TEACHING ABOUT DECISION TREES FOR CLASSIFICATION PROBLEMS

Joachim Engel[1], Tim Erickson[2] and Laura Martignon[1]
[1]Ludwigsburg University of Education, Ludwigsburg, Germany
[2]Epistemological Engineering, Oakland, CA, USA
engel@ph-ludwigsburg.de

In times of big data tree-based algorithms are an important method of machine learning which supports decision making, e.g., in medicine, finance, public policy and many more. Trees are a versatile method to represent decision processes that mirror human decision-making more closely than sophisticated traditional statistical methods like multivariate regression or neural networks. We introduce and illustrate the tool ARBOR, a digital learning tool which is a plug-in to the freely available data science education software CODAP. It is designed to critically appreciate and explore the steps of automatically generated decision trees.

INTRODUCTION

Today's technology lets us record and collect high-dimensional data for very large sample sizes over extended periods of time. The tremendous data volume, e.g., in medicine, environment, finance or social studies, offers unprecedented opportunities to examine, study and predict subsequent trends and patterns. Yet, in these tsunamis of multivariate data, it is increasingly difficult to discern structure in the data by eye. Traditional statistical techniques often fail to take full advantage of the opportunities in complex data as they are too rigid to recover non-linearities and do not facilitate the easy exploration of interactions in datasets with many confounded and correlated variables. Tree-based methods can help us predict and discover structure in observational data of this kind: data with many inter-related covariates, mixed-type variables, non-linear relationships, confounders, co-linearity, etc. They work particularly well when it is possible to divide the data into relatively homogeneous groups according to some outcome variable.

When the target variable is metric (numerical), we call the tree a *regression* tree, and can use it to predict, for example, the mean or median value in the group. In this paper we focus on categorical outcome variables: our trees will be *classification* trees. Each branching in our tree represents a question we ask of the data, and the terminal nodes or leaves of the resulting trees can be interpreted as decisions.

Decision trees created algorithmically from training data are simple and yet powerful tools capable of achieving high accuracy in many tasks while being highly interpretable. The "knowledge" learned by a decision tree appears as a hierarchical structure, a blueprint for decisions. This structure holds and displays the knowledge in such a way that even non-experts can immediately apply it.

In ProCivicStat (ProCivicStat Partners 2018, see https://iase-web.org/islp/pcs) we developed teaching materials for understanding statistics about society involving multivariate authentic datasets on topics like migration, income disparity, sustainable development and much more. Data about social phenomena such as life expectancy, child mortality, access to clean water and number of physicians per 1000 inhabitants do not stand in isolation. Their description and understanding involves a network of variables that are *correlated*, *interact*, or have *non-linear* relationships with each other. Understanding these phenomena and their data is a fundamental prerequisite for sound and sustainable policy decisions. Therefore, tree-based methods are an appropriate approach to explore these types of data. As part of ProCivicStat, the digital tool ARBOR had been developed by the second author to teach about tree-based data exploration (Engel, Erickson & Martignon 2018). Cathy O'Neill (2016) pointed out that algorithms are never neutral but reflect the goals and ideology of those who create them. ARBOR is a tool to look behind automated decision making, thus contributing to a critical appreciation of tree-based algorithms.

AN INTRODUCTORY EXAMPLE

We start with a simple example for a decision tree: Should I attend a certain upcoming conference or not? The decision will (amongst other criteria like available time and funds) depend on my previous experience with conferences, captured by the following variables:

- Community `(COM)`: Will I connect with colleagues?
- Inspiration `(INS)`: Will the conference give me new inspirations for my work?

- Own results (OWN): Do I have new ideas and results to present?

A tree, based on experience with previous conferences, for deciding to go or not to go to the next conference might look like the following (Figure 1):

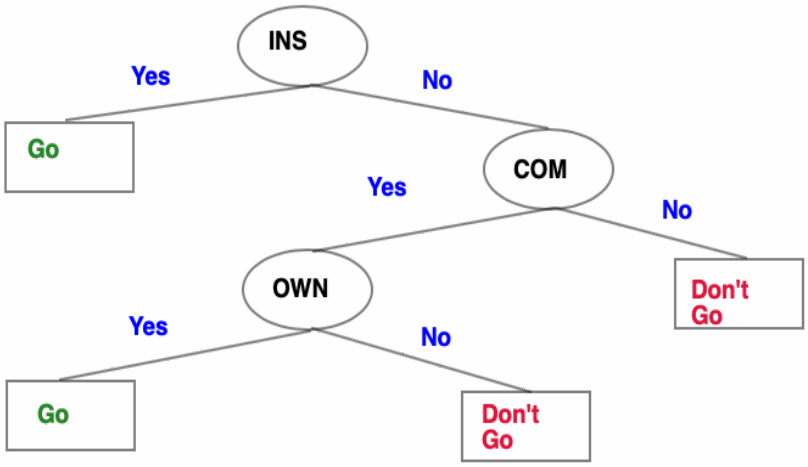

Figure 1: A simple tree representing decisions to attend a conference

Getting inspiration is the most important criterion for deciding to attend the next conference. I will certainly attend the conference if it will give me new ideas, but I will also go—even if it doesn't present me anything new—provided that I'll connect with colleagues and have something to present of my own. My experiences from past conferences ("the training sample") led me to the rules represented by the above tree. In this simple example, all three predictor variables `INS`, `COM` and `OWN` were categorical, in fact, binary. The same is true for the target variable (`Go` or `Don't Go` to the conference). In applications, we usually have many more predictors and they are a mixture of categorical and numeric variables.

TREES AS ROBUST DECISION TOOLS

The goal is to create a model (represented by a tree) that predicts the value of the target variable by "learning" simple decision rules inferred from features of the training sample. This is achieved by specifying regions of the covariate space such that the outcome is sufficiently homogeneous and the number of observations in each region is sufficiently large while the regions themselves are sufficiently numerous and unstructured to allow for complex relationships between covariates and the outcome.

We can think of binary trees as partitioning the sample space recursively into more and more regions that are increasingly homogeneous with respect to the target variable. The nodes of the tree correspond to partition sets. If the set is divided into two parts in the recursive process of partitioning, then the corresponding "branching" node has two "child" nodes. The terminal nodes of a tree correspond to final partition sets and are called *leaves*. Tree-based methods are members of a growing class of methods from the machine learning literature designed to yield a balanced solution to the dilemma of allowing flexible functional forms while avoiding overfitting. The following example, with data from Chambers & Hastie (1992) illustrates the equivalence between recursive partition schemes and binary trees.

EXAMPLE: KYPHOSIS AFTER SPINAL SURGERY

We explore the presence of a postoperative kyphosis after some corrective spinal surgery on 81 children. For simplicity and ease of a two-dimensional graphical representation of the sample space we limit ourselves to two predictor variables only:

- `Age`: age of the child in months

- `Start`: the number of the first (topmost) vertebra operated on

The target variable is

- `Kyphosis`: a factor with levels `absent, present` indicating the presence or absence of a postoperative kyphosis (a type of deformation).

The aim is to derive a tree (or an equivalent recursive partitioning scheme) on the basis of the training sample of 81 cases. Our tree will predict the occurrence of a kyphosis for new children on the basis of their age and the topmost vertebra to be operated on. Running the R package `tree` on the training data results in the following tree and its corresponding recursive partitioning scheme (see Fig. 2). Note that this tree has a misclassification of 11 / 81 = 0.1358, i.e., 11 children in the training set are misclassified. As different types of misclassification have different consequences, we note that 3/59 were wrongfully classified "absent" (false positive) while 8/22 were wrongfully classified "present" (false positive). We could lower both rates by using a larger tree with more splits and leaves. But, an overfitted tree would fare worse in predicting the outcome for new data.

Figure 2: Partition for the sample space (restricted to two metric variables) and associated tree for the Kyphosis data. In the tree, "yes" or "true" answers go to the left.

The simple tree in Figure 2 sheds light on the structure of the data: if the topmost vertebra to be operated on is below the 12th vertebra (`Start` > 12.5), a resulting kyphosis is rather unlikely. If a vertebra further upwards had been operated on, the influence of age is nonlinear: for children between the age of 3 years (35 months) to 11.5 years (139 months) the surgery may result in a kyphosis (14 out of 22 children in the training sample), not so for younger or older children.

To grow a tree based on training data, you must make three important choices:

1. How and where to split a partition set
2. When to stop the tree-growing procedure
3. How to assign an estimate to every terminal node

As humans, we might make the choices intuitively. An algorithm, however, needs specific direction.

1. For an algorithm to choose a good split, it needs a "purity measure." Then the goodness of a split is evaluated by how much the overall purity in the child nodes is improved over the parent node. An obvious choice is the misclassification rate, or a weighted score for different types of misclassification taking into consideration that different types of misclassification have different consequences. Other purity measures recommended in the literature are cross-entropy or the Gini index (Hastie et al., 2008).
2. An algorithm needs a stopping rule. If it were to allow the tree to become too big, the resulting tree would not perform well on new data because of overfitting. This is similar to variable selection in

multivariate regression. Just like possible purity measures, stopping criteria come in many different forms, including:

- A maximum number of nodes in the tree: Once this maximum is reached, the process is halted.
- A minimum number of observations for each node: partitioning of nodes stops if the number of observations goes below a certain number.
- A threshold for the purity reduction: if the purity improvement with further splitting would be smaller than the threshold, then stop.
- Powerful algorithms such as `CART` (Breiman et al., 1984) or the `tree` package in `R` follow a more sophisticated approach: After growing an oversize tree, they prune the tree based on cross-validation.

3. One straightforward and obvious solution to the third issue is to classify a terminal node by a local majority vote: what's the most common outcome for data from the training set?

ARBOR: A TOOL FOR LEARNING ABOUT DECISION TREES

Complex algorithms such as CART, or simple algorithms such as "fast and frugal" (Martignon et al., 2003) can be powerful and useful, but we believe that using them masks potential problems: first, the astute reader will recognize that the algorithm in the previous example made choices that frankly may not be justified. For example, most of us might rather wait for more data before stating definitively that being older than 139.5 months means you probably won't suffer from kyphosis. Put more broadly, people deploying these tools may not understand what assumptions, criteria and limitations underlie them, what the training set is for, or even what the nodes of a tree really mean. That is, there is something to learn about decision trees before one can responsibly use an algorithm to create them.

To that end, a hands-on activity with physical "data cards" is a good starting point before turning to technology. Learners receive a deck of cards with information about football (soccer) and handball players. Each card has physical characteristics like height, weight, age etc. (the predictors) and the sport they play (the target variable). They separate the deck into a training sample and a test sample, then use the training sample to develop rules for classifying the cards, predicting the sport they play using only the physical characteristics. Then they can ask: How can we render these rules as a series of yes/no questions? How can we represent them as a tree? How well do these rules work when applied to the test sample? What are the most important predictors? How should we measure the goodness of a rule?

The goal of this exercise is not to derive optimal rules and the best tree possible. Instead we aim to get a feel for how to evaluate different trees, to invent and discuss different measures of homogeneity or "node purity" and also to learn about overfitting, i.e. to see that a large tree is not necessarily a good tree, because it picks up too much noise. Although cards are a good place to start, when the data and associated classification rules become more complex, it becomes impractical to test and evaluate them "by hand."

The second author developed ARBOR, a digital learning tool, as a plug-in to the freely available software CODAP (Finzer, 2017). It is designed to familiarize learners with tree methods. Like the activity with the cards, and unlike the implementation of trees in software for professional data analysis, ARBOR has no automated algorithms that compute optimal splits and right-sized trees. The user, through drag and drop moves, decides step by step which variables to use for consecutive splits, how to specify the split, and when to terminate tree growth. Misclassification rates evaluating the goodness of the chosen split are immediately reported, which allows comparison with alternative splits. The purpose of ARBOR is not the derivation of an optimal tree, but to let the user explore the flexibility of the tree method and the consequences of various splits, thus to gain appreciation for trees as an automatized method of learning from data.

Figure 3 shows the successive steps in constructing a decision tree for 48 players with ARBOR on the basis of weight, height, age and position. Predictors for splitting nodes are chosen using drag-and-drop gestures. Users can adjust cut points manually. Misclassifications (see Fig. 3, right panel: TP=true positive, TN=true negatives, FP=false positives, FN=false negatives) are recorded in real time. The tree in Figure 3 classifies a player as soccer if his height is below 186 cm or if the weight is below 93 kg and his position is goalkeeper or defender. Otherwise he is classified for handball. Notice that this rule misclassifies 4 players out of the training sample of 48 (one handballer falsely classified as soccer, three soccer players falsely as handballer. We encourage the reader to play with these data by clicking on https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=100528

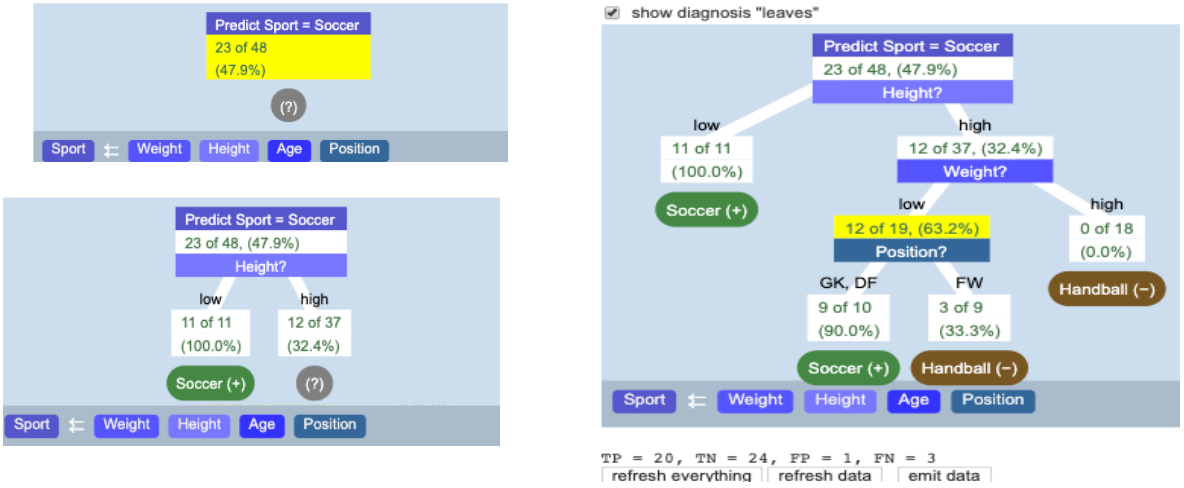

Figure 3: Successive steps in construction a decision tree with ARBOR, to be read from upper left (root node, indicating that 23 out of 48 are soccer players to the final tree with terminal nodes

The capacities of CODAP for data visualization are accessible to help the user make good choices. For example, Figure 4 shows that a split using "`Height` < 186 cm" leads to a pure node of 11 soccer players.

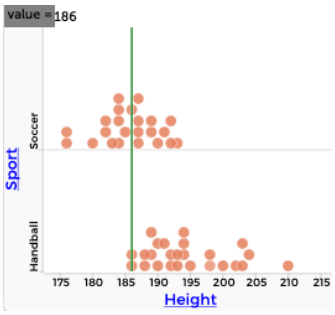

Figure 4: CODAP graph showing that all players with a `Height` under 186 cm are soccer players.

XENOBIOLOGIST: A DATA GAME FOR DIAGNOSING EXTRATERRESTRIALS

Activities of the previous section are aimed at creating a feel for how trees are being constructed, how to measure node purity and how to evaluate the performance of the whole tree. However, by focusing on the training sample only, one may be tempted to grow a very large tree that—with enough splits—may even result in zero misclassifications for the training sample. Oversized trees usually perform poorly with classifying new data. Overfitting is a common problem in all model fits and occurs when noise in the data is taken for structure. For testing the quality of a decision tree, the training sample is biased and we need new data (the "test sample"). The second author designed a game where new data are created through a random generator.

> **Scenario**: You're a xenobiologist studying alien creatures. Sometimes they get sick. At the beginning of our tale, you have records for 10 creatures, some of whom were suffering from *ague* (an alien disease). As specialist for extraterrestrials, find out which of these poor aliens are suffering and design a diagnosis plan.

On starting the game, the data table presents information about the true health status and various somatic variables of 10 aliens. The task is to create a decision rule that predicts the health status based on six predictors: hair color, eye color, number of tentacles and antennae, height and weight. In the next step, new aliens show up for which you know their color of hair and eyes, etc. Are they healthy, or do these creatures suffer from ague? At first, you decide on individual cases "by hand" (see Fig. 6). For correct decisions, the game score will rise; otherwise you'll lose points. Finally, you can make automatic decisions by creating a tree (Fig. 5), which the system uses as an algorithm.

creatures (10 cases)

index	health	hair	eyes	antennae	tentacles	height	weight	source
1	sick	pink	orange	6	6	112.7	213.9	training
2	well	blue	purple	7	10	127.8	279.6	training
3	sick	pink	purple	6	10	119.4	229.2	training
4	sick	pink	purple	6	8	128	248.8	training
5	well	blue	orange	8	8	115.1	171.9	training
6	well	blue	purple	6	8	108.7	163.2	training
7	well	blue	orange	8	6	135.7	294.7	training
8	well	blue	orange	7	10	143.8	298.2	training
9	sick	pink	orange	8	8	124	238.4	training
10	sick	pink	orange	6	6	112	196.2	training

Figure 5: Data table with health status for 10 alien creatures (left) and a decision tree (later in the game, right) with data for 46 creatures.

This first disease—ague—is simple, involving only one attribute and resulting in a one-split tree that perfectly predicts all outcomes. The game has a suite of increasingly complex maladies. This game is freely available under https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=33583

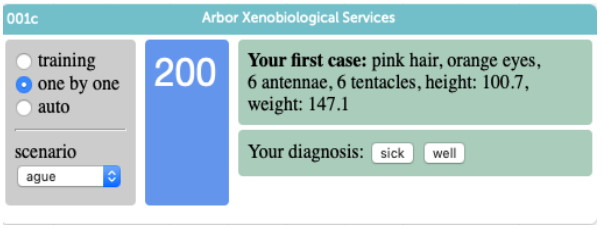

Figure 6: New aliens are arriving. With info about their physical status, you must decide if they are sick or well.

SUMMARY

Trees are a modern tool for classification and nonparametric regression, suitable for large mixed-type multivariate data. The digital learning tool ARBOR is designed to familiarize learners with this powerful and versatile method for data exploration and prediction. Instead of relying on algorithms, ARBOR requires the user to make successive choices about which variable to split, how to split, and when to stop growing the tree. By using ARBOR, students can come to understand what the algorithms accomplish, and perhaps even more to the point, come to understand the nature of trees themselves. We implemented the teaching design described here several times in seminars for second year students preparing to be mathematics teachers. A very similar approach has been successfully used with high school students in their last year (Thomas Wassong and Yannik Fleischer, personal communication). ARBOR is a free plug-in to CODAP, the free data analysis platform (Finzer, 2017).

REFERENCES

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.I. (1984). *Classification and regression trees*. Belmont, California: Wadsworth.

Chambers, J.M., and Hastie, T. J. (Eds., 1992) *Statistical Models in S*. Pacific Grove, CA: Wadsworth and Brooks/Cole.

Engel, J., Erickson, T. & Martignon, L. (2018). Teaching and learning about tree-based methods for exploratory data analysis. In: A. Sorto (Ed.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics*, Kyoto, Arizona, USA. Voorburg, The Netherlands: International Statistical Institute.

Finzer, W. (2017). Common Online Data Analysis Platform (CODAP). https://codap.concord.org

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning. Data Mining, Inference and Prediction* (2. Ed.). New York: Springer.

Martignon, L., Vitouch, O., Takezawa, M., & Forster, M. (2003). Naïve and yet enlightened: from natural frequencies to fast and frugal decision trees. In: D. Hardman & L. Macchi (Eds.), *Psychological Perspectives on Reasoning, Judgment and Decision Making*. John Wiley & Sons.

O'Neil, C. (2016). *Weapons of Math Destruction.* London: Penguin Books.

ProCivicStat Partners (2018). *Engaging Civic Statistics: A Call for Action and Recommendations*. A product of the ProCivicStat Project. Retrieved April 22, 2019 from: http://iase-web.org/islp/pcs/