Note on testing for heterogeneity with grf

Alex Coppock and Molly Offer-Westort

2025-06-19

This note summarizes an email exchange between the two of us on how to use the `grf` package to assess treatment effect heterogeneity that we thought could be useful more broadly. In short, Alex's naive application of a "simple" test for heterogeneity wildly over-rejected the null, so he asked Molly for help, who fixed the problem.

The setting is a two-arm randomized experiment with many covariates. We're going to use the `grf` package to fit a generalized random forest model, which will generate predictions of the conditional average treatment effect for every unique covariate profile in the dataset. A relatively straightforward test against the null hypothesis of effect homogeneity (recommended here: https://grf-labs.github.io/grf/articles/diagnostics.html) is a T-test that compares the average value of the above median estimates to the below median estimate.

Alex's code below follows the procedures and code described in Athey and Wager (2019):

> A first, simple approach to testing for heterogeneity involves grouping observations according to whether their out-of-bag CATE estimates are above or below the median CATE estimate, and then estimating average treatment effects in these two subgroups separately using the doubly robust approach (8).

Although the authors note some limitations to this approach:

> This procedure is somewhat heuristic, as the "high" and "low" subgroups are not independent of the scores $\hat{\Gamma}_i$ used to estimate the within-group effects; however, the subgroup definition does not directly depend on the outcomes or treatments (Y_i, W_i) themselves, and it appears that this approach can provide at least qualitative insights about the strength of heterogeneity.

Here's a simulation (written in `DeclareDesign`) in which the outcome is correlated with the covariates, but the treatment effect is constant for all units.

```
library(grf)
library(tidyverse)
library(DeclareDesign)

# this part of the design will stay the same
N <- 1000
design_stub <-
  declare_model(
    N = N,
    X1 = rbinom(N, 1, 0.5),
    X2 = rbinom(N, 1, 0.5),
    X3 = rbinom(N, 1, 0.5),
    X4 = rbinom(N, 1, 0.5),
```

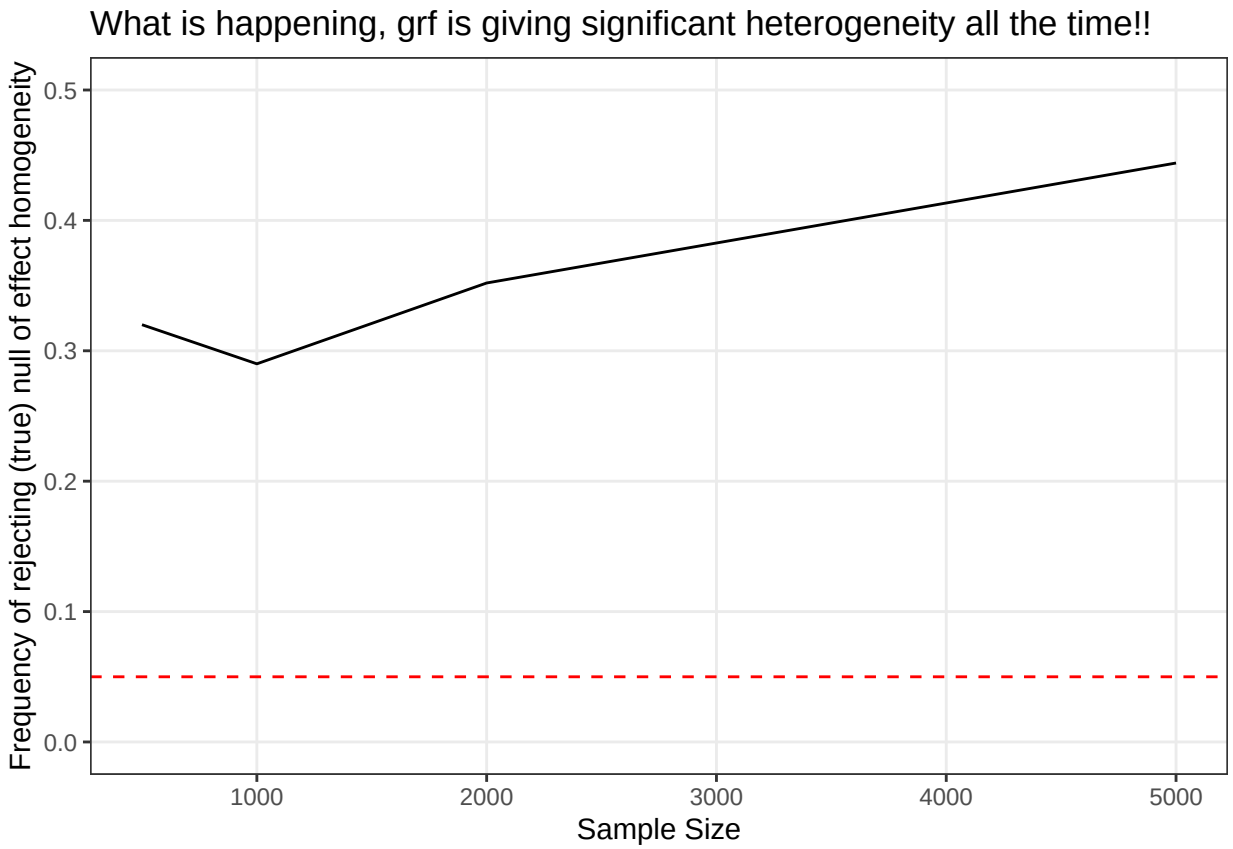
```
    X5 = rbinom(N, 1, 0.5),
    U = rnorm(N),
    potential_outcomes(
      Y ~  0.1 * X1 + 0.2 * X2 + 0.3 * X3 + 0.4 * X4 + 0.5 * X5 + U
    )
  ) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z))
```

Here is Alex's naive approach to implementing the test:

```
grf_tester <- function(data) {
  fit <- with(data, causal_forest(
    X = cbind(X1, X2, X3, X4, X5),
    Y = Y,
    W = Z,
    W.hat = 0.5
  ))
  preds <- predict(fit)$predictions
  high.effect <- preds > median(preds)
  ate.high <- average_treatment_effect(fit, subset = high.effect)
  ate.low <- average_treatment_effect(fit, subset = !high.effect)
  dif <- ate.high[1] - ate.low[1]
  dif.se <- sqrt(ate.high[2]^2 + ate.low[2]^2)

  tibble(p.value = 2 * (1 - pnorm(abs(dif / dif.se))))
}

design_1 <- design_stub +
  declare_estimator(handler = grf_tester)
```

When we simulate, we get a figure that shows serious over-rejection of the true null hypothesis that actually gets worse as sample size grows.

```
simulations_1 <-
  design_1 |>
  redesign(N = c(500, 1000, 2000, 5000)) |>
  simulate_designs(sims = 500)
```



Molly's solution

Molly wrote back:

I thought this was going to be easier, but the problem is not trivial!

Given that there is no relationship of the treatment with response, the CATEs should be centered around zero. While the estimates are honest, if you condition post estimation on the honest estimates that are above median, these estimates will more frequently be statistically distinguishable from estimates that are below median. What you need to diagnose is if the model is applied to a hold-out sample, if those differences will still hold.

However, even doing a cross-fitting approach where I fit the models on one fold, used that model to predict ranks on another, and then used the model from the first fold to get doubly robust scores for estimation in the second fold, I was still getting inflated rejection. I think because the means model used for ranking contaminates the DR scores used for estimation, even though they have a residual bias correction.

I ended up with a procedure where you need to make sure the data + model used for ranking is completely separate from the data + model used for estimation.

```
grf_tester_improved <-
  function(data) {

    # Create folds
    folds <- sample(rep(1:2, length.out = nrow(data)))
    idxA <- which(folds == 1)
    idxB <- which(folds == 2)
```

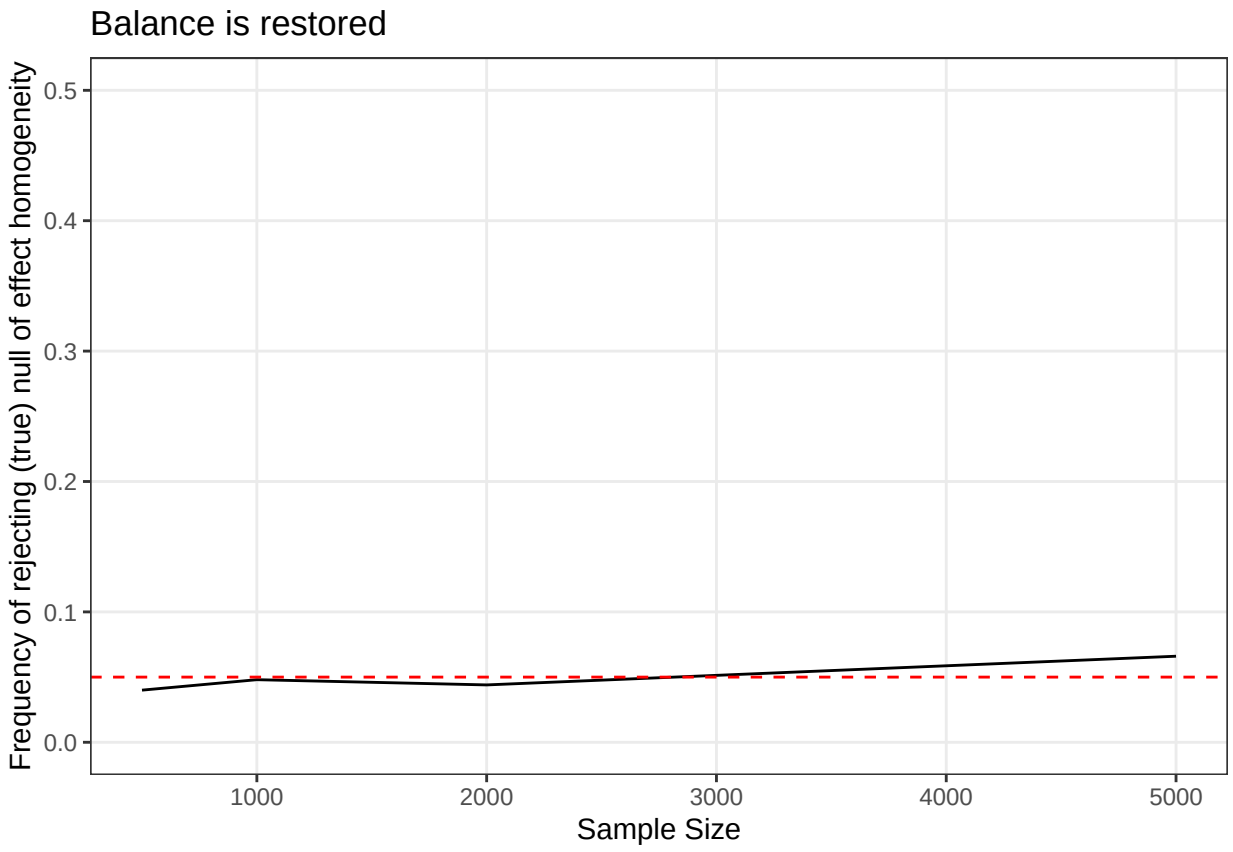
```r
dataA <- data[idxA, ]
dataB <- data[idxB, ]

# Train on fold A
fitA <- with(
  dataA,
  causal_forest(
    X = cbind(X1, X2, X3, X4, X5),
    Y = Y,
    W = Z,
    W.hat = 0.5
  )
)

# Train on fold B
fitB <- with(
  dataB,
  causal_forest(
    X = cbind(X1, X2, X3, X4, X5),
    Y = Y,
    W = Z,
    W.hat = 0.5
  )
)

# Using model B, predict on fold A for subgroups
predsA <- predict(
  fitB,
  newdata = with(dataA, cbind(X1, X2, X3, X4, X5))
)$predictions
high.effectA <- predsA > median(predsA)

# Using model A, predict on fold B for subgroups
predsB <- predict(
  fitA,
  newdata = with(dataB, cbind(X1, X2, X3, X4, X5))
)$predictions
high.effectB <- predsB > median(predsB)

# Estimate treatment effects using within-fold models
ate.highA <- average_treatment_effect(fitA, subset = high.effectA)
ate.lowA <- average_treatment_effect(fitA, subset = !high.effectA)
difA <- ate.highA[1] - ate.lowA[1]
dif.seA <- sqrt(ate.highA[2]^2 + ate.lowA[2]^2)

ate.highB <- average_treatment_effect(fitB, subset = high.effectB)
ate.lowB <- average_treatment_effect(fitB, subset = !high.effectB)
difB <- ate.highB[1] - ate.lowB[1]
dif.seB <- sqrt(ate.highB[2]^2 + ate.lowB[2]^2)

tibble(
  p.valueA = 2*(1 - pnorm(abs(difA / dif.seA))),
  p.valueB = 2*(1 - pnorm(abs(difB / dif.seB)))
```

```
    )
  }
```

Now we combine that function with the design stub, re-simulate, and it works as described!

```
design_2 <- design_stub +
  declare_estimator(handler = grf_tester_improved)
```

```
simulations_2 <-
  design_2 |>
  redesign(N = c(500, 1000, 2000, 5000)) |>
  simulate_designs(sims = 500)
```



Athey, Susan, and Stefan Wager. 2019. "Estimating Treatment Effects with Causal Forests: An Application." *Observational Studies* 5 (2): 37–51.