Characterizing Effect Heterogeneity 2

Cyrus Samii

2019-06-13

Contents

1 Estimating Conditional Treatment Effects (CATES)

1.1 Setting

Some notation following Chernozhukov et al.:

- Potential outcomes, $Y(1), Y(0)$.
- Covariates X.
- Baseline conditional average (BCA): $b_0(X) = E[Y(0)|X]$.
- CATE: $s_0(X) = E[Y(1)|X] - E[Y(0)|X]$.
- CIA holds: $D \perp\!\!\!\perp (Y(1), Y(0))|X$.
- Subvector of stratifying covariates, $X_1 \subseteq X$, such that
- propensity score is $p(X) = P[D = 1|X_1]$, with $0 < p_0 \leq p(X) \leq p_1 < 1$.
- Observe $Y = DY(1) + (1 - D)Y(0)$, in which case,
- $Y = b_0(X) + Ds_0(X) + U$, with $E[U|X, D] = 0$, where
- $b_0(X) = E[Y|D = 0, X]$ and $s_0(X) = E[Y|D = 1, X] - E[Y|D = 0, X]$. CIA allows us to write b_0 and s_0 in terms of observables.
- Observe N iid draws of (Y, X, D) with law P. Draws are indexed by $i = 1, ..., N$.

A result to keep in mind (already noted in Athey and Imbens PNAS, I believe):
Define

$$H = H(D, X) = \frac{D - p(X)}{p(X)(1 - p(X))}.$$

Given the assumptions above, we have,

$$\begin{aligned} E[YH|X] &= E\left[\frac{Y(D - p(X))}{p(X)(1 - p(X))}\middle| X\right] \\ &= \frac{1}{p(X)(1 - p(X))} E\left[D(D - p(X))Y(1) + (1 - D)(D - p(X))Y(0)|X\right] \\ &= E[Y(1)|X] - E[Y(0)|X] \\ &= s_0(X). \end{aligned}$$

We can use YH as an "unbiased signal" about $s_0(X)$. It is, however, a noisy signal, and so in using this, Chernozhukov et al. make adjustments (e.g., for their second BLP method). Nonetheless, it does provide a target that one can use in trying to tune methods for estimating CATEs (see below the section on choosing an ML method).

1.2 Overview and Goals

We consider three types of high-dimensional methods for estimating heterogenous treatment effects: trees, random forests, and "elastic net" regularized regression. The focus initially will be on point predictions, rather than inference on such predictions. This is because the applications that we use work with summaries of the point predictions rather than the point predictions in and of themselves. See the section below on "features of CATES", referencing Chernozhukov et al. (2017, arxiv), for more on this point.

We are also interested in methods that work well when we entertain a high-dimensional covariate vector. I say "entertain" because it may be that, in fact, the covariates that predict heterogeneity are few, but this is something that we do not know *a priori,* and rather we have at our disposal many covariates that we want to consider as candidates for predicting effect heterogeneity. This leads us to machine learning approaches that use regularization to balance that ability to make very fine grained predictions with penalties for overfitting.

We consider the following algorithms:

- Trees
- Random Forests
- Elastic Net

1.3 Data preparation

Bringing in Data (note that we need to harmonize data wrt to section 1):

```
mex_data <- as.data.frame(import("progresa_mat.dta"))
covs <- setdiff(names(mex_data),
                c('indiv_id',
                  'treatment',
                  'enrolled',
                  'treated_adj',
                  'y_adj',
                  'ml_single_parent',
                  'group'))
```

Checking the data:

```
misCheck <- as.matrix(apply(mex_data, 2, function(x){sum(is.na(x))}))
colnames(misCheck) <- "Number missing"
kable(misCheck)
```

	Number missing
indiv_id	0
enrolled	0
ml_age	0
ml_n_child	0
ml_male	0
ml_hh_head_male	0
ml_single_parent	0
ml_al_father	0
ml_al_mother	0
ml_lw_father	0
ml_lw_mother	0
ml_literacy	0
ml_toilet	0
ml_water_piped	0
ml_lights	0
ml_tv	0
ml_car	0
ml_refrige	0
ml_n_total	0
ml_yrs_educ	0
ml_hh_head_edu	0
ml_father_educ	0
ml_mother_educ	0
ml_male_mi	0
ml_al_father_mi	0
ml_al_mother_mi	0
ml_literacy_mi	0
ml_toilet_mi	0
ml_water_piped_mi	0
ml_lights_mi	0
ml_tv_mi	0
ml_car_mi	0

	Number missing
ml_refrige_mi	0
ml_yrs_educ_mi	0
ml_father_educ_mi	0
ml_mother_educ_mi	0
treatment	0

Trees

1.4 Trees for potential outcomes

We start with using a regression tree to model potential outcomes. This helps us to understand how the trees work and the settings that we need to fix. We just use all available covariates. The `rpart()` function loads when we load the `causalTree` package. We will use 10-fold cross validation (CV) to estimated CV-error that we will minimize in selecting the complexity parameter to prune the tree. We evaluate covariate importance using the built-in evaluation function. For a classification problem such that this, a given covariate's importance is based on the sum of standardized goodness of classification values for all splits that involves it.

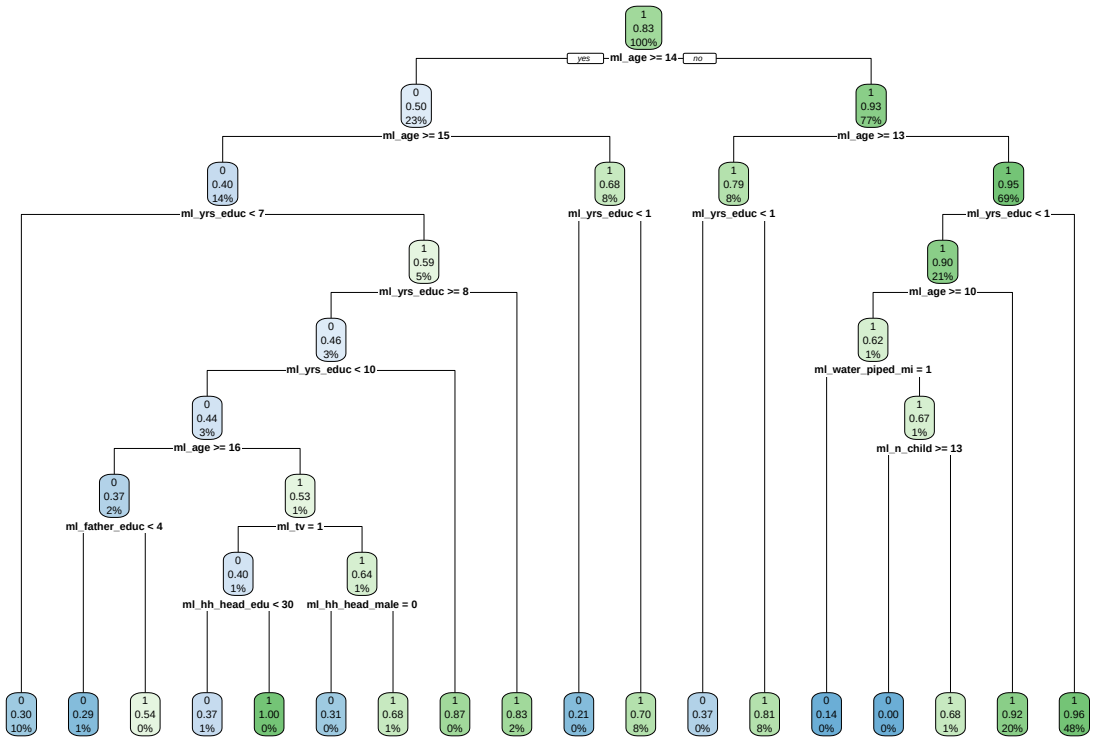
Here is the function:

```
print(rpartOp)
```

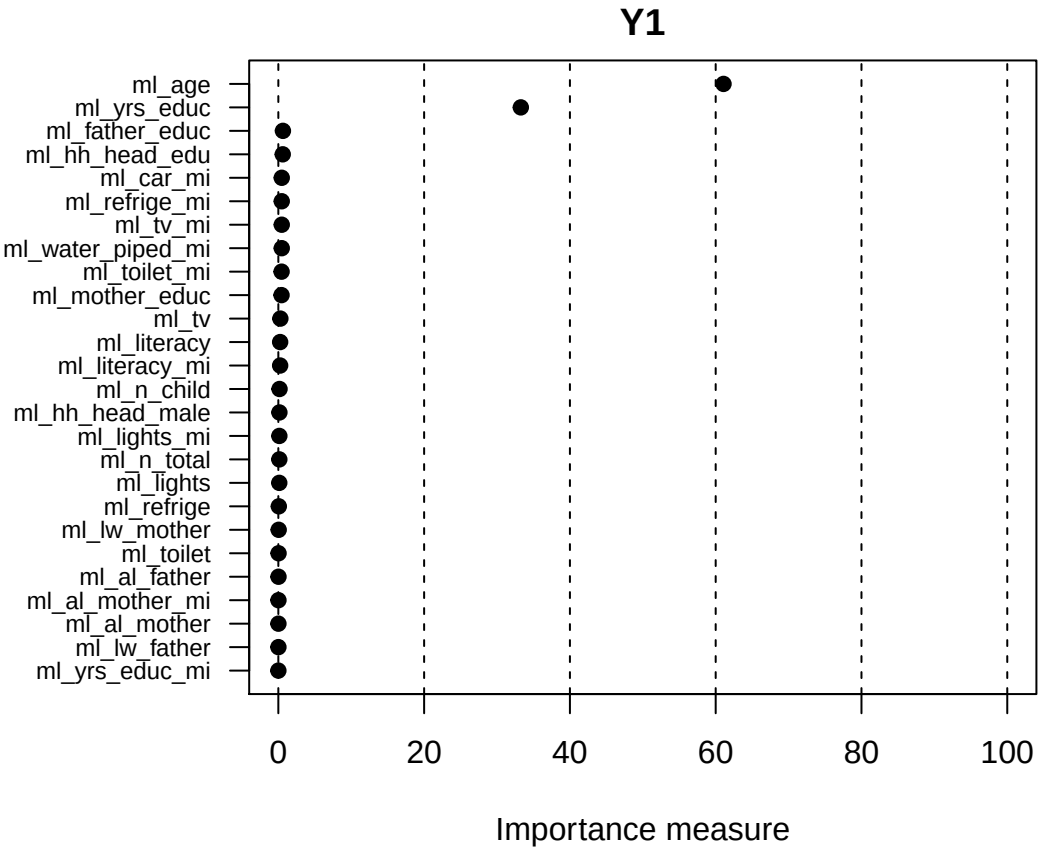
```
## function (formulaIn = NULL, methodIn = NULL, dataIn = NULL, xvalIn = NULL,
##     minbucketIn = NULL, cp_incIN = NULL, target = NULL, treatVar = NULL)
## {
##     if (target == "po") {
##         treeOut <- rpart(formulaIn, method = methodIn, data = dataIn,
##             control = rpart.control(xval = xvalIn, minbucket = minbucketIn,
##                 cp = cp_incIN), model = TRUE)
##     }
##     if (target == "fx") {
##         treeOut <- causalTree(formulaIn, data = dataIn, treatment = treatVar,
##             split.Rule = "CT", split.Honest = F, cv.option = "CT",
##             cv.Honest = F, split.alpha = 1, model = TRUE, control = rpart.control(xval = xvalIn,
##                 minbucket = minbucketIn, cp = cp_incIN))
##     }
##     opcp <- treeOut$cptable[, 1][which.min(treeOut$cptable[,
##         4])]
##     optreeOut <- prune(treeOut, opcp)
##     resList <- list(tree = optreeOut, varimp = 100 * (optreeOut$variable.importance/sum(optreeOut$va
##     return(resList)
## }
```

And now results:

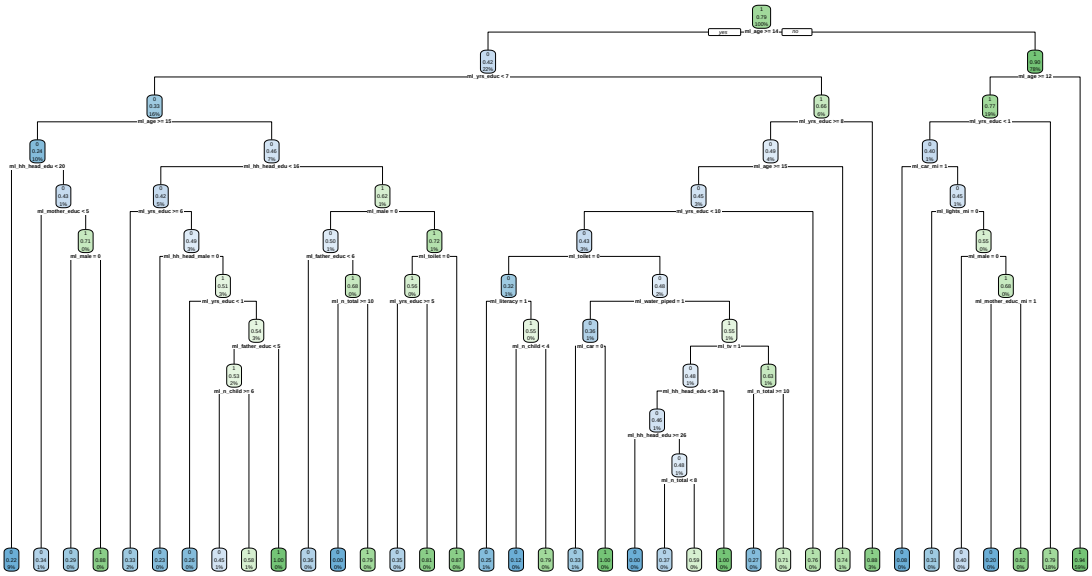
```
mexFormula <- as.formula(paste("enrolled ~", paste(covs, collapse="+")))
mex_data1 <- subset(mex_data, treatment==1)
mex_data0 <- subset(mex_data, treatment==0)
tree_y1 <- rpartOp(formulaIn=mexFormula,
                   methodIn="class",
                   dataIn=mex_data1,
                   xvalIn=10,
                   minbucketIn=2,
                   cp_incIN=0,
                   target="po")
rpart.plot(tree_y1$tree)
```



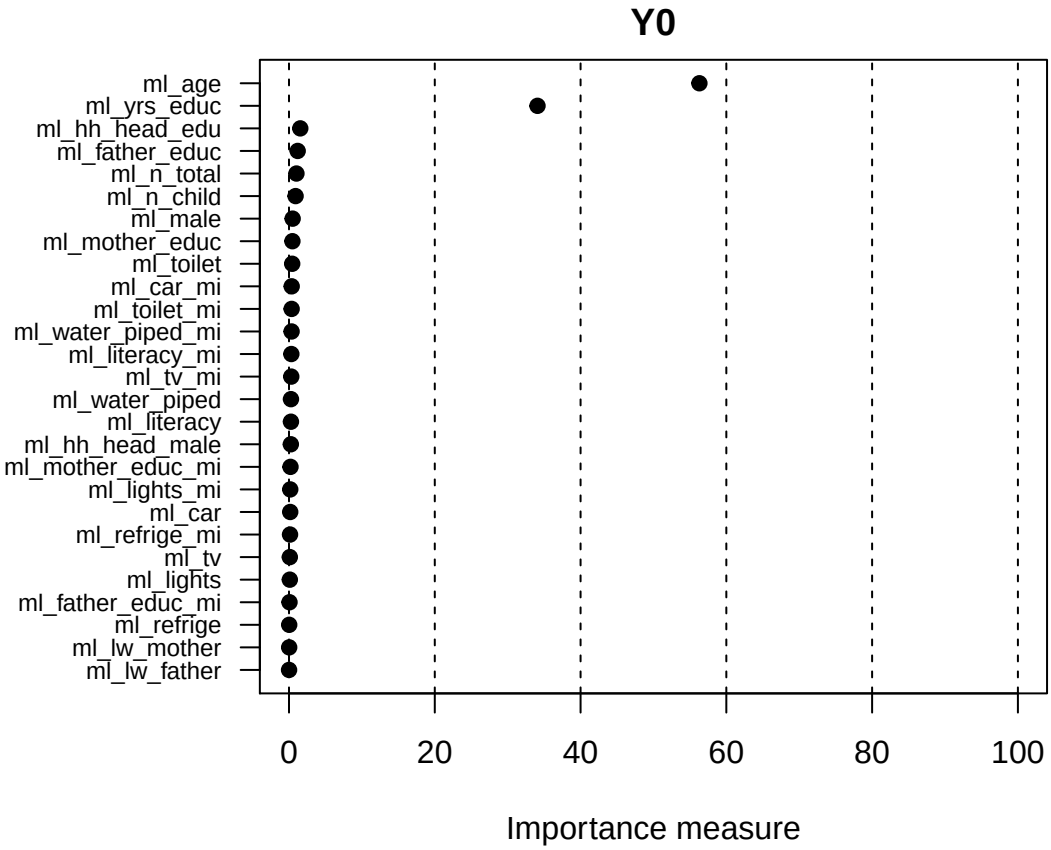

```
par(mar=c(4,10,2,2))
rpart_impplot(treeIn=tree_y1, mainIn="Y1", labelScale=0.75)
```



```
tree_y0 <- rpartOp(formulaIn=mexFormula,
                   methodIn="class",
                   dataIn=mex_data0,
                   xvalIn=10,
                   minbucketIn=2,
                   cp_incIN=0,
                   target="po")
rpart.plot(tree_y0$tree)
```



```
par(mar=c(4,10,2,2))
rpart_impplot(treeIn=tree_y0, mainIn="Y0", labelScale=0.75)
```



Looking at the results for the treated and control outcomes, we see that in both cases, the optimal trees gain predictive traction almost entirely by splitting on age and years of education. The other variables contribute to predictive accuracy only to a neglible degree.

What are the implications of a covariate's importance for predicting potential outcomes when it comes to predicting effect heterogeneity? We have

$$E[Y(1) - Y(0)|X] = E[Y(1)|X] - E[Y(0)|X]$$

by the linearity of expectations. That being the case, if $E[Y(1)|X] = E[Y(1)]$ and $E[Y(0)|X] = E[Y(0)]$, then it must be that $E[Y(1) - Y(0)|X] = E[Y(1) - Y(0)]$. As such, being a variable that is predictive of either of the potential outcomes is a *necessary* condition for a covariate for it to be predictive of effects. That said, variables that are predictive of potential outcomes may not be predictive of effects. E.g., we could have that

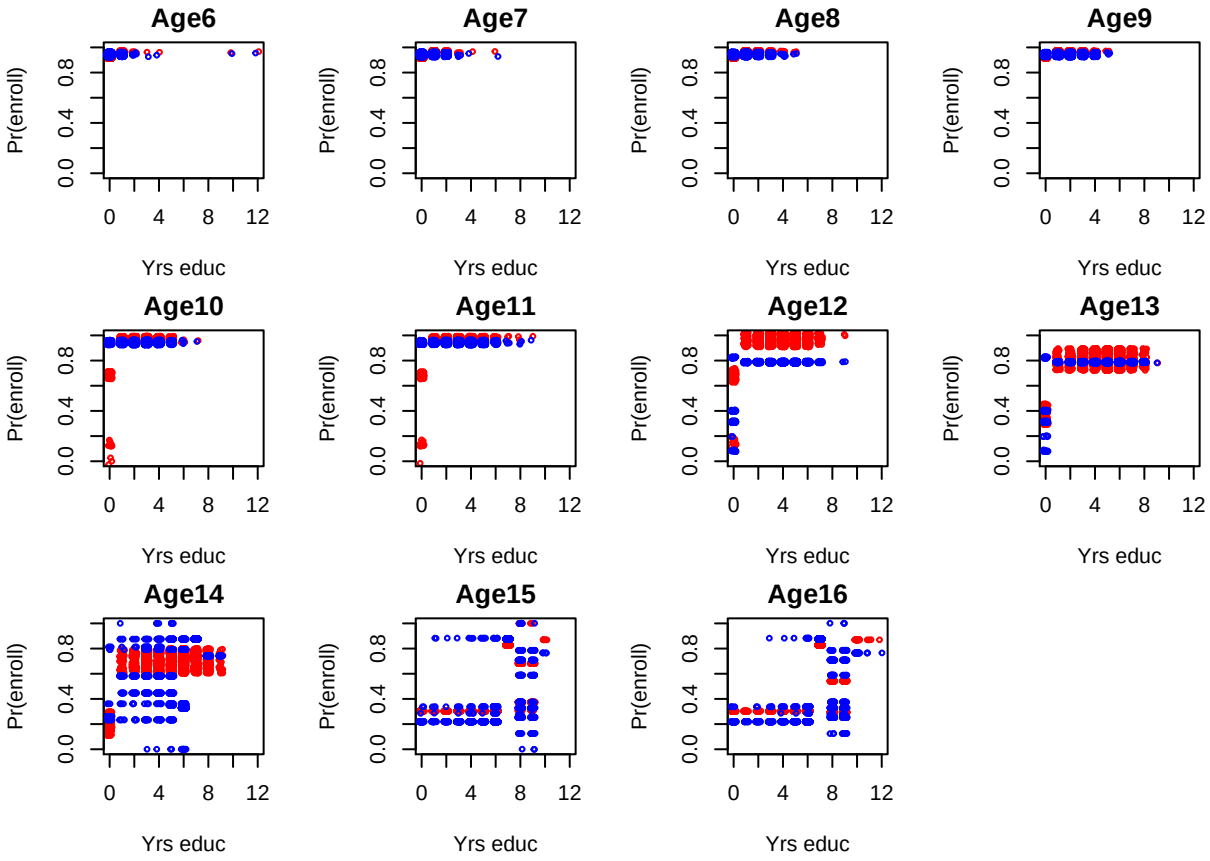
$$E[Y(d)|X] = \alpha + \beta d + \gamma X$$

for $d = 0, 1$, in which case

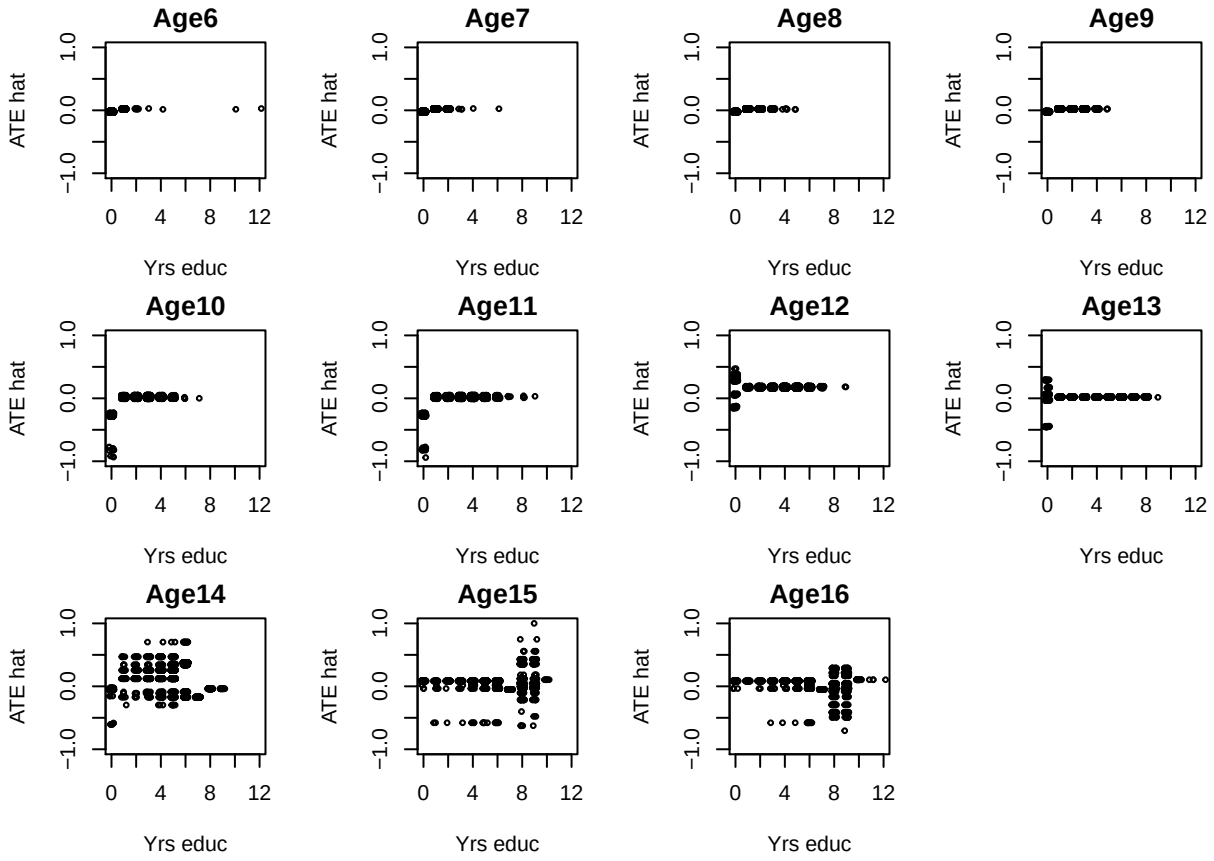
$$E[Y(1) - Y(0)|X] = (\alpha + \beta + \gamma X) - (\alpha + \gamma X) = \beta,$$

which does not depend on X. That said, a starting point for selecting variables to consider as contributors to effect heterogeneity would those that have some predictive relationship with potential outcomes. (Note that this is similar to what is done in Imai and Strauss 2011).

Let us view the predicted potential outcome trees, plotting over the two covariates that seem the matter:



From these plots we can see that for lower age groups, the predicted means are very close, and so effects are basically zero. Then, we have variation for higher age groups. We can see all of this more clearly by plotting the estimated conditional treatment effects themselves:



While this approach is useful for getting a handle on effect heterogeneity, it is not optimized with respect to treatment effects per se. We will turn to such methods next.

Trees for effect heterogeneity

We now turn to trees for effect heterogeneity. There are different ways to go about this. The first is to use Horvitz-Thompson scaling. The next to construct a tree optimized with respect to effect heterogeneity.

1.4.1 Horvitz-Thompson scaling

Recall the HT scaling result above. We can construct our unbiased signal of $s_0(X)$ as

$$\hat{S}(X_i) = Y_i \frac{(D_i - p(X_i))}{p(X_i)(1 - p(X_i))}.$$

We assume complete random assignment, in which case we have, $p(X_i) = p$. For the Progresa experiment:

```
pUp <- mean(mex_data$treatment)
pUp
```

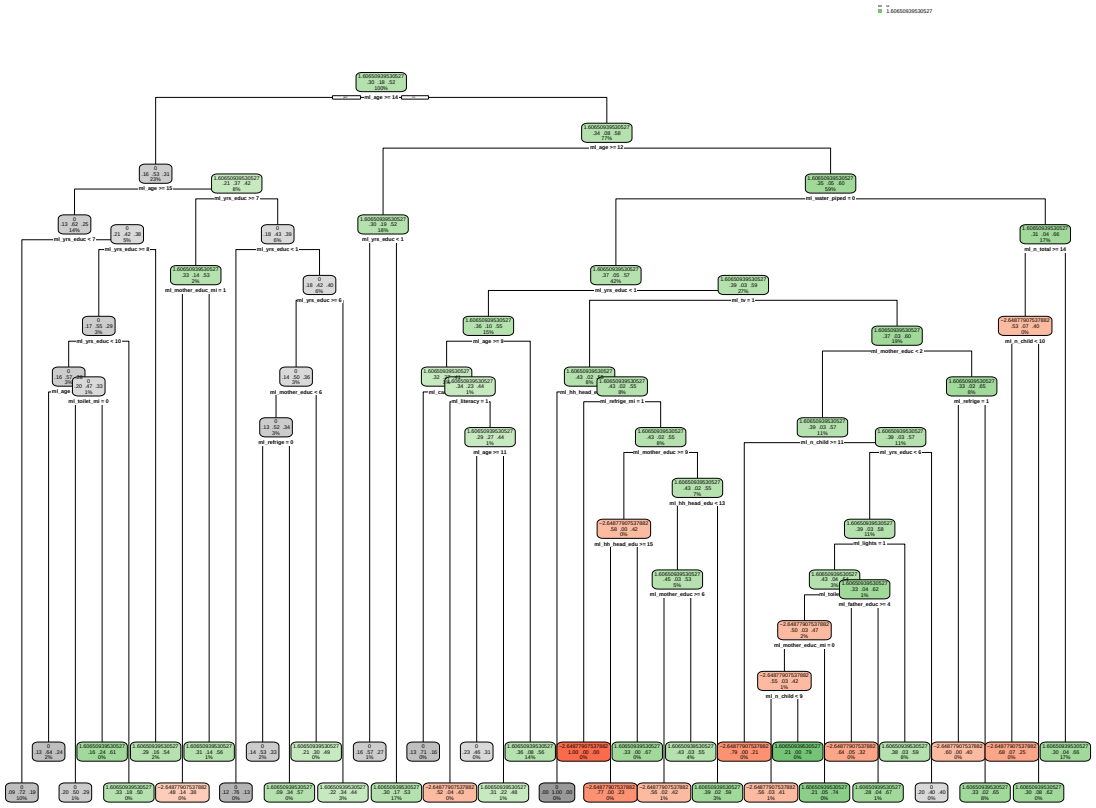
```
## [1] 0.6224676
```

```
mex_data$Senrolled <- with(mex_data, enrolled*((treatment-pUp)/(pUp*(1-pUp))))
```

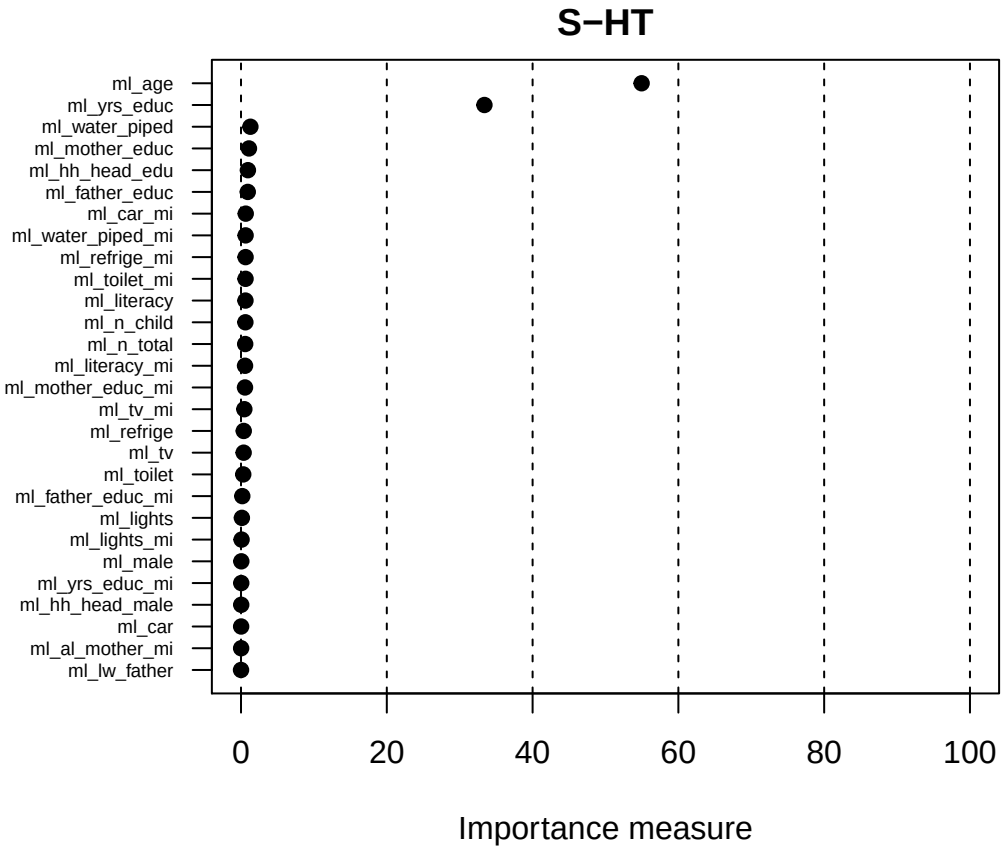
Then we fit the tree to this:

```
mexFormula_S <- as.formula(paste("Senrolled ~", paste(covs, collapse="+")))
tree_S <- rpartOp(formulaIn=mexFormula_S,
                  methodIn="class",
                  dataIn=mex_data,
                  xvalIn=10,
                  minbucketIn=2,
                  cp_incIN=0,
                  target="po")
rpart.plot(tree_S$tree)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



```
par(mar=c(4,10,2,2))
rpart_impplot(treeIn=tree_S, mainIn="S-HT", labelScale=0.6)
```



Pretty much the same story as what we had seen before.

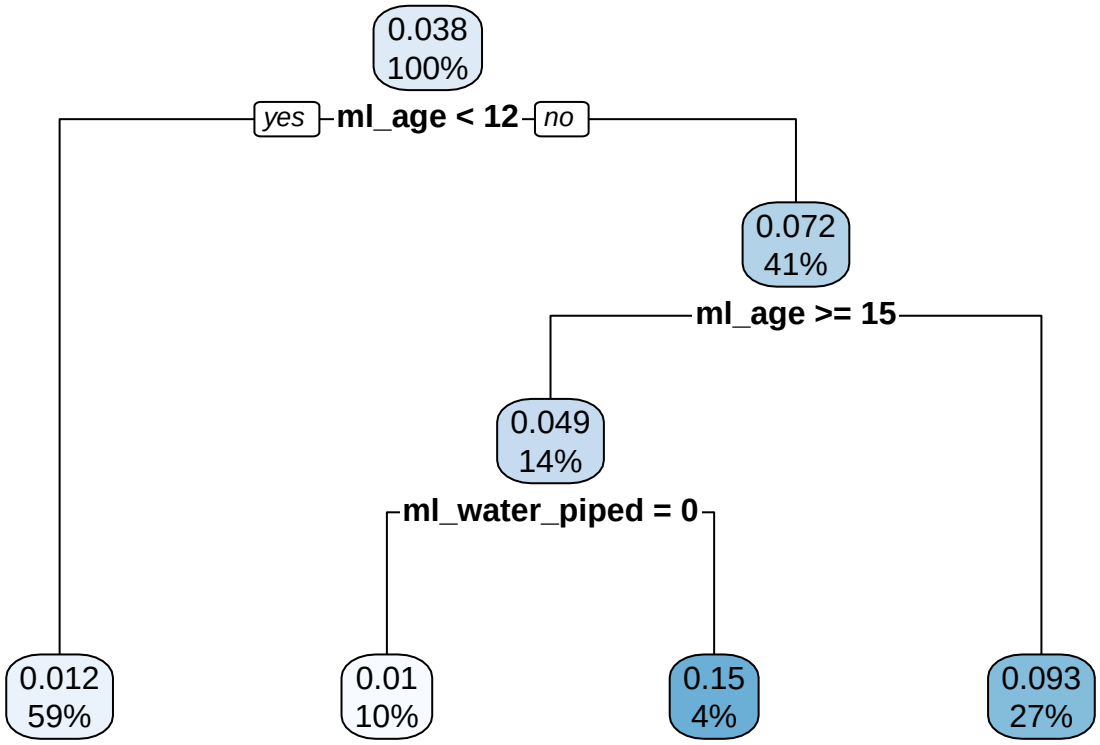
1.4.2 Causal Tree

We now turn to the "causal tree," as per Athey and Imbens. We use their recommended causal tree (CT) splitting rule. We do not use their honest splitting or CV rules because we are not interested in in-sample inference.

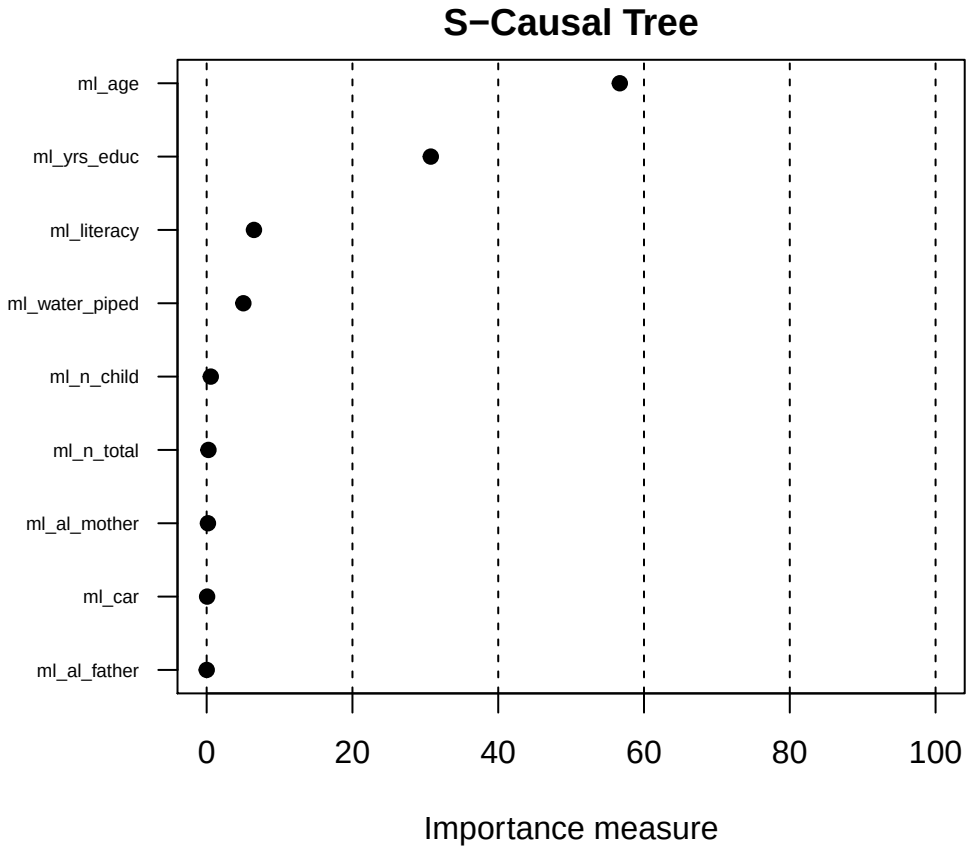
```
tree_fx_tree <- causalTree(mexFormula,
                      data=mex_data,
                      treatment=mex_data$treatment,
                      split.Rule="CT",
                      split.Honest=F,
                      cv.option="CT",
                      cv.Honest=F,
                      split.alpha=1,
                      control=rpart.control(xval=10,
                                            minbucket=2,
                                            cp=0.00005))
```

```
## [1] 2
## [1] "CT"
```

```
opcp_fx <- tree_fx_tree$cptable[,1][which.min(tree_fx_tree$cptable[,4])]
optreeOut_fx <- prune(tree_fx_tree, opcp_fx)
rpart.plot(optreeOut_fx)
```




```
tree_fx <- list(tree = optreeOut_fx,
                varimp = 100*(optreeOut_fx$variable.importance/sum(optreeOut_fx$variable.importance)))
par(mar=c(4,10,2,2))
rpart_impplot(treeIn=tree_fx, mainIn="S-Causal Tree", labelScale=0.6)
```



Similar story as above.

Note that the CV-optimal tree and the variable importance ranking don't quite match in that the latter puts some weight on literacy. This is because the variable importance measure looks at all trees where it is included in a split. But presumably what it offers is redundant relative to something else that is included in the optimal tree.