FFTrees: An R package to create and visualise Fast and Frugal decision Trees

Nathaniel Phillips¹, Hansjörg Neth², Wolfgang Gaismaier¹
¹University of Basel, ²University of Konstanz

Introduction and Goals

Complex real-world decisions under uncertainty call for rapid and robust classification strategies. Fast and Frugal Trees (FFTs) are a quintessential family of simple heuristics designed to make transparent, robust binary classification decisions under uncertainty [2, 3, 4]. FFTs are remarkably effective and efficient, and easy to implement compared to more complex algorithms such as logistic regression (LR) and CART. FFTs have been successfully implemented in a variety of applied domains, including medical, legal, and financial decision-making [1, 2, 4, 5].

The purpose of this research was to create an R package called FFTrees that allows anyone to create their own FFTs with the following criteria:

- **Easy to use**: Build effective trees with as little as 2 lines of code.
- **Customisable**: Easily customise trees to match user criteria such as the maximum number of cues or the desired balance of hits (HR) vs. false-alarm rates (FAR).
- **Transparent**: Visualise trees and their performance using simple displays that anyone can understand and easily implement.

Open Source: Source code is freely available and well documented on GitHub with tutorials and example datasets.

Installing and using FFFTrees

You can install FFFTrees (latest version v1.1.0) from CRAN. To open the main package vignette, run:

```
install.packages("FFTrees", lib = "library\(\"FFTrees\(\)\))
```

```
# Load the package
library("FFTrees")
```

Example: Predicting heart disease

Let's create FFTs to predict heart disease from a real-world dataset [https://archive.ics.uci.edu/ml/datasets/Heart+Disease] called *heart-statlog*. The data contains 270 observations of 14 different heart-disease risk factors such as rest blood pressure, serum cholesterol, and smoking habits. The target variable is the presence of heart disease (0 for not having heart disease, 1 for having heart disease).

```
predictors = c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal")
```

```
# Create a formula object for heart disease
heart_disease = heart_statlog$target
```

Before we create the tree, we'll randomly split the original *heart-statlog* dataset into a 80% training set and a 20% test set.

```
# Create a random split of training and test data
training_data <- sample.split(heart_statlog, splitratio = 0.8)
```

To create the tree, we will use the function `fftree` with three arguments: `formula`, a formula specifying a binary function as a set of potential predictors to be considered in the tree (the formula specifies the argument will consider all predictors for use in the tree), `data`, a dataframe of training data, and `target`, an optional dataframe of test data.

```
# Create a fast and frugal tree for heart disease
heart_tree <- fftree(formula = heart_disease ~ ., data = training_data$train)
```

The tree, including training and test statistics, are now stored in the `heart_tree` object. Every `fftree` object has multiple trees which inherently balance hit rates and false-alarm rates.

```
# Print the ffftree object
print(heart_tree)
```

```
# Predict using the ffftree object
predictions <- predict(heart_tree, newdata = heart_statlog)
```

```
# Performance (Prediction)
```

Additional arguments to FFFTrees

```
# Additional arguments to ffftree()

```

```
# Additional arguments to ffftree()

```

Contact and FFFTrees Source Code

```
nathaniel.phillips@unibas.ch
```

```
www.github.com/hphilipp/FFTrees
```

References