

# TSIL 2021: Born-again tree approach for predicting treatment outcomes

Marjolein Fokkema

Load libraries:

```
library("foreign")
library("glmertree")
library("partykit")
library("dbarts")
library("rstanarm")
library("bayestestR")
library("ggplot2")
```

## Method

Experimental approach:

- Perform 10 repeats of 10-fold CV (on observation level).
- Fit default GLM and GLMM trees and evaluate accuracy.
- Fit BART and multilevel BART and evaluate accuracy.
- Generate one outcome for each treatment, for each observation.
- Fit born-again GLM and GLMMM trees to those outcomes. To mitigate effect of increased sample size, assign each row a weight of 0.5.
- In computing predictions from multilevel models, include the random effect (i.e., conditional, not marginal predictions).

Disclaimers and drawbacks:

- Function `bart_vi` allows for including only a single grouping factor. Thus, we have to fit a two-level model. I chose a model with participants nested within studies. GLMM trees allow for fitting three-level models (timepoints nested within participants, nested within studies). But surrogate approach requires use of BART.
- We can reduce the width of credible and confidence interval by increasing the number of repeats of 10-fold CV. If we do 100 repeats, we get ‘significant’ differences between the approaches. If we do 10 repeats, we do not get ‘significant’ differences.

## Experiment 1: IPDMA ADM with and without Short-Term Psychodynamic Psychotherapy (STPP)

Load data:

```
## Load data:
IPDMA <- read.spss("3. STPP+ADM vs (BSP+)ADM combined.sav", to.data.frame = TRUE)
levels(IPDMA$Condition) <- c("ADM", "ADM+STPP")
sapply(IPDMA, class)

##          Study      PatientID      Condition      BSP      Gender
##      "factor"    "numeric"    "factor"      "factor"    "factor"
##          Age      MarStat      Education      EducationB      JobStat
##      "numeric"    "factor"    "factor"      "factor"    "factor"
##      Religion      Epdur      PriorTx      PriorEp      HisHos
##      "factor"    "factor"    "factor"      "factor"    "factor"
## PDcomorbidity ADcomorbidity ADcomorbidityB ADcomorbidityC      CGIS
##      "factor"    "factor"    "factor"      "factor"    "numeric"
##          GAF      Z anx      HAMD17      rawHAMDpre      zHAMDpre
##      "numeric"    "numeric"    "factor"      "numeric"    "numeric"
##      rawHAMDpost      zHAMDpost      rawHAMDfu      zHAMDfu
##      "numeric"    "numeric"    "numeric"      "numeric"

IPDMA <- IPDMA[!(is.na(IPDMA$rawHAMDpost)|is.na(IPDMA$JobStat)), ] # completers only
nrow(IPDMA) # total number of participants
```

```
## [1] 376
```

```
table(IPDMA$Study) # no of participants in each study
```

```
##
## de Jonghe, 2001      Lopez, 2004      Maina, 2010      Burnand, 2002      Maina, 2007
##           86           20           50           74           32
## Martini, 2011      Vitriol, 2009
##           35           79
```

Run analyses:

```
nfolds <- 10L
nreps <- 10L
tree_size <- MSE <- data.frame(gt = rep(NA, times = nreps*nfolds))
MSE$bart <- MSE$bart_m <- MSE$surr <- MSE$surr_m <- MSE$gmt <- MSE$gt
tree_size$surr <- tree_size$gmt <- tree_size$surr_m <- tree_size$gt
set.seed(42)
for (k in 1:nreps) {
  fold_ids <- sample(rep(1:10, times = ceiling(nrow(IPDMA)/nfolds)),
                    size = nrow(IPDMA), replace = TRUE)
  for (i in 1:nfolds) {

    train_dat <- IPDMA[fold_ids != i, ]
    test_dat <- IPDMA[fold_ids == i, ]

    ## Fit default GLM tree
    gt <- glmtree(rawHAMDpost ~ Condition | rawHAMDpre + Gender + Age + JobStat,
                  data = train_dat)
    gt_preds <- predict(gt, newdata = test_dat)
    MSE$gt[(k-1)*10+i] <- mean((gt_preds - test_dat$rawHAMDpost)^2)
```

```

tree_size$gt[(k-1)*10+i] <- (length(gt)-1)/2

## Fit default GLMM tree
gmt <- lmerTree(rawHAMDpost ~ Condition | Study | rawHAMDpre + Gender +
               Age + JobStat, data = train_dat)
gmt_preds <- predict(gmt, newdata = test_dat, re.form = NULL)
MSE$gmt[(k-1)*10+i] <- mean((gmt_preds - test_dat$rawHAMDpost)^2)
tree_size$gmt[(k-1)*10+i] <- (length(gmt$tree)-1)/2

## Fit BART
br <- bart2(rawHAMDpost ~ rawHAMDpre + Condition + Gender + Age + JobStat,
            data = train_dat, n.trees = 200, keepTrees = TRUE, verbose = FALSE)
postp <- predict(br, newdata = test_dat, type = "ppd")
postm <- apply(postp, 2, median)
MSE$bart[(k-1)*10+i] <- mean((postm - test_dat$rawHAMDpost)^2)

## Prepare surrogate data
surr_dat <- train_dat[, -which(names(train_dat) == "Condition")]
surr_dat <- rbind(surr_dat, surr_dat)
surr_dat$Condition <- factor(rep(c("ADM", "ADM+STPP"), each = nrow(train_dat)))

## Fit multilevel BART
br_vi <- rBart_vi(rawHAMDpost ~ rawHAMDpre + Condition + Gender + Age + JobStat,
                 data = train_dat, group.by = train_dat$Study,
                 n.trees = 200, keepTrees = TRUE, verbose = FALSE,
                 test = rbind(test_dat, surr_dat),
                 group.by.test = c(test_dat$Study, surr_dat$Study))
postp_vi <- fitted(br_vi, type = "ppd", sample = "test")[1:nrow(test_dat)]
MSE$bart_vi[(k-1)*10+i] <- mean((postp_vi - test_dat$rawHAMDpost)^2)

## Fit born-again GLMM tree
surr_dat$rawHAMDpost <- fitted(br_vi, type = "ppd", sample = "test")[-(1:nrow(test_dat))]
gmt_surr <- lmerTree(rawHAMDpost ~ Condition | Study | rawHAMDpre + Gender +
                    Age + JobStat, data = surr_dat,
                    weights = rep(.5, times = nrow(surr_dat)))
surr_m_preds <- predict(gmt_surr, newdata = test_dat, re.form = NULL)
MSE$surr_m[(k-1)*10+i] <- mean((surr_m_preds - test_dat$rawHAMDpost)^2)
tree_size$surr_m[(k-1)*10+i] <- (length(gmt_surr$tree)-1)/2

## Fit born-again GLM tree
postp_surr <- predict(br, newdata = surr_dat, type = "ppd")
surr_dat$rawHAMDpost <- apply(postp_surr, 2, median)
gt_surr <- glmTree(rawHAMDpost ~ Condition | rawHAMDpre + Gender + Age + JobStat,
                  data = surr_dat, weights = rep(.5, times = nrow(surr_dat)))
surr_preds <- predict(gt_surr, newdata = test_dat)
MSE$surr[(k-1)*10+i] <- mean((surr_preds - test_dat$rawHAMDpost)^2)
tree_size$surr[(k-1)*10+i] <- (length(gt_surr)-1)/2
}
}
saveRDS(MSE, "MSE_ipdma.RDS")
saveRDS(tree_size, "treesize_ipdma.RDS")

```

## Evaluate and compare performance

```
MSE <- readRDS("MSE_ipdma.RDS")
tree_size <- readRDS("treesize_ipdma.RDS")

## Benchmark
var(IPDMA$rawHAMDpost)

## [1] 89.77962

## Fitted models (left-most 3 are fixed-effects, right-most three are mixed-effects)
sapply(MSE, mean)[c(1, 4, 6, 2, 3, 7)]

##      gt      surr      bart      gmt      surr_m      bart_vi
## 60.74533 56.77023 55.97104 46.11635 43.57281 44.94500

sapply(MSE, sd)[c(1, 4, 6, 2, 3, 7)]

##      gt      surr      bart      gmt      surr_m      bart_vi
## 14.20744 12.40908 12.77231 12.16009 11.37879 11.64635

sapply(tree_size[, c(1, 4, 2, 3)], mean)

##      gt      surr      gmt      surr_m
##   2.91   9.90   2.58   9.57

sapply(tree_size[, c(1, 4, 2, 3)], sd)

##      gt      surr      gmt      surr_m
## 0.6046119 1.2350111 0.5717243 1.2330057

## Plot performance of standard (non-hierarchical) models
MSE_l <- stack(MSE[, c("gt", "surr", "bart")])
levels(MSE_l$ind) <- c("GLM tree", "BART_based \nGLM tree", "BART")
level_order <- c("GLM tree", "BART", "BART_based \nGLM tree")
## Compute credible intervals
set.seed(42)
mod_acc <- stan_glm(values ~ 0 + ind, data = MSE_l, refresh=0)
post_acc <- describe_posterior(mod_acc)
sum_dat <- data.frame(acc_med = post_acc$Median,
                      acc_min = post_acc$CI_low,
                      acc_max = post_acc$CI_high)
sum_dat$approach <- factor(c("GLM tree", "BART_based \nGLM tree", "BART"))

ap1 <- ggplot(MSE_l, aes(x = factor(ind, levels = level_order), y = values, col = ind)) +
  geom_point(position = position_jitterdodge(jitter.width = .2, jitter.height = 0,
                                             dodge.width = 0, seed = 12),
             size = 1, alpha = 0.6) +
  geom_errorbar(data = sum_dat,
               aes(x = factor(approach, levels = level_order), y = acc_med, ymin = acc_min, max = acc_max,
                    position = position_nudge(0), colour = "BLACK", width = 0.05, size = 1) +
  ylab("Mean Squared Error") + xlab("") +
  ggtitle("Accuracy for IPDMA on STPP. Based on 10 repeats of 10-fold CV. Black bars are 95% credible intervals") +
  theme(axis.text.x = element_text(hjust = .25),
        axis.ticks.x = element_line(color = "white"),
        legend.position = "none",
        plot.title = element_text(size = 7))
```

```

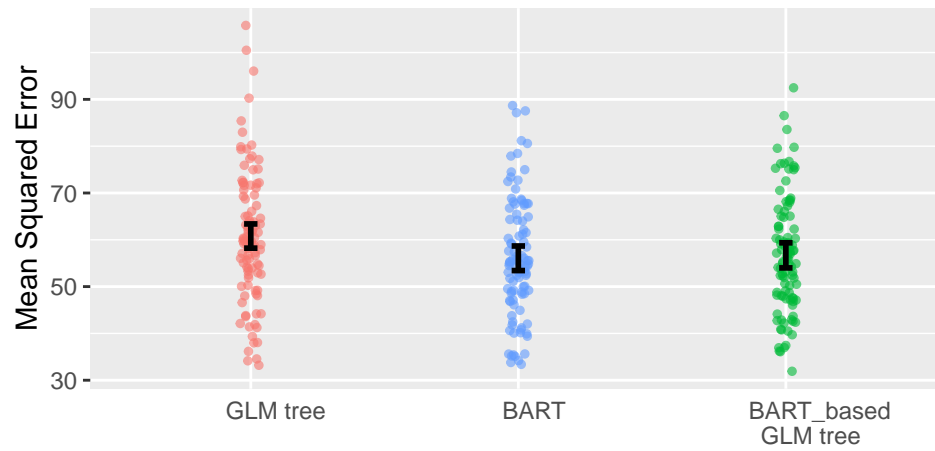
## Plot performance of multilevel models
MSE_l <- stack(MSE[, c("gmt", "surr_m", "bart_vi")])
levels(MSE_l$ind) <- c("GLMM tree", "M-BART_based \nGLMM tree", "M-BART")
set.seed(42)
mod_acc <- stan_glm(values ~ 0 + ind, data = MSE_l, refresh=0)
post_acc <- describe_posterior(mod_acc)
sum_dat <- data.frame(acc_med = post_acc$Median,
                      acc_min = post_acc$CI_low,
                      acc_max = post_acc$CI_high)
sum_dat$approach <- factor(c("GLMM tree", "M-BART_based \nGLMM tree", "M-BART"))
ap2 <- ggplot(MSE_l, aes(x = ind, y = values, col = ind)) +
  geom_point(position = position_jitterdodge(jitter.width = .2, jitter.height = 0,
                                             dodge.width = 0, seed = 12),
             size = 1, alpha = 0.6) +
  geom_errorbar(data = sum_dat,
               aes(x = approach, y = acc_med, ymin = acc_min, max = acc_max),
               position = position_nudge(0), colour = "BLACK", width = 0.05, size = 1) +
  ylab("Mean Squared Error") + xlab("") +
  ggtitle("Accuracy for IPDMA on STPP. Based on 10 repeats of 10-fold CV. Black bars are 95% credible i")
  theme(axis.text.x = element_text(hjust = .25),
        axis.ticks.x = element_line(color = "white"),
        legend.position = "none",
        plot.title = element_text(size = 7))

## Plot treesize
treesize_l <- stack(tree_size)
levels(treesize_l$ind) <- c("GLM tree", "GLMM tree", "M-BART_based \nGLMM tree",
                          "BART_based \nGLM tree")
ap3 <- ggplot(treesize_l, aes(x = ind, y = values, col = ind)) +
  geom_point(position = position_jitterdodge(jitter.width = .2, jitter.height = 0,
                                             dodge.width = 0, seed = 12),
             size = 1, alpha = 0.6) +
  ylab("Number of splits") + xlab("") +
  ggtitle("Tree size for IPDMA on STPP. Based on 10 repeats of 10-fold CV.") +
  theme(axis.text.x = element_text(hjust=.25),
        axis.ticks.x = element_line(color="white"),
        legend.position = "none",
        plot.title = element_text(size = 7))

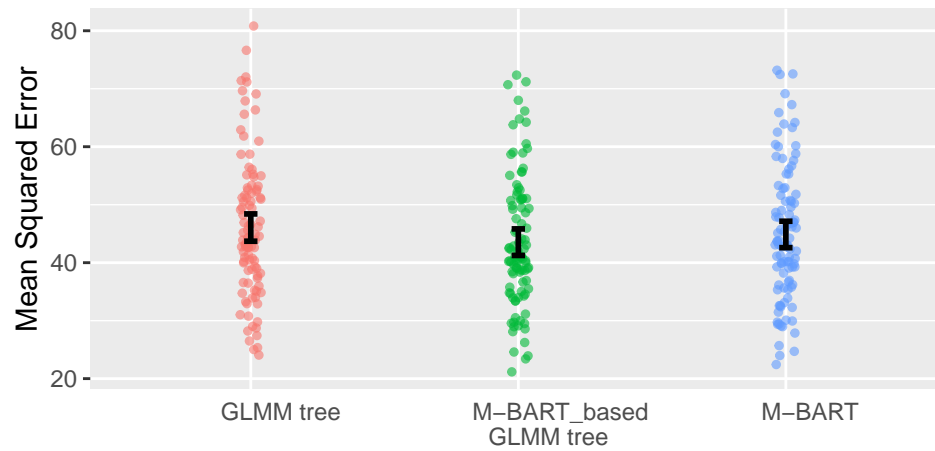
print(ap1); print(ap2); print(ap3)

```

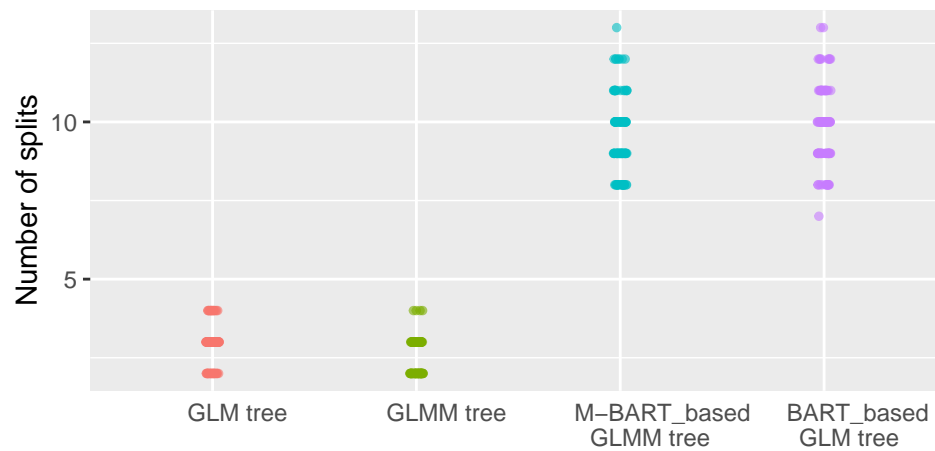
Accuracy for IPDMA on STPP. Based on 10 repeats of 10-fold CV. Black bars are 95% credible intervals.



Accuracy for IPDMA on STPP. Based on 10 repeats of 10-fold CV. Black bars are 95% credible intervals.



Tree size for IPDMA on STPP. Based on 10 repeats of 10-fold CV.



Differences in predictive accuracy get more dramatic if we increase number of repetitions of 10-fold CV from 10 to 100:

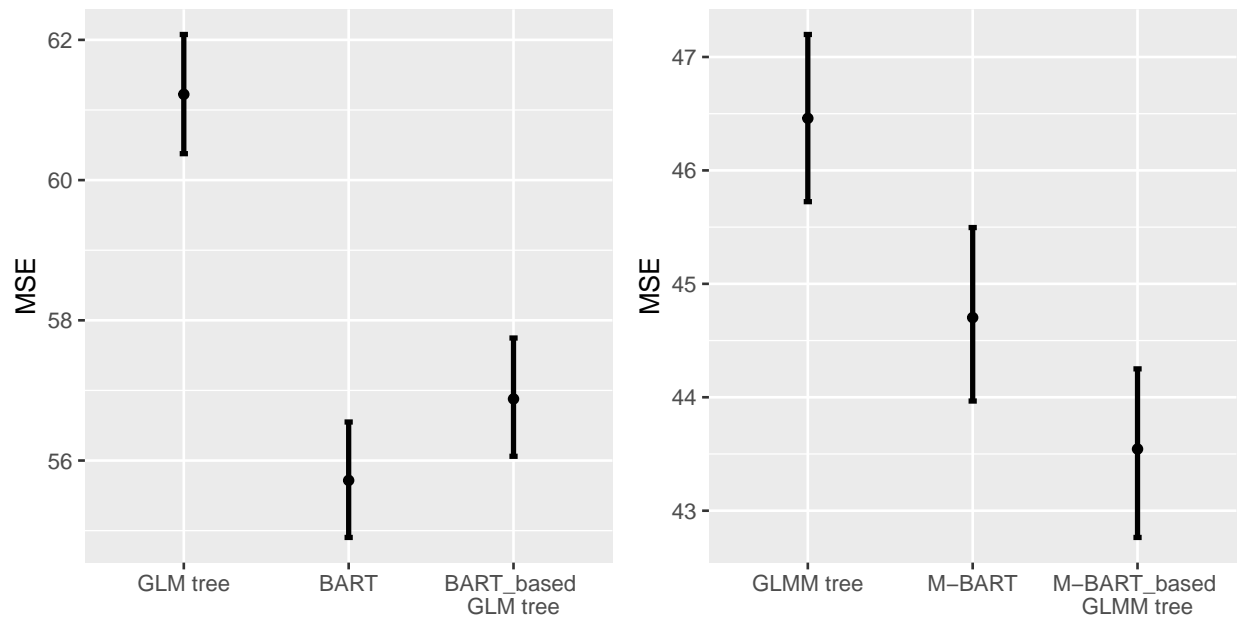
```

MSE <- readRDS("MSE_ipdma 100 reps of CV.RDS")
library("gridExtra")

## Standard (non-hierarchical) models
MSE_l <- stack(MSE[, c("gt", "surr", "bart")])
levels(MSE_l$ind) <- c("GLM tree", "BART_based \nGLM tree", "BART")
level_order <- c("GLM tree", "BART", "BART_based \nGLM tree")
set.seed(42)
mod_acc <- stan_glm(values ~ 0 + ind, data = MSE_l, refresh=0)
post_acc <- describe_posterior(mod_acc)
sum_dat <- data.frame(acc_med = post_acc$Median,
                      acc_min = post_acc$CI_low,
                      acc_max = post_acc$CI_high)
sum_dat$approach <- factor(c("GLM tree", "BART_based \nGLM tree", "BART"))
ap1 <- ggplot(MSE_l, aes(x = factor(ind, levels = level_order), y = values, col = ind)) +
  geom_errorbar(data = sum_dat,
               aes(x = factor(approach, levels = level_order), y = acc_med,
                    ymin = acc_min, max = acc_max),
               position = position_nudge(0), colour = "BLACK", width = 0.05, size = 1) +
  geom_point(data = sum_dat, aes(x = approach, y = acc_med),
            position = position_nudge(0), colour = "BLACK") +
  xlab(" ") + ylab("MSE") + ggtitle(" ")

## Multilevel models
MSE_l <- stack(MSE[, c("gmt", "surr_m", "bart_vi")])
levels(MSE_l$ind) <- c("GLMM tree", "M-BART_based \nGLMM tree", "M-BART")
set.seed(42)
mod_acc <- stan_glm(values ~ 0 + ind, data = MSE_l, refresh=0)
post_acc <- describe_posterior(mod_acc)
sum_dat <- data.frame(acc_med = post_acc$Median,
                      acc_min = post_acc$CI_low,
                      acc_max = post_acc$CI_high)
sum_dat$approach <- factor(c("GLMM tree", "M-BART_based \nGLMM tree", "M-BART"))
ap2 <- ggplot(MSE_l, aes(x = ind, y = values, col = ind)) +
  geom_errorbar(data = sum_dat,
               aes(x = approach, y = acc_med, ymin = acc_min, max = acc_max),
               position = position_nudge(0), colour = "BLACK", width = 0.05, size = 1) +
  geom_point(data = sum_dat, aes(x = approach, y = acc_med),
            position = position_nudge(0), colour = "BLACK") +
  xlab(" ") + ylab("MSE") + ggtitle(" ")
grid.arrange(ap1, ap2, nrow = 1L)

```



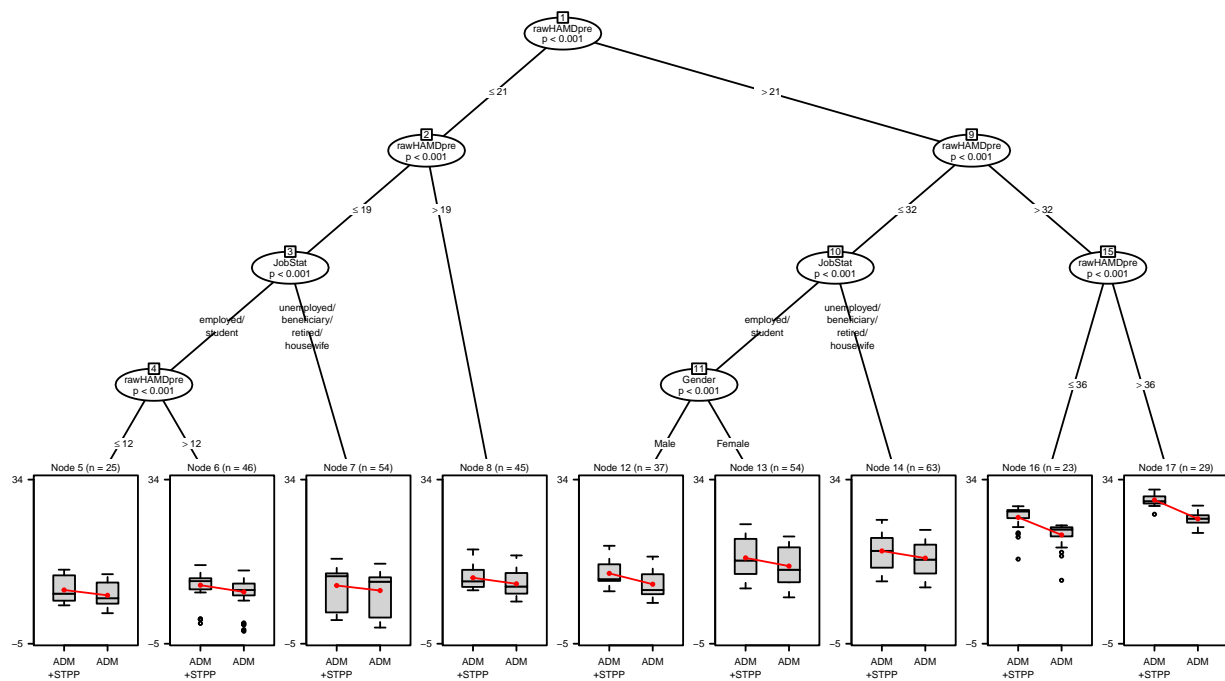
Fit and plot tree on complete data:

```
## Fit surrogate GLMM tree
surr_dat <- IPDMA[ , -which(names(IPDMA) == "Condition")]
surr_dat <- rbind(surr_dat, surr_dat)
surr_dat$Condition <- factor(rep(c("ADM", "\nADM\n+STPP"), each = nrow(IPDMA)))

set.seed(42)
br_vi <- rbart_vi(rawHAMDpost ~ rawHAMDpre + Condition + Gender + Age + JobStat,
  data = IPDMA, group.by = IPDMA$Study, n.trees = 200,
  keepTrees = TRUE, verbose = FALSE, test = surr_dat,
  group.by.test = surr_dat$Study)
surr_dat$rawHAMDpost <- fitted(br_vi, type = "ppd", sample = "test")

levels(surr_dat$JobStat) <- c("employed/\nstudent",
  "unemployed/\nbeneficiary/\nretired/\nhousewife")
gmt_surr <- lmertree(rawHAMDpost ~ Condition | Study | rawHAMDpre + Gender + Age + JobStat,
  data = surr_dat, weights = rep(.5, times = nrow(surr_dat)))
plot(gmt_surr$tree, gp = gpar(cex = .3))
```





## Experiment 2: IPDMA CBT versus PHA

```
## Prepare data
metadata <- read.dta("Database IPDMA CBT PHA Version 11.dta")
metadata[metadata == 999] <- NA
metadata[metadata == 888] <- NA
vars <- c("studyid", "Tx_group", "Age", "Gender", "education",
          "ComorbidAnxietyDisorder", "HRSDt0", "HRSDt1")
factors <- c("studyid", "Tx_group", "Gender", "education", "ComorbidAnxietyDisorder")
metadata$education <- factor(metadata$education, ordered = T)

for (i in 1:length(factors)) {
  metadata[,factors[i]] <- factor(metadata[,factors[i]])
}
metadata <- metadata[vars] # select only relevant variables
metadata <- metadata[complete.cases(metadata[,vars]),] # select only complete data
metadata <- metadata[!metadata$Tx_group == "placebo",] # remove placebo observations
metadata$Tx_group <- factor(metadata$Tx_group)
nrow(metadata) # total number of observations
```

```
## [1] 694
```

```
table(metadata$studyid) # participants per
```

```
##
##      David      Mohr      Hegerl      Jarett      DeRubeis      Faramarzi      Elkin
##          0          0          0          58          150           0          78
##      Rush      Miranda      Hollon      Kennedy      Dunlop      Zegal      Murphy
##          0          128          0          0          64          147          0
##      Dunner Dimidjian
##          0          69
```

Run the experiment:

```
nreps <- 10L
nfolds <- 10L
set.seed(42)
tree_size <- MSE <- data.frame(gt = rep(NA, times = nreps*nfolds))
MSE$bart <- MSE$bart_vi <- MSE$surr_m <- MSE$surr <- MSE$gmt <- MSE$gt
tree_size$surr <- tree_size$gmt <- tree_size$surr_m <- tree_size$gt
set.seed(42)
for (k in 1:nreps) {
  fold_ids <- sample(rep(1:10, times = ceiling(nrow(metadata)/nfolds)),
                    size = nrow(metadata), replace = TRUE)

  for (i in 1:nfolds) {

    train_dat <- metadata[fold_ids != i, ]
    test_dat <- metadata[fold_ids == i, ]

    ## Fit GLM tree
    gt <- glmtree(HRSDt1 ~ Tx_group | HRSDt0 + Gender + Age + education +
                  ComorbidAnxietyDisorder, data = train_dat)
    gt_preds <- predict(gt, newdata = test_dat)
    MSE$gt[(k-1)*10+i] <- mean((gt_preds - test_dat$HRSDt1)^2)
    tree_size$gt[(k-1)*10+i] <- (length(gt)-1)/2
```

```

## Fit GLMM trees
gmt <- lmertree(HRSDt1 ~ Tx_group | studyid | HRSDt0 + Gender + Age +
               education + ComorbidAnxietyDisorder, data = train_dat)
gmt_preds <- predict(gmt, newdata = test_dat, re.form = NULL)
MSE$gmt[(k-1)*10+i] <- mean((gmt_preds - test_dat$HRSDt1)^2)
tree_size$gmt[(k-1)*10+i] <- (length(gmt$tree)-1)/2

## Fit BART
br <- bart2(HRSDt1 ~ HRSDt0 + Tx_group + Gender + Age + education +
            ComorbidAnxietyDisorder, data = train_dat, n.trees = 200,
            keepTrees = TRUE, verbose = FALSE)
postp <- predict(br, newdata = test_dat, type = "ppd")
postm <- apply(postp, 2, median)
MSE$bart[(k-1)*10+i] <- mean((postm - test_dat$HRSDt1)^2)

## Prepare surrogate data
surr_dat <- train_dat[, -which(names(train_dat) == "Tx_group")]
surr_dat <- rbind(surr_dat, surr_dat)
surr_dat$Tx_group <- factor(rep(c("CBT", "PHA"), each = nrow(train_dat)))

## Fit multilevel BART
br_vi <- rbart_vi(HRSDt1 ~ HRSDt0 + Tx_group + Gender + Age + education +
                  ComorbidAnxietyDisorder, data = train_dat,
                  group.by = train_dat$studyid, n.trees = 200, keepTrees = TRUE,
                  verbose = FALSE, test = rbind(test_dat, surr_dat),
                  group.by.test = c(test_dat$studyid, surr_dat$studyid))
postp_vi <- fitted(br_vi, type = "ppd", sample = "test")[1:nrow(test_dat)]
MSE$bart_vi[(k-1)*10+i] <- mean((postp_vi - test_dat$HRSDt1)^2)

## Fit born-again GLMM tree
surr_dat$HRSD_t1 <- fitted(br_vi, type = "ppd", sample = "test")[-(1:nrow(test_dat))]
gmt_surr <- lmertree(HRSDt1 ~ Tx_group | studyid | HRSDt0 + Gender + Age +
                    education + ComorbidAnxietyDisorder, data = surr_dat,
                    weights = rep(.5, times = nrow(surr_dat)))
surr_m_preds <- predict(gmt_surr, newdata = test_dat, re.form = NULL)
MSE$surr_m[(k-1)*10+i] <- mean((surr_m_preds - test_dat$HRSDt1)^2)
tree_size$surr_m[(k-1)*10+i] <- (length(gmt_surr$tree)-1)/2

## Fit born-again GLM trees
postp_surr <- predict(br, newdata = surr_dat)
surr_dat$HRSD_t1 <- apply(postp_surr, 2, median)
gt_surr <- glmertree(HRSDt1 ~ Tx_group | HRSDt0 + Gender + Age + education +
                    ComorbidAnxietyDisorder, data = surr_dat,
                    weights = rep(.5, times = nrow(surr_dat)))
surr_preds <- predict(gt_surr, newdata = test_dat)
MSE$surr[(k-1)*10+i] <- mean((surr_preds - test_dat$HRSDt1)^2)
tree_size$surr[(k-1)*10+i] <- (length(gt_surr)-1)/2
}
}
saveRDS(MSE, "MSE_metadata.RDS")
saveRDS(tree_size, "treesize_metadata.RDS")

```

## Evaluate and compare performance

```
MSE <- readRDS("MSE_metadata.RDS")
tree_size <- readRDS("treesize_metadata.RDS")

## Benchmark
var(metadata$HRSdt1)

## [1] 39.24745

## Fitted models (left-most 3 are fixed-effects, right-most three are mixed-effects)
sapply(MSE, mean)[c(1, 3, 6, 2, 4, 5)]
```

```
##      gt      surr      bart      gmt      surr_m      bart_vi
## 38.13097 37.89664 36.98055 36.38229 35.91066 35.83187

sapply(MSE, sd)[c(1, 3, 6, 2, 4, 5)]
```

```
##      gt      surr      bart      gmt      surr_m      bart_vi
## 6.168936 6.173103 5.944282 5.857651 5.752931 5.663875

sapply(tree_size[, c(1, 4, 3, 2)], mean)
```

```
##      gt      surr      gmt      surr_m
## 2.69 1.83 2.58 1.94

sapply(tree_size[, c(1, 4, 3, 2)], sd)
```

```
##      gt      surr      gmt      surr_m
## 0.8609954 0.3775252 0.7272475 0.2777980
```

Plot performance:

```
## Standard (non-hierarchical) models
MSE_l <- stack(MSE[, c("gt", "surr", "bart")])
levels(MSE_l$ind) <- c("GLM tree", "BART_based \nGLM tree", "BART")
level_order <- c("GLM tree", "BART", "BART_based \nGLM tree")
## Compute credible intervals
set.seed(42)
mod_acc <- stan_glm(values ~ 0 + ind, data = MSE_l, refresh=0)
post_acc <- describe_posterior(mod_acc)
sum_dat <- data.frame(acc_med = post_acc$Median,
                      acc_min = post_acc$CI_low,
                      acc_max = post_acc$CI_high)
sum_dat$approach <- factor(c("GLM tree", "BART_based \nGLM tree", "BART"))
ap1 <- ggplot(MSE_l,
              aes(x = ind, y = values, col = ind)) +
  geom_point(position = position_jitterdodge(jitter.width = .2, jitter.height = 0,
                                             dodge.width = 0, seed = 12),
             size = 1, alpha = 0.6) + ylab("Mean Squared Error") + xlab("") +
  geom_errorbar(data = sum_dat,
               aes(x = factor(approach, levels = level_order), y = acc_med, ymin = acc_min, max = acc_max),
               position = position_nudge(0), colour = "BLACK", width = 0.05, size = 1) +
  ggtitle("Accuracy for IPDMA on CBT vs PHA, Based on 10 repeats of 10-fold CV. Black bars are 95% credible intervals")
theme(axis.text.x = element_text(hjust=.25),
      axis.ticks.x = element_line(color="white"),
      legend.position = "none",
      plot.title = element_text(size = 7))
```

```

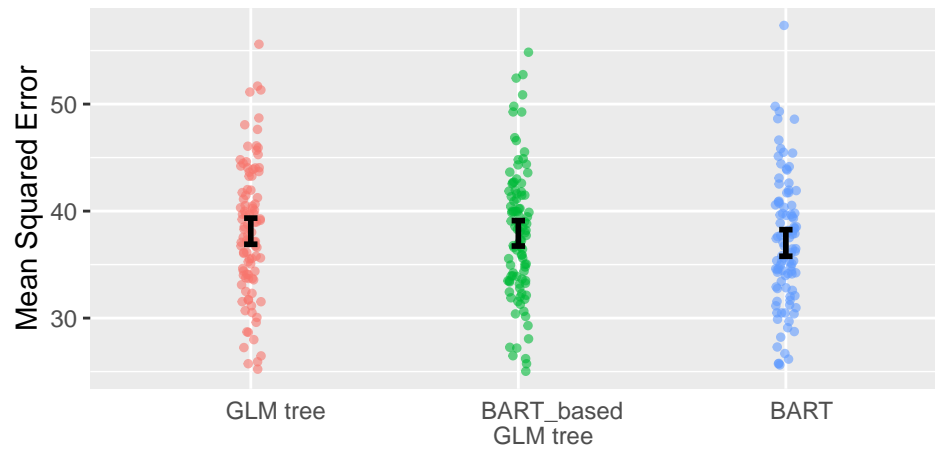
## Multilevel models
MSE_l <- stack(MSE[, c("gmt", "surr_m", "bart_vi")])
levels(MSE_l$ind) <- c("GLMM tree", "M-BART_based \nGLMM tree", "M-BART")
set.seed(42)
mod_acc <- stan_glm(values ~ 0 + ind, data = MSE_l, refresh=0)
post_acc <- describe_posterior(mod_acc)
sum_dat <- data.frame(acc_med = post_acc$Median,
                      acc_min = post_acc$CI_low,
                      acc_max = post_acc$CI_high)
sum_dat$approach <- factor(c("GLMM tree", "M-BART_based \nGLMM tree", "M-BART"))
ap2 <- ggplot(MSE_l, aes(x = ind, y = values, col = ind)) +
  geom_point(position = position_jitterdodge(jitter.width = .2, jitter.height = 0,
                                             dodge.width = 0, seed = 12),
             size = 1, alpha = 0.6) +
  geom_errorbar(data = sum_dat,
               aes(x = approach, y = acc_med, ymin = acc_min, max = acc_max),
               position = position_nudge(0), colour = "BLACK", width = 0.05, size = 1) +
  ylab("Mean Squared Error") + xlab("") +
  ggtitle("Accuracy for IPDMA on CBT vs PHA, Based on 10 repeats of 10-fold CV. Black bars are 95% cred.")
  theme(axis.text.x = element_text(hjust=.25),
        axis.ticks.x = element_line(color="white"),
        legend.position = "none",
        plot.title = element_text(size = 7))

## Evaluate tree size
treesize_l <- stack(tree_size)
levels(treesize_l$ind) <- c("GLM tree", "GLMM tree", "M-BART_based \nGLMM tree",
                           "BART_based \nGLM tree")
ap3 <- ggplot(treesize_l,
              aes(x = ind, y = values, col = ind)) +
  geom_point(position = position_jitterdodge(jitter.width = .2, jitter.height = 0,
                                             dodge.width = 0, seed = 12),
             size = 1, alpha = 0.6) + ylab("Number of splits") + xlab("") +
  ggtitle("CBT vs PHA IPDMA") +
  theme(axis.text.x = element_text(hjust=.25),
        axis.ticks.x = element_line(color="white"),
        legend.position = "none",
        plot.title = element_text(size = 7))

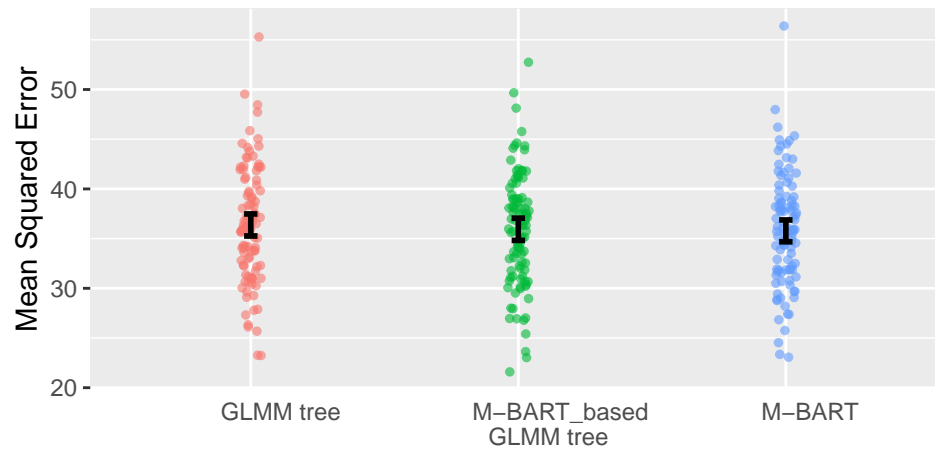
print(ap1); print(ap2); print(ap3)

```

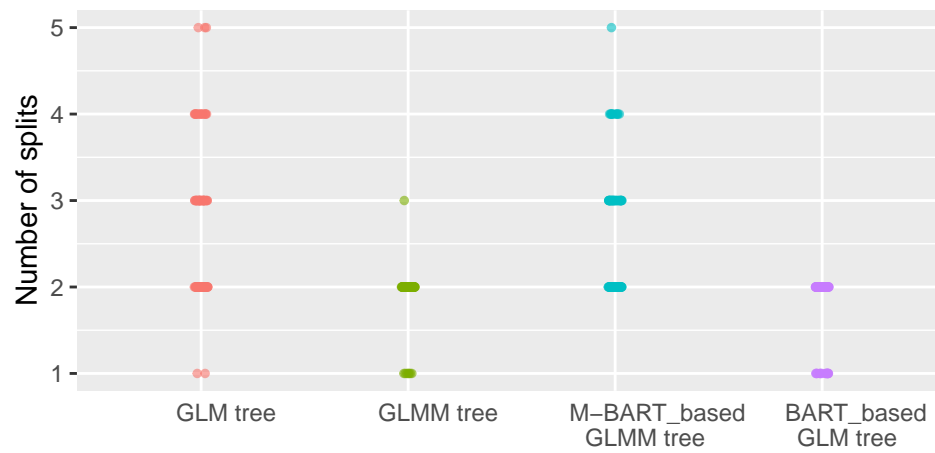
Accuracy for IPDMA on CBT vs PHA, Based on 10 repeats of 10-fold CV. Black bars are 95% credible int



Accuracy for IPDMA on CBT vs PHA, Based on 10 repeats of 10-fold CV. Black bars are 95% credible int



CBT vs PHA IPDMA



## Misc

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Dutch_Netherlands.1252 LC_CTYPE=Dutch_Netherlands.1252
## [3] LC_MONETARY=Dutch_Netherlands.1252 LC_NUMERIC=C
## [5] LC_TIME=Dutch_Netherlands.1252
##
## attached base packages:
## [1] grid      stats      graphics  grDevices utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] gridExtra_2.3      ggplot2_3.3.5      bayestestR_0.10.0  rstanarm_2.21.1
## [5] Rcpp_1.0.7         dbarts_0.9-19      glmertree_0.2-0    partykit_1.2-13
## [9] mvtnorm_1.1-2      libcoin_1.0-8      lme4_1.1-27.1      Matrix_1.3-4
## [13] foreign_0.8-81
##
## loaded via a namespace (and not attached):
## [1] nlme_3.1-152      matrixStats_0.60.0  xts_0.12.1
## [4] insight_0.14.2    threejs_0.3.3       rstan_2.21.2
## [7] tools_4.1.0       utf8_1.2.1          R6_2.5.0
## [10] DT_0.18           rpart_4.1-15        DBI_1.1.1
## [13] colorspace_2.0-2  withr_2.4.2         tidyselect_1.1.1
## [16] prettyunits_1.1.1 processx_3.5.2       curl_4.3.1
## [19] compiler_4.1.0    cli_3.0.0           shinyjs_2.0.0
## [22] labeling_0.4.2    colourpicker_1.1.0  scales_1.1.1
## [25] dygraphs_1.1.1.6  ggribges_0.5.3      callr_3.7.0
## [28] StanHeaders_2.21.0-7 stringr_1.4.0        digest_0.6.27
## [31] minqa_1.2.4       rmarkdown_2.9       base64enc_0.1-3
## [34] pkgconfig_2.0.3   htmltools_0.5.1.1   highr_0.9
## [37] fastmap_1.1.0     htmlwidgets_1.5.3   rlang_0.4.11
## [40] shiny_1.6.0       farver_2.1.0        generics_0.1.0
## [43] jsonlite_1.7.2    zoo_1.8-9           crosstalk_1.1.1
## [46] gtools_3.9.2      dplyr_1.0.7         inline_0.3.19
## [49] magrittr_2.0.1    Formula_1.2-4       loo_2.4.1
## [52] bayesplot_1.8.1   munsell_0.5.0       fansi_0.5.0
## [55] lifecycle_1.0.0   stringi_1.6.2       yaml_2.2.1
## [58] inum_1.0-4        MASS_7.3-54         pkgbuild_1.2.0
## [61] plyr_1.8.6        parallel_4.1.0      promises_1.2.0.1
## [64] crayon_1.4.1      miniUI_0.1.1.1      lattice_0.20-44
## [67] splines_4.1.0     knitr_1.33          ps_1.6.0
## [70] pillar_1.6.1      igraph_1.2.6        boot_1.3-28
## [73] markdown_1.1      shinystan_2.5.0     codetools_0.2-18
## [76] reshape2_1.4.4    stats4_4.1.0        rstantools_2.1.1
## [79] glue_1.4.2        evaluate_0.14       V8_3.4.2
## [82] RcppParallel_5.1.4 vctrs_0.3.8         nloptr_1.2.2.2
```

## [85]	httpuv_1.6.1	gtable_0.3.0	purrr_0.3.4
## [88]	assertthat_0.2.1	xfun_0.24	mime_0.11
## [91]	xtable_1.8-4	later_1.2.0	rsconnect_0.8.24
## [94]	survival_3.2-11	tibble_3.1.2	shinythemes_1.2.0
## [97]	ellipsis_0.3.2		