

Following Traces of Lost Models

Simon Urbanek
AT&T Labs - Research
urbanek@research.att.com

Abstract

In modern statistics, the variety of models available for classification or regression is huge. Due to advances in computational technology and broadly available statistical software, we are now able to fit many models to the given data. Ensemble methods such as bagging or boosting take advantage of this development to improve prediction accuracy, but individual models are lost in the quantity of ensemble elements. Tools available for the analysis of multiple models or model ensembles are, however, rare. In this paper, we present a visualization method that is part of a larger framework for exploratory model analysis. This visualization tool allows us to compare many tree-based models using Trace Plots with respect to variables used and cut-points chosen. It can be used for both analysis of model properties and finding special subgroups in the data. It is suitable for analyzing model ensembles or comparing individual models. The method can accommodate a large number of tree models. We will illustrate the method in a practical application.

Keywords: Visualization, Tree models, Classification and regression trees

1. Introduction

Tree-based methods provide an appealing class of models, because they require very little assumptions about data distribution, they handle missing data and multiple interactions well. In addition, individual tree models are easily interpretable and allow inclusion of domain knowledge. Finally, ensemble techniques such as bagging or random forests augment the prediction precision of individual tree models to a highly competitive level. Ensemble techniques combine multiple individual models into one large, combined model. Such resulting model has often better predictive results, but the interpretability of its components is lost. In this paper we present a method that allows us to visualize many tree models and shed some light on the models used in tree ensembles, gaining more insight about both the models and the data.

In the first part we want to quickly review main properties of tree models and motivate the use of

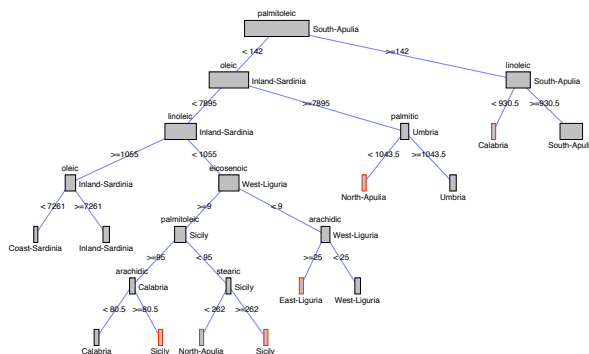


Figure 1: A classification tree.

ensembles. In the second part we introduce the *trace plots* as a way of looking at multiple tree models simultaneously. Then we show a practical examples and conclude in a summary section.

2. Tree-based models

Tree models are usually applied to classification, regression or survival estimation problems. A tree model is a recursive partitioning of the covariates space with separate prediction models for each partition. The most commonly used tree models use splitting rules orthogonal to the covariates' axes and constant model for each partition as introduced in Breiman et al. (1984). Each node of a tree describes a partitioning rule. A sample classification tree is shown in Fig. 1.

Most commonly used tree model construction relies on a greedy algorithm which recursively optimizes one node at a time by considering all possible splits and choosing the one maximizing an optimality criterion. Clearly, this can lead to globally sub-optimal models, because the optimization does not consider any subsequent partitioning. Secondly, in cases where more competing splits are close to the optimum, small changes in the training data can lead to a choice of a different variable or cut-point and hence very different models.

Ensemble methods attempt to tackle the optimality problem by exploiting the model instability and building multiple different models. This can be

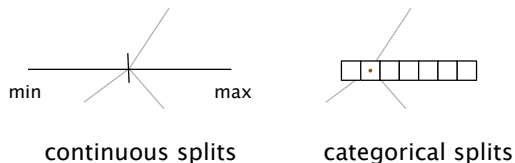


Figure 2: Representation of splits in trace plots.

achieved by changes in the training data (bootstrapping cf. bagging, Breiman 1996) or more random split selection (cf. random forests, Breiman 1999). Since the resulting models are driven by the training set, the variety of the constructed models reflect the properties of the data and provide information about the model fit. Close look at the used models will also reveal whether ensemble methods can achieve better predictive result in that particular case.

The comparison of tree models is far from trivial, due to the complexity of the models. They can differ in their hierarchical structure, variables used in the splitting rules and predictions in terminal nodes. Each of such aspects can be measured by various metrics, such as the tree metric, partition metric or fit metric. Those are, however, based on pairwise comparisons and thus an attempt to quantify a series of several tree models leads to a high-dimensional problem, which is hard to solve. Application of multi-dimensional scaling was attempted, but did not yield results that are satisfactory in that the resulting mapping is usually too distorted.

We propose a method that allows us to visually compare both the hierarchical structure and the splitting rules of arbitrary many tree models.

3. Methodology

The basis of the *trace plot* is a rectangular grid consisting of split variables as columns and node depths as rows. The depth of a node is its distance from the root node, i.e. number of edges in the shortest path from the node to the root of the tree. Each cell in this grid represents a possible tree node.

In order to distinguish actual split points, each cell contains a glyph representing possible split points as illustrated in Fig. 2. For continuous variables it consists of a horizontal axis and a split point is represented by a tick mark. Categorical variables are shown as boxes corresponding to possible split combinations. Each two adjacent inner nodes are connected by an edge between their split points.

A trace plot of the classification tree from Fig. 1 is shown in Fig. 3. The root node features a split



Figure 3: *Trace plot* of the classification tree from Fig. 1.

on the variable *palmitoleic*, which is represented by the rightmost column. Its parent nodes use splits on the variables *linoleic* and *oleic*, hence the two edges leading from the root node to the next row of splits. There are no further inner nodes as children of the *linoleic* split, therefore the branch ends there. Analogously, all inner nodes are drawn in the trace plot until terminal nodes are reached.

It is evident that all splits of the tree can be reconstructed from its representation in the trace plot, because every cut point is shown in the trace plot. Equally, it is possible to reconstruct the hierarchical structure of the tree due to the presence of edges in the trace plot.

Moreover the trace plot removes an ambiguity present in the hierarchical view: the order of the children nodes is irrelevant for the model, whereas swapping left and right children in the hierarchical view produces quite different hierarchical plots. In the trace plot the order of the children is defined by the grid and therefore fixed.

In the following section we will illustrate the use of trace plots for analysis of tree ensembles on a real dataset.

4. Example

Although we illustrate the use of trace plots on a classification example, trace plots can be also used for regression and survival trees analogously. The tree in Fig. 1 is based on a classification task of Italian olive oils dataset (see Forina et al. 1983). Each olive oil is classified according to its region of origin, based on measurements of contained fatty acids. There are 9 distinct regions representing the categories of the response variable. With tree models

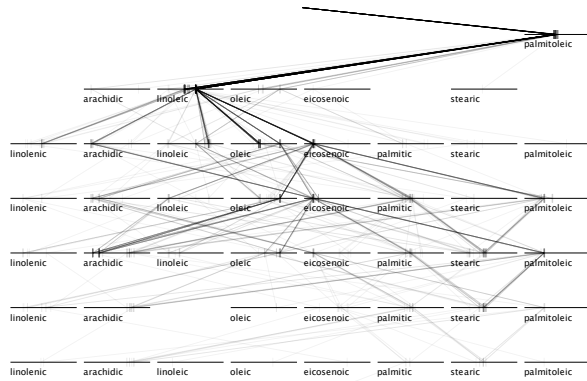


Figure 4: *Trace plot* of 100 bootstrapped trees.

there is an important question that should always be asked: how stable is the model? The danger of instability is inherent to tree models due to local optimization performed at each node. One way of assessing the extent of instability is to perform bootstrapping.

One important advantage of trace plots is the ability to display multiple tree models simultaneously, superimposing all on the same grid. A trace plot of 100 bootstrapped classification trees for the olive oils data is shown in Fig. 4.

In order to prevent overplotting, we use semi-transparent edges. Consequently, often used paths are more opaque than seldom used paths. We can clearly see that the first split almost always uses the *linoleic* variable. In the next step, however, there are several alternatives for the splits. Interestingly, both variables *oleic* and *eicosenoic* seem to be important, because the ‘missed’ split appears in the next level as the ‘X’-shaped pattern shows.

Some patterns seem to be repeated further down the tree, indicating a rather stable subgroup that can be reached by several different ways along the tree. In this particular example we can recognize substructures that affirm the partial stability of the tree models.

The instability in this particular example is in most cases given by the sequence in which the subgroups are separated. This is partially due to the fact that we are dealing with a multi-class problem, thus the reduction of impurity can be achieved by splitting off an arbitrary class or a group of classes. Nevertheless, our tree specimen from Fig. 1 is a rather rare one as we see in the trace plot in Fig. 4, because its trace does not match with the main, opaque paths.

5. Conclusion

Trace plots offer a convenient way to display many tree models at once. They are based on a grid of covariates and node depths and splits with parent/child relationship connected by edges. It is possible to see split point locations and covariates in splits at a glance. This technique can be applied in the framework of exploratory model analysis to learn more about the behavior of tree-based models for a given dataset. The applications range from stability analysis to detection of interesting subgroups and assessment of covariate importance.

Trace plots still offer a variety of future enhancements. Permutation of grid columns leads to different views. Although manual re-ordering provides a good start, automated methods such as optimization of edge crossings can be of value. Color-coded edges can be used to represent majority class in the data passed down the path of a classification tree or predicted value of a path in a regression tree.

The value of trace plots can be even more enhanced by adding interactive features such as query and highlighting, facilitating further drill-down to both data and model groups. The proposed methods have been integrated in the interactive software for visualization and analysis of tree models - KLIMIT (see Urbanek 2002). Conventional tree visualization techniques are well complemented with trace plots as a tool for analysis of forests, global overview and navigation.

References

- L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123/140, 1996.
- L. Breiman. Random forests - random features. Technical Report TR567, University of California Berkeley, Statistics Department, 1999.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- M. Forina, C. Armanino, S. Lanteri, and E. Tiscornia. Classification of olive oils from their fatty acid composition. In H. Martens and H. Russwurm, editors, *Food Research and Data Analysis*, pages 189–214. Applied Science Publishers, London, 1983.
- Simon Urbanek. Different ways to see a tree - KLIMIT. In *Proceedings in Computational Statistics, Compstat 2002*, pages 303–308. Physica, Heidelberg, 2002.