Some investigation and debugging of multilevel BART-based BA trees

Marjolein Fokkema

11-4-2022

Problem: None of the born-again GLMM trees make splits

Possible causes:

1) Not enough signal in the data, so it is actually realistic and correct that there are no splits.
2) Programming mistakes.
3) We are not drawing from the posterior like we think we do.
4) Inclusion of random effects messes up born-again approach. Perhaps we should first try an approach without random effects.
5) Proposed born-again approach (with draws from BART posterior) does not work.

Example dataset

We load the `safety` dataset, fit a standard GLMM tree and multilevel BART ensemble and evaluate predictive accuracy:

```
library("foreign")
safety <- read.spss("Safety.sav", to.data.frame = TRUE)
```

```
## re-encoding from CP1252
```

```
## Warning in read.spss("Safety.sav", to.data.frame = TRUE): Undeclared level(s) 2,
## 3, 4, 5, 6 added in variable: crowded
```

```
head(safety)
```

```
##   street person age agediv10    sex  economic crowded  age10c  sexc     econc
## 1      1      1  52      5.2   male 0.7438186     low  0.4849 -0.52 0.7438186
## 2      1      2  20      2.0   male 0.7438186     low -2.7151 -0.52 0.7438186
## 3      1      3  44      4.4 female 0.7438186     low -0.3151  0.48 0.7438186
## 4      1      4  43      4.3   male 0.7438186     low -0.4151 -0.52 0.7438186
## 5      1      5  68      6.8   male 0.7438186     low  2.0849 -0.52 0.7438186
## 6      1      6  50      5.0   male 0.7438186     low  0.2849 -0.52 0.7438186
```

```
##   crowdedc    unsafe
## 1    -2.96 sometimes
## 2    -2.96     never
## 3    -2.96     never
## 4    -2.96 sometimes
## 5    -2.96     often
## 6    -2.96     never
```

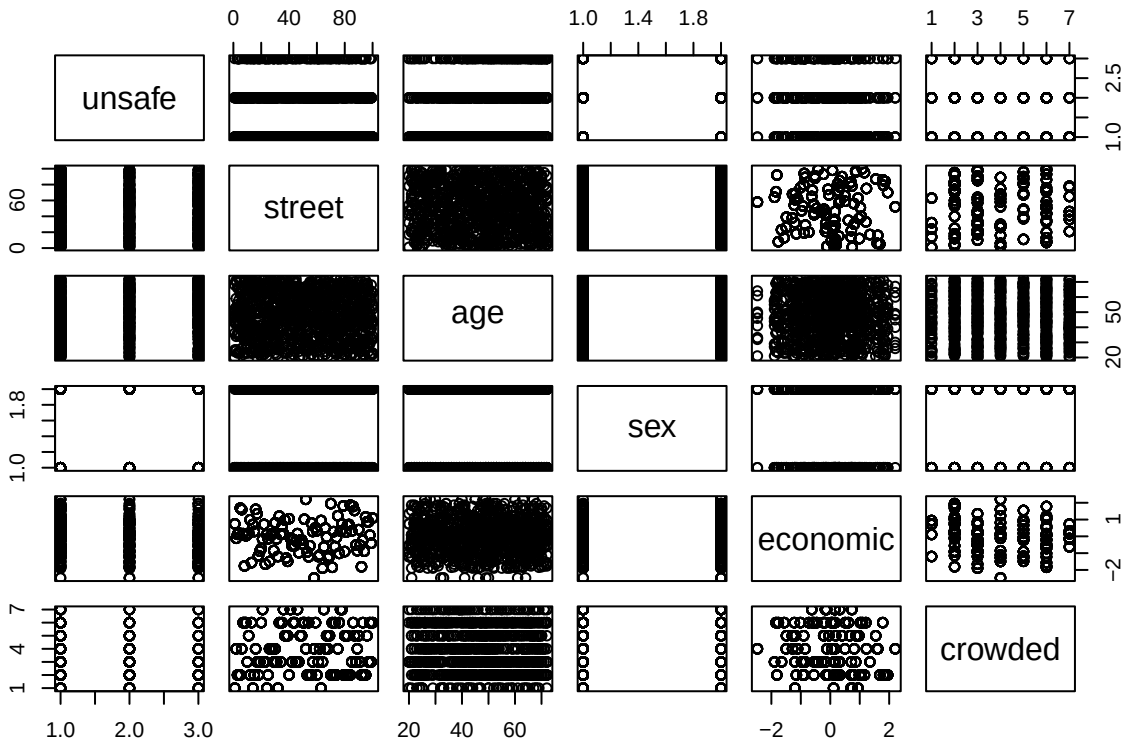
```
dim(safety)
```

```
## [1] 1000   12
```

```
#plot(safety)
```

Response, cluster id and predictor variables are:

```
vars <- c("unsafe", "street", "age", "sex", "economic", "crowded")
plot(safety[ , vars])
```




```
table(safety$unsafe) ## enough variance present
```

```
##
##     never sometimes     often
##       505       306       189
```

```
safety$unsafe <- as.numeric(safety$unsafe)
table(complete.cases(safety[ , vars])) ## only complete cases, nice!
```

```
##
## TRUE
## 1000
```

Select training and test sets:

```
set.seed(420)
train_ids <- sample.int(n = nrow(safety), size = floor(.75*nrow(safety)),
                        replace = F)
safety_train <- safety[train_ids, ]
safety_test  <- safety[-train_ids, ]
var(safety$unsafe)
```

```
## [1] 0.5947387
```

```
var(safety_train$unsafe)
```

```
## [1] 0.5744566
```

```
var(safety_test$unsafe)
```

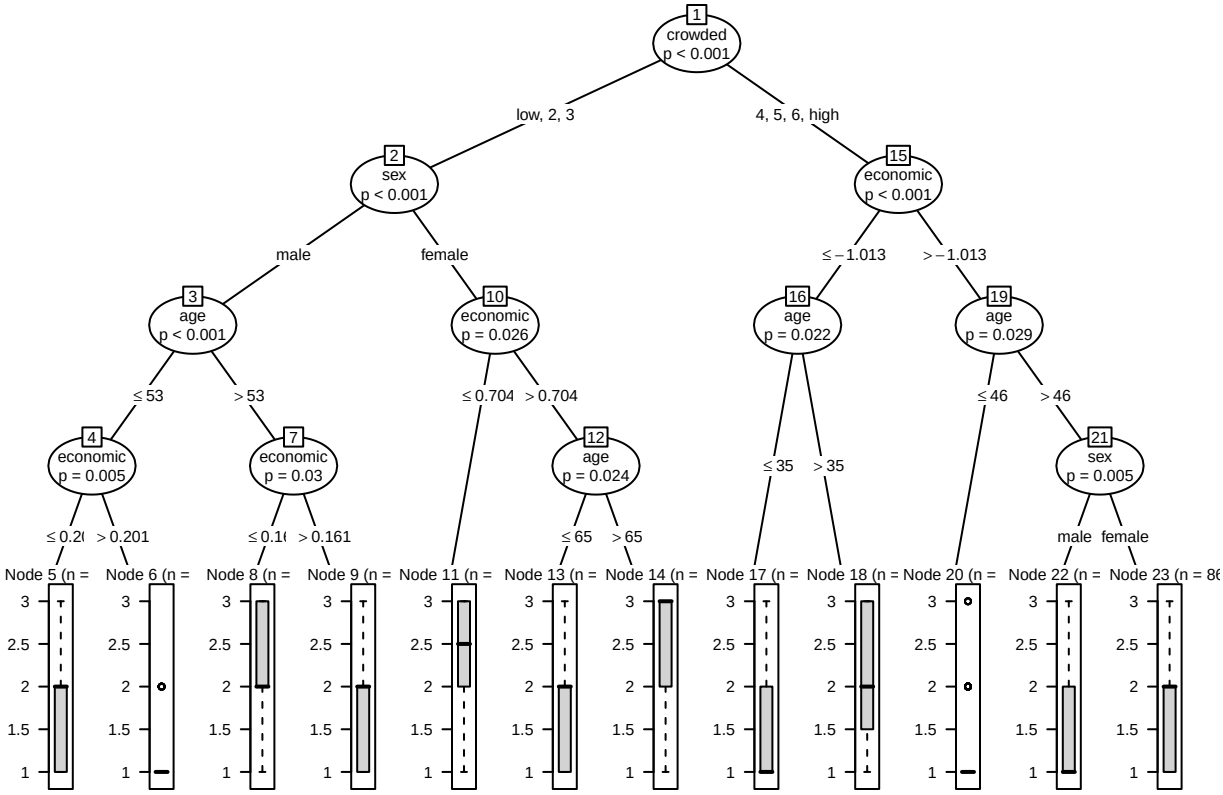
```
## [1] 0.6573655
```

Fit mixed-effects tree

```
library("glmertree")
lmmt <- lmertree(unsafe ~ 1 | street | age + sex + economic + crowded,
                 data = safety_train)
length(lmmt$tree)
```

```
## [1] 23
```

```
plot(lmmt$tree, gp = gpar(cex = .5))
```




```
VarCorr(lmmt)
```

```
##  Groups   Name        Std.Dev.
##  street   (Intercept) 0.10698
##  Residual             0.62917
```

The tree has quite some splits. The ICC is large enough for random effects to be of relevance.

```
lmmt_preds <- predict(lmmt, newdata = safety_test)
mean((lmmt_preds - safety_test$unsafe)^2) ## MSE
```

```
## [1] 0.5561308
```

To check if there is actual signal of relevance in the data, we obtain predictions from the tree only (i.e., random effects assumed 0):

```
lmmt_preds2 <- predict(lmmt, newdata = safety_test, re.form = ~0)
mean((lmmt_preds2 - safety_test$unsafe)^2) ## MSE
```

```
## [1] 0.5699146
```

The random effects are not that influential. There seems to be at least some signal captured by the tree.

Fit multilevel BART ensemble

```
library("dbarts")
set.seed(420)
mbart <- rbart_vi(unsafe ~ age + sex + economic + crowded,
                  data = safety_train, group.by = safety_train$street,
                  test = safety_test, group.by.test = safety_test$street,
                  n.trees = 200, keepTrees = TRUE)
#mbart$fit[[4]]$plotTree(199)
head(mbart$fit[[4]]$getTrees(199)) ## fit is a list with 4 elements, correspond to chains?
```

```
##   sample tree   n var value
## 1      1  199 750  -1     0
## 2      2  199 750  -1     0
## 3      3  199 750  -1     0
## 4      4  199 750  -1     0
## 5      5  199 750  -1     0
## 6      6  199 750  -1     0
```

It appears that tree 199 did not implement any splits. Do any trees have splits?

```
splits <- matrix(NA, ncol = 4, nrow = 200)
rownames(splits) <- paste0("tree", 1:200)
colnames(splits) <- paste0("chain", 1:4)
for (j in 1:4) {
  for (i in 1:200) {
    splits[i, j] <- sum(mbart$fit[[j]]$getTrees(i)$var)
  }
}
table(splits)
```

```
## splits
## -800
##  800
```

No splits seem to be implemented whatsoever. Although I am not entirely sure that having `-1` for `var` means that no split was made.

Obtaining posterior distributions

```
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ppd",
                      group.by = safety_test$street, combineChains = TRUE)
bart_preds_test  <- fitted(object = mbart, type = "ppd", sample = "test")
mean((bart_preds_test - safety_test$unsafe)^2)
```

```
## [1] 0.5076839
```

Multilevel BART seems to do better than the tree, so born-again approach may improve.

Q: Can we extract random effects covariance matrix from multilevel BART object?

Whatever `fitted` does, works pretty well. But we do not know what `fitted` does, nor do we know what `predict` does.

We check what the different options look like for respondents with $y = 1, 2, 3$:

```
par(mfrow = c(2, 2))
## type = "ppd"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ppd",
                      group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

```
## [1] 0.7293736
```

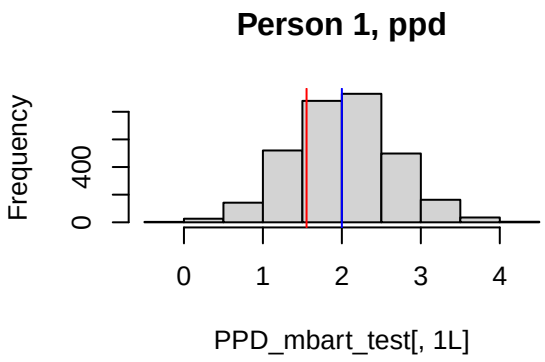
```
hist(PPD_mbart_test[ , 1L], main = "Person 1, ppd")
abline(v = bart_preds_test[1], col = "red") ## red is for 'fitted'
abline(v = safety_test$unsafe[1], col = "blue") # blue is for observed

## type = "bart"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "bart",
                      group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

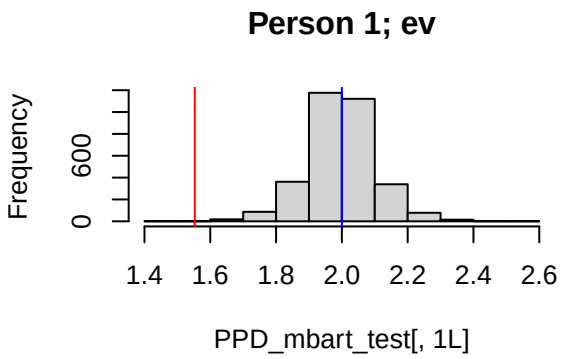
```
## [1] 0.74
```

```
hist(PPD_mbart_test[ , 1L], main = "Person 1, bart")
abline(v = bart_preds_test[1], col = "red")
abline(v = safety_test$unsafe[1], col = "blue")

## type = "ev"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ev",
                      group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

```
## [1] 0.730992
```

```
hist(PPD_mbart_test[ , 1L], main = "Person 1; ev")
abline(v = bart_preds_test[1], col = "red")
abline(v = safety_test$unsafe[1], col = "blue")

## type = "ranef"
#PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ranef",
#                      group.by = safety_test$street, combineChains = TRUE)
## Yields a 3200x0 matrix, which is weird
```






```
par(mfrow = c(2, 2))
## type = "ppd"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ppd",
                          group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

```
## [1] 0.7293736
```

```
hist(PPD_mbart_test[ , 3L], main = "Person 3, ppd")
abline(v = bart_preds_test[3], col = "red") ## red is for 'fitted'
abline(v = safety_test$unsafe[3], col = "blue") # blue is for observed

## type = "bart"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "bart",
                          group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

```
## [1] 0.74
```

```
hist(PPD_mbart_test[ , 3L], main = "Person 3, bart")
abline(v = bart_preds_test[3], col = "red")
abline(v = safety_test$unsafe[3], col = "blue")

## type = "ev"
```

```
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ev",
                          group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

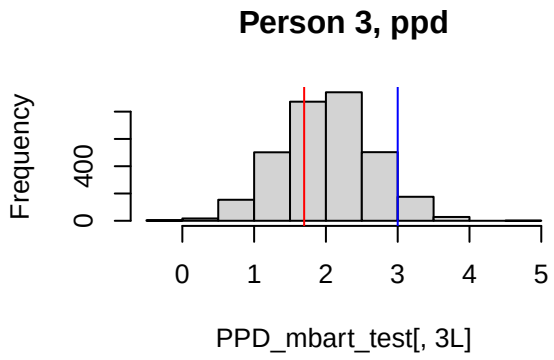
```
## [1] 0.730992
```

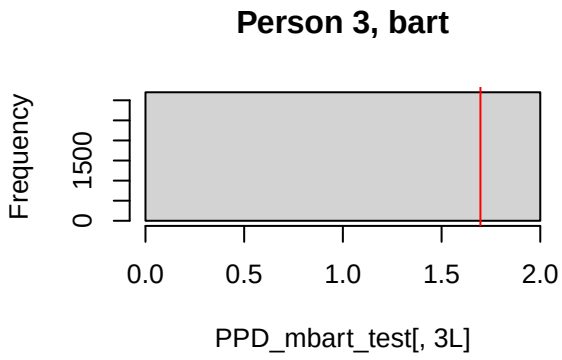
```
hist(PPD_mbart_test[ , 3L], main = "Person 3; ev")
abline(v = bart_preds_test[3], col = "red")
abline(v = safety_test$unsafe[3], col = "blue")
bart_preds_test[3]
```

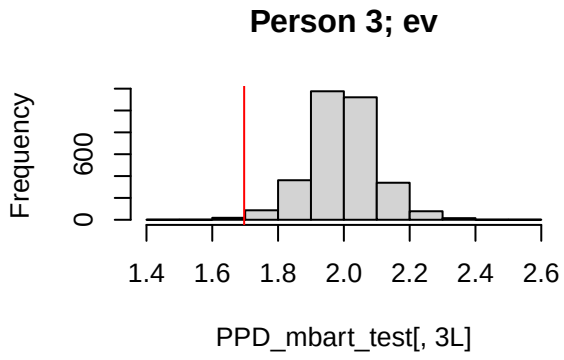
```
## [1] 1.696893
```

```
safety_test$unsafe[3]
```

```
## [1] 3
```








```
par(mfrow = c(2, 2))
## type = "ppd"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ppd",
                          group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

```
## [1] 0.7293736
```

```
hist(PPD_mbart_test[ , 5L], main = "Person 5, ppd")
abline(v = bart_preds_test[5], col = "red") ## red is for 'fitted'
abline(v = safety_test$unsafe[5], col = "blue") # blue is for observed

## type = "bart"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "bart",
                      group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

```
## [1] 0.74
```

```
hist(PPD_mbart_test[ , 5L], main = "Person 5, bart")
abline(v = bart_preds_test[5], col = "red")
abline(v = safety_test$unsafe[5], col = "blue")

## type = "ev"
PPD_mbart_test <- predict(mbart, newdata = safety_test, type = "ev",
                      group.by = safety_test$street, combineChains = TRUE)
mean((safety_test$unsafe - apply(PPD_mbart_test, 2, median))^2)
```

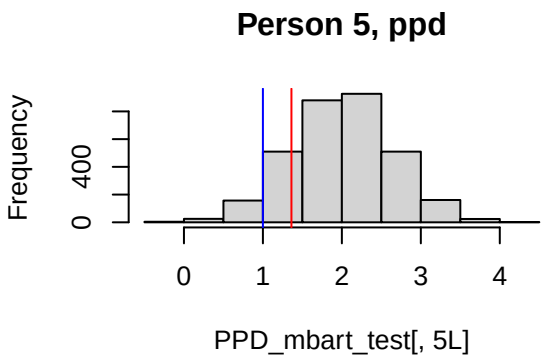
```
## [1] 0.730992
```

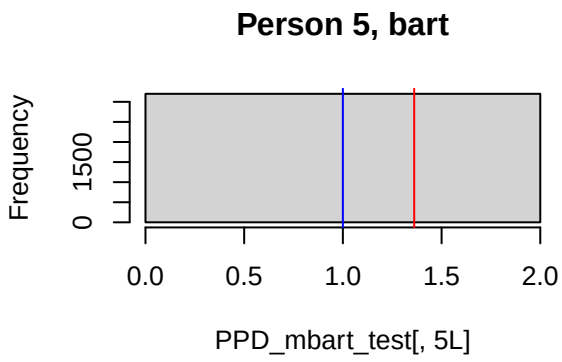
```
hist(PPD_mbart_test[ , 5L], main = "Person 5; ev")
abline(v = bart_preds_test[5], col = "red")
abline(v = safety_test$unsafe[5], col = "blue")
bart_preds_test[5]
```

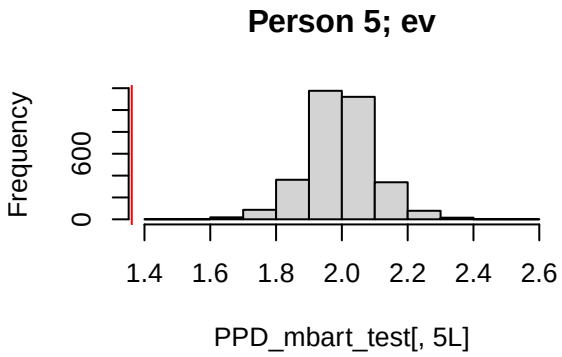
```
## [1] 1.361389
```

```
safety_test$unsafe[5]
```

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







According to the documentation of `rbart_vi`, this is the model:

$y_i \sim N(f(x_i) + \alpha_{g_i}, \sigma^2)$

$\alpha_j \sim N(0, \tau^2)$

So, y_i follows normal distribution with mean (`"ev"`) $f(x_i) + \alpha_{g_i}$ and variance σ^2. I thus suspect that `"ev"` returns samples from $f(k_i)+\alpha_{g_i}$, `"ppd"` returns samples from $f(k_i)+\alpha_{g_i}$ as well as samples from the residuals (which have variance σ^2). We don't know what `"bart"` returns, but it would make sense if it is $f(k_i)$; but counter intuitively this seems to be a constant in the current dataset

PPD has a column for each observation, and a row (why 3200?) for each MCMC sampling iteration?

I have no idea how predictions are generated from the posterior distribution. What I do know is that fitted does not return the median of the PPD returned by predict:

```
par(mfrow = c(1, 2))
cor(bart_preds_test, apply(PPD_mbart_test, 2, median))
```

```
## [1] 0.1942601
```

```
cor(bart_preds_test, apply(PPD_mbart_test, 2, mean))
```

```
## [1] 0.1961839
```

```
var(bart_preds_test)
```

```
## [1] 0.1553448
```

```
dim(PPD_mbart_test)
```

```
## [1] 3200  250
```

```
dim(safety_test)
```

```
## [1] 250  12
```

```
PPD_mbart_test2 <- predict(mbart, newdata = safety_test, type = "ppd",
                     group.by = safety_test$street, combineChains = FALSE)
cor(bart_preds_test, apply(PPD_mbart_test2[1, 1:800, 1:250], 2, median))
```

```
## [1] 0.2086044
```

```
cor(bart_preds_test, apply(PPD_mbart_test2[1, 1:800, 1:250], 2, mean))
```

```
## [1] 0.1742957
```

```
PPD_mbart_test3 <- predict(mbart, newdata = safety_test, type = "bart",
                     group.by = safety_test$street, combineChains = TRUE)
cor(bart_preds_test, apply(PPD_mbart_test3, 2, median))
```

```
## Warning in cor(bart_preds_test, apply(PPD_mbart_test3, 2, median)): the standard
## deviation is zero
```

```
## [1] NA
```

```
cor(bart_preds_test, apply(PPD_mbart_test3, 2, mean))
```

```
## Warning in cor(bart_preds_test, apply(PPD_mbart_test3, 2, mean)): the standard
## deviation is zero
```

```
## [1] NA
```

```
PPD_mbart_test4 <- predict(mbart, newdata = safety_test, type = "ev",
                     group.by = safety_test$street, combineChains = TRUE)
cor(bart_preds_test, apply(PPD_mbart_test4, 2, median))
```

```
## [1] 0.1942601
```

```
cor(bart_preds_test, apply(PPD_mbart_test4, 2, mean))
```

```
## [1] 0.1961839
```

```
var(mbart$ranef.mean)
```

```
## [1] 0.00175122
```

```
var(lmmt$ranef$street)
```

```
##             (Intercept)
## (Intercept) 0.001951822
```

```
str(mbart$tau)
```

```
##  num [1:4, 1:800] 0.1007 0.0947 0.1208 0.0875 0.1103 ...
```

```
mbart$fitted
```

```
## NULL
```

TODO

With the `safety` data, it appears we obtain an ensemble without any splits? We need a dataset which surely splits. The `MHserviceDemo` data from package `glmertree` is one to try.

Born again approach using constant prediction

```
art_gen_dat <- safety_train
bart_preds_train  <- fitted(object = mbart, type = "ppd", sample = "train")
art_gen_dat$unsafe <-  bart_preds_train
lmmt_ba <- lmertree(unsafe ~ 1 | street | age + sex + economic + crowded,
                    data = art_gen_dat)
length(lmmt_ba$tree)
```

```
## [1] 93
```

```
VarCorr(lmmt_ba)
```

```
##  Groups   Name        Std.Dev.
##  street   (Intercept) 0.163428
##  Residual             0.027028
```

The BA tree is much bigger than the original tree. The variance of the random effects has remained the same.

We assess predictive accuracy and fidelity to black-box predictions:

```
lmmt_ba_preds <- predict(lmmt_ba, newdata = safety_test)
mean((lmmt_ba_preds - safety_test$unsafe)^2) ## Accuracy MSE
```

```
## [1] 0.5107559
```

```
mean((lmmt_ba_preds - bart_preds_test)^2) ## Fidelity MSE
```

```
## [1] 0.001115562
```

Fidelity is very high (BA tree might simply mimic all splits from BART ensemble?), predictive accuracy has improved, though at price of increased complexity.

Born again approach using full PPD

I have no idea how to get the (right) posterior distribution at the moment.

```
PPD_mbart_train <- predict(mbart, newdata = safety_train, type = "ppd",
                     group.by = safety_train$street, combineChains = TRUE)
dim(PPD_mbart_train)
```

```
## [1] 3200  750
```

```
## stack rows of PPD
#as.vector(PPD_mbart_train)
#art_gen_dat_large <- cbind(safety_train, )
#bart_preds_train  <- fitted(object = mbart, type = "ppd", sample = "train")
#art_gen_dat$unsafe <-  bart_preds_train
#lmmt_ba <- lmertree(unsafe ~ 1 | street | age + sex + economic + crowded,
#                     data = art_gen_dat)
#length(lmmt_ba$tree)
#VarCorr(lmmt_ba)
#lmmt_ba_preds <- predict(lmmt_ba, newdata = safety_test)
#mean((lmmt_ba_preds - safety_test$unsafe)^2) ## Accuracy MSE
#mean((lmmt_ba_preds - bart_preds_test)^2) ## Fidelity MSE
```