Panel Data - Regression trees with Random Effects for Panel Data

Jayashree Raman

9/18/2018

```
library(REEMtree)
```

```
## Warning: package 'REEMtree' was built under R version 3.5.1
```

```
## Loading required package: nlme
```

```
## Loading required package: rpart
```

```
library(prediction)
```

```
## Warning: package 'prediction' was built under R version 3.5.1
```

```
library(sqldf)
```

```
## Warning: package 'sqldf' was built under R version 3.5.1
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.1
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.1
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:nlme':
##
##     collapse
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.5.1
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Load data, scaling of variables

```
SessData <- na.omit(read.csv(file="usersessions-avgbase.csv", header=TRUE, row.names = NULL, sep="|")[,
SessData$session_dt <- as.Date(SessData$session_dt,format= "%Y-%m-%d")
SessData$session_num <- as.numeric(SessData$session_start)
SessData$id <- as.numeric(str_replace_all(SessData$userid, 'user_', ''))
SessData$session_num_scale <- scale(SessData$session_num)
SessData$previous_duration_scale <- scale(SessData$previous_duration)
SessData$absence_time_scale <- scale(SessData$absence_time)
SessData$session_length_mvavg_scale <- scale(SessData$session_length_mvavg)
SessData$session_length_mvavg_time_scale <- scale(SessData$session_length_mvavg_time)
SessData$session_length_mvavg_30_scale <- scale(SessData$session_length_mvavg_30)

SessData$session_length_log <- log(SessData$session_length)
SessData$session_num_log <- log(SessData$session_num)
SessData$avg_base_log <- log(SessData$avg_base)
SessData$previous_duration_log <- log(SessData$previous_duration)
SessData$absence_time_log <- log(SessData$absence_time)
SessData$session_length_mvavg_log <- log(SessData$session_length_mvavg)
SessData$session_length_mvavg_time_log <- log(SessData$session_length_mvavg_time)
SessData$session_length_mvavg_30_log <- scale(SessData$session_length_mvavg_30)
```

Looping for clustering - 0 cluster is for the full data

```
df_train <- subset(SessData, session_dt < "2009-04-01")
df_test <- subset(SessData, session_dt >= "2009-04-01")

mae_baseline <-  mean(abs(df_test$session_length-df_test$avg_base))
mae_baseline_log <-  mean(abs(df_test$session_length_log-df_test$avg_base_log))

mae_pred <- 'NA'
rmse_pred <- 'NA'

mdl_reemtree <-REEMtree(session_length~session_length_mvavg_time+session_length_mvavg_30+previous_durati

mdl_reemtree_scale <-REEMtree(session_length~session_length_mvavg_time_scale+session_length_mvavg_30_sca

mdl_reemtree_log <-REEMtree(session_length_log~session_length_mvavg_time_log+session_length_mvavg_30_log

predval <- predict.REEMtree(mdl_reemtree, df_test, id = df_test$userid, EstimateRandomEffects = TRUE)

predval_scale <- predict.REEMtree(mdl_reemtree_scale, df_test, id = df_test$userid, EstimateRandomEffect

predval_log <- predict.REEMtree(mdl_reemtree_log, df_test, id = df_test$userid, EstimateRandomEffects =

mae_pred <- mean(abs(df_test$session_length-predval))
rmse_pred<-sqrt(mean(abs(df_test$session_length-predval)^2))

mae_pred_scale <- mean(abs(df_test$session_length-predval_scale))
rmse_pred_scale <-sqrt(mean(abs(df_test$session_length-predval_scale)^2))

mae_pred_log <- mean(abs(df_test$session_length_log-predval_log))
rmse_pred_log <-sqrt(mean(abs(df_test$session_length_log-predval_log)^2))

cat(' MAE ', mae_pred/mae_baseline)
```

```
##  MAE  0.8204726
```

```
cat(' Scale MAE ', mae_pred_scale/mae_baseline)
```

```
##  Scale MAE  0.8162514
```

```
cat(' Log MAE ', mae_pred_log/mae_baseline_log)
```

```
##  Log MAE  0.8517013
```