-views you can tell KLIMT to display all leaves (terminal nodes) on one level in order to make visual comparison among the leaves easier. Different methods of connecting the nodes are also available so you can use the “classical” view where nodes are connected directly, or the “rods and wires” view which is similar to the tree in Fig. 1.

A tree is grown using a prespecified criterion such as deviance or Gini-Index. In KLIMT you can visualize this information as well. For example if the tree was grown using the deviance then you can enable the criterion display and new symbols will appear near each node. Red circles on inner nodes are proportional to the deviance gain in that node, i.e. $\Delta D = Dp - \sum_i Dc_i$ where Dp = deviance of the node and Dc_i = deviances of children. Red rectangles near leaves are proportional to the residual deviance of the leaf. Looking at these symbols the quality of a split can be estimated at a glance. Problematic splits or nodes with very high remaining deviance can be identified easily.

KLIMT is also equipped with an interface to R software². This has numerous advantages for users and makes KLIMT very easily extensible and customizable. If you are familiar with R then you can use your usual environment for analyzing the data, preparing the dataset or constructing different models. Once you have a tree you can start KLIMT with a single command from within R. Detailed

²Whenever R is stated the entire family of software products R, S or Splus is meant. The interface has been tested thoroughly mainly with R, but is intended to work with all other members of the family.

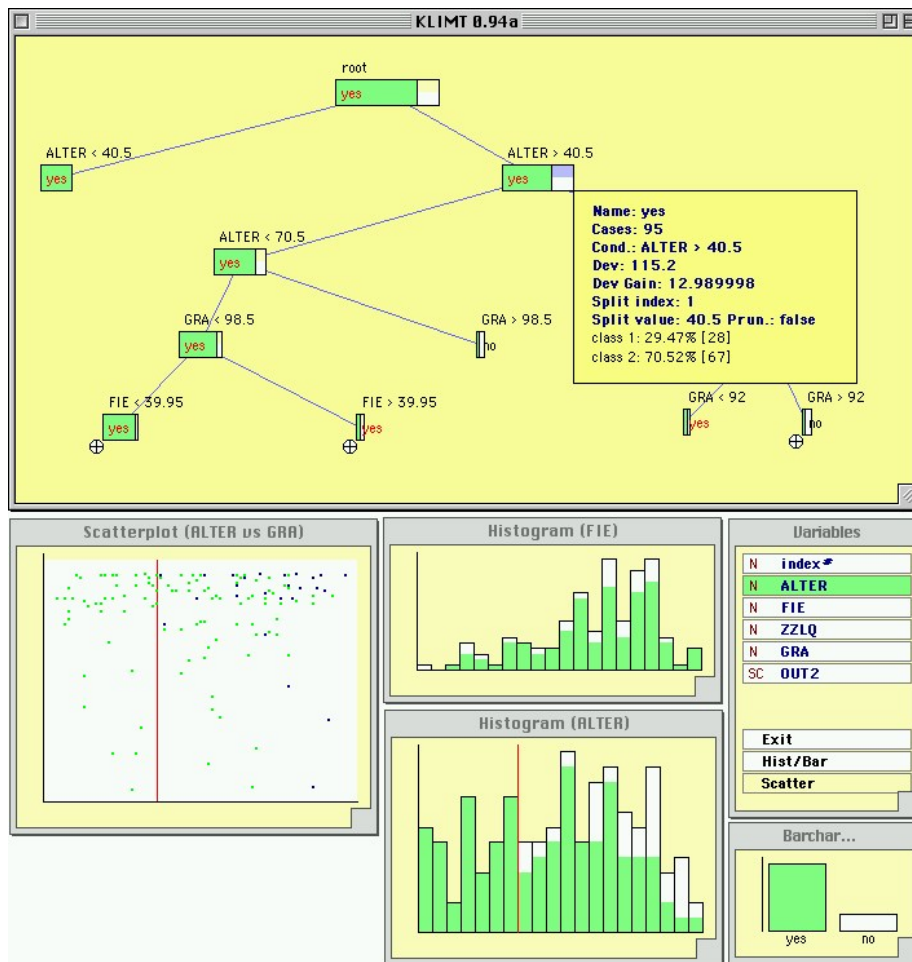


Figure 3: Actual snapshot of KLIMT.

description of further advantages of this interface can be found in the “Future Plans” section.

The newest feature in KLIMT is the ability to generate PostScript or EPS files for easy printing and use in documents or other applications. Any tree or graph can be exported in native PS format. KLIMT does not simply rasterize the screen output, but uses its unique *Portable Graphics Subsystem* to generate fully scalable and flexible output. Figure 2 was created using this method. PostScript is just one of many plug-in export filters that can be used in KLIMT. This feature ensures that KLIMT can also be used for presenting results.

3 Implementation

Provided you have a dataset and want to build and analyse a tree-based model, what do you have to do in order to use KLIMT for your analysis? We decided not to re-invent the wheel and did not hard-code the algorithm for growing trees in our program. Instead KLIMT relies on a tree created by other statistical software³, preferably R. This makes KLIMT very flexible, because if the *tree* library of R is enhanced over time so is KLIMT. If we implemented the growing algorithm directly in the program, we would have to update the software continually. Also many statisticians are familiar with R and use it for their analysis so KLIMT seamlessly extends their capabilities without the need of learning how to grow trees in a different software package - for trees you simply call the KLIMT command instead of the `plot` command.

Another important goal during the design of KLIMT was compatibility. We wanted our software to be compatible with as many computers and operating systems as possible. That's why we decided to write KLIMT in Java. No special libraries or additional classes are necessary. Moreover the use of JDK 1.1 ensures compatibility with older Macintosh operating systems. KLIMT has been successfully tested on Unix, Macintosh and Windows platforms, even different formats of ASCII files on these platforms are handled correctly.

KLIMT is currently work in progress. There are still some features that are waiting to be implemented. The next section will mention just a few of them.

4 Future plans

Our main goal is to enhance interactivity with R even more. As of now the interface to R is one-way only. KLIMT can read structures like trees or datasets from R and can be started directly from R, but there is no way of passing objects from KLIMT back to R. We are working on a direct communication where R and KLIMT share objects so they can both use each other's functions transparently. Once this interface is beyond the current prototype stage there are countless ways of enhancing KLIMT's features by the user. One can use the graphical highlighting methods of KLIMT to select certain cases and create a corresponding dataset in R for further analysis. Changing properties of the tree in KLIMT, like the splitting criterion or the sequence of splits, by intuitive drag-and-drop operations will tell R to immediately re-grow the tree using the new properties and display it in KLIMT along with the old tree. Thanks to linked highlighting both trees would be linked so one can identify changes in the model with interactive methods easily by selecting cases in one tree and seeing their respective position in the other one. Usage of event handlers allows R functions to be called on KLIMT-events: e.g. running a query on a node in KLIMT would call a function in R which calculates basic statistics on the cases

³If used as stand-alone program, KLIMT does expect a file consisting of a dataset (any ASCII form) and a tree. The tree must be in the format used by R. Any statistical software that can produce such a file can be used as a source for KLIMT.

in the node. We could go even as far as `ggobi` did - introducing an API to allow R to control KLIMT, like letting R highlight certain cases or perform more complex animations of the tree and linked plots by loops in R. Our goals are very similar to what `ggobi` achieved as described by Temple Lang [6] - the one-way interface corresponds to the functionality of `xgobi` and the currently described full interface to that of `ggobi`. Use of a similar API makes a connectivity between `ggobi` and KLIMT easily possible. Both projects are based on interactive graphics principles therefore the wish to connect them seamlessly is logical.

Besides R-connectivity our other plan is to introduce interactive, semi-automatic pruning. During the growing of a tree there are various parameters that influence the height of the tree. Let's assume the tree was built using the deviance criterion then the growing method expects there to be lower limits for deviance and node size, so nodes with smaller deviance or less cases than specified won't be split. Interactive altering of these parameters, for example by using a slider, would result in an updated tree with branches pruned according to the specified parameters. By dragging the parameter slider you could watch the tree grow and shrink in response to the changes.

Finally there is a problem with displaying small subsets. If there are 200 cases of one class in a node and only 4 cases of another class then even if you highlight all cases of the second class, you won't be able to see the 4 cases in the node due to the limited resolution of a computer display. The idea of redmarking was introduced in MANET for exactly these situations. Area displays may mislead in three ways because of insufficient screen resolution: a bar in a histogram or barchart may not be drawn at all, because the number of cases is too small; no cases may be highlighted in a bar although a few are selected; all cases may be highlighted although a few are not selected. In all cases a thin red line is drawn under the corresponding bar. We propose to introduce a similar scheme in KLIMT, both for its histograms and barcharts and also for the nodes of the tree. The issue of graphical stability is discussed in general in Hofmann [4].

KLIMT is currently a prototype software. We plan to implement the missing features and some of the features mentioned in this section before first public release which is roughly scheduled for Fall 2001. Any feedback, ideas or suggestions are always welcome.

5 Conclusion

Static tree plots don't contain enough information for thorough model analysis and attempts to add such information don't in general yield the desired effect. To try to solve this dilemma we developed KLIMT. It uses interactive graphics to allow exploratory analysis for trees. Main features include interactive queries, reordering of nodes, statistical plots, linked highlighting for all components, pruning and growing criterion visualization. KLIMT is a stand-alone Java program that operates on any trees generated by other statistical applications, preferably R because of an integrated interface which allows KLIMT to

be used directly from within R. KLIMT is work in progress with many more features to be implemented, check our homepage for further developments - <http://www.klimt-project.com/>

References

- [1] Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and regression trees*, Wadsworth (1984)
- [2] Carr, D. B., Sun R.: *Some graphics for Recursive Partitioning*, Interface (2001)
- [3] Dirschedl, P.: *Klassifikationsbäume - Grundlagen und Neuerungen*. In Fleischer, W., Nagel, M., and Ostermann, R. (Eds.), *Interaktive Datenanalyse with ISP* (pp. 15-30). Essen: Westarp Wissenschaften (1992).
- [4] Hofmann, H.: *Graphical Stability of Data Analysing Software*, In Klar, R., Opitz, O. (Ed.), *Classification and Knowledge Organisation*, (pp. 36-43). Freiburg: Springer. (1997)
- [5] Hofmann, H.: *MANET* www1.math.uni-augsburg.de/Manet/ In Augsburg: Rosuda (2000)
- [6] Temple Lang D., Swayne D. F.: *ggobi meets R: an extensible environment for interactive dynamic data visualization*, Proceedings of the 2nd International Workshop on Distributed Statistical Computing, TU Vienna (2001)
- [7] Unwin, A. R., Hawkins, G., Hofmann, H., and Siegl, B.: *Interactive Graphics for Data Sets with Missing Values - MANET* Journal of Computational and Graphical Statistics, 5(2), 113-122 (1996)
- [8] Unwin, A. R.: *Requirements for interactive graphics software for exploratory data analysis*. Computational Statistics, 14, 7-22 (1999)
- [9] Venables W.N., Ripley B.D.: *Modern applied statistics with S-plus*, Springer (1994)
- [10] Wilkinson, L.: *The grammar of graphics*, Springer (1999)
- [11] Winkler, S.: *CASSATT* www1.math.uni-augsburg.de/Cassatt/ In Augsburg: Rosuda (2000)

List of Figures

1	Classification tree as plotted by R.	2
2	Tree displayed by KLIMT.	3
3	Actual snapshot of KLIMT.	5