Package for ExtraTrees method for classification and regression

Jaak Simm and Ildefons Magrans de Abril

2013-10-05

Contents

1 Introduction

This document provides detailed guidance on using the package `extraTrees`.

2 Training and predicting

Usage of `extraTrees` was made similar to `randomForest` package as Extra-Trees (extremely randomized trees) method is similar RandomForest. The main difference is that when at each node RandomForest chooses the best cutting threshold for the feature, ExtraTrees instead chooses the cut (uniformly) randomly. Similarly to RandomForest the feature with the biggest gain (or best score) is chosen after the cutting threshold has been fixed.

This package includes an extension to ExtraTrees that we found useful in some experiments: instead of a single random cut we choose **several** random cuts for each feature. This reduces the probability of making very poor cuts but still maintains the stochastic cutting approach of ExtraTrees. Using more than one cut (e.g., 3-5 cuts) can improve the accuracy, usually when the standard ExtraTrees performs worse than RandomTrees.

A simple usage example is given in Figure 1. Try changing the value of numRandomCuts to 5 and see how the performance changes. For some data also the value of mtry (the number of chosen features at each node) should be increased.

```
library(extraTrees)
## train and test data:
n <- 1000
p <- 10
f <- function(x) {
  (x[,1]>0.5) + 0.8*(x[,2]>0.6) + 0.5*(x[,3]>0.4) + 0.2*x[,5] +
  0.1*runif(nrow(x))
}
x <- matrix(runif(n*p), n, p)
y <- as.numeric(f(x))
xtest <- matrix(runif(n*p), n, p)
ytest <- f(xtest)

## extraTrees:
et   <- extraTrees(x, y, numRandomCuts=1)
yhat <- predict(et, xtest)
yerr <- mean( (ytest-yhat)^2 )
print( sprintf("Squared error: %f", yerr) )
```

Figure 1: Example of using `extraTrees` with 1 cut (the default).

`METHODS` There two main methods:

- `extraTrees` that does the training,
- `predict` that does the prediction after the trees have been trained.

For classification ExtraTrees at each node chooses the cut based on minimizing the Gini impurity index and for regression the variance.

3 Large scale usage

Although ExtraTrees is quite fast (about 10 faster than randomForest on the same data and number of trees), there are cases when the data set is still too big for the default setup.

3.1 Increasing allocated memory

If your data has high number of data points and/or high number of dimensions, then you can run out of Java memory. This causes following error:

```
java.lang.OutOfMemoryError: Java heap space
```

To solve that you need to increase the memory by supplying "-Xmx1g" for 1GB or "-Xmx4g" for 4GB in R's Java options before loading `extraTrees`:

```
## To solve the problem give more memory to Java.
## Using 1GB Java memory for extraTrees (specified by 1g):
options( java.parameters = "-Xmx1g" )
library(extraTrees)
## train and test data:
n <- 1000
p <- 10
f <- function(x) {
  (x[,1]>0.5) + 0.8*(x[,2]>0.6) + 0.5*(x[,3]>0.4) + 0.2*x[,5] +
  0.1*runif(nrow(x))
}
x <- matrix(runif(n*p), n, p)
y <- as.numeric(f(x))
xtest <- matrix(runif(n*p), n, p)
ytest <- f(xtest)

## extraTrees with 1 CPU thread (the default):
system.time({et <- extraTrees(x, y, numThreads=1)})
## extraTrees with 2 CPU thread:
system.time({et <- extraTrees(x, y, numThreads=2)})
```

Figure 2: Example of how to use `extraTrees` in large scale settings.

```
  options( java.parameters = "-Xmx1g" )
  library(extraTrees)
```

This is shown in the first lines of Figure 2. Make sure your machine has enough free memory available before you do that.

3.2 Using multiple cores

Secondly, if the running time is too long you can use multi-core computation by increasing the `numThreads` option (default is 1) in `extraTrees`. This is shown the last lines in Figure 2.