Evaluating the Benefits of Sample Splitting for Double Machine Learning

Michael Sarkis, Jack McCarthy

4/24/2022

Double ML

In this section we will replicate the methods put forth in the paper by Chernozhukov et al., dubbed "CCD-DHNR", which establishes an unbiased Bayesian machine learning framework for treatment effect estimation.

Simulated Dataset

First we will generated a toy dataset on which we may test each method. We will assume the data is generated according to the following simpler model derived from the general form considered in CCDDHNR:

$$
\begin{aligned}
Y_i &= D_i\theta + g_0(X_i) + \epsilon_i, \quad && \epsilon_i \sim N(0,1),\\
D_i &= m_0(X_i) + \tau_i, && \tau_i \sim N(0,1),
\end{aligned}
$$

where Y is the outcome, D is the treatment, and X is the vector of covariates. $g_0(X)$ and $m_0(X)$ relate the covariates to the value of the response and the treatment respectively. We define these "nuisance" functions to be

$$
\begin{aligned}
g_0(x) &= x_1 + \sigma(x_3),\\
m_0(x) &= x_3 + \sigma(x_1),
\end{aligned}
$$

where $\sigma(x)$ is the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

We also define $X_i \sim N(0, \Sigma)$ where $\Sigma_{ij} = 0.5^{|i-j|}$

Assuming a continuous response and treatment, N total observations, a p-dimensional covariate vector, and a true value of $\theta = 0.5$, we may generate a toy dataset as follows:

```
# data dimensions
N = 250
p = 100
theta = 0.5

# covariance matrix
i_mat <- matrix(rep(1:p, p), p)
```

```r
j_mat <- matrix(rep(1:p, each=p), p)
Sigma <- 0.5^abs(i_mat - j_mat)

generate_data <- function(N, p, theta, S) {
  # generate covariates
  X <- mvrnorm(n=N, mu=rep(0, p), Sigma=Sigma)

  # generate treatment
  D <- m_0(X) + rnorm(N)

  # generate response
  Y <- D*theta + g_0(X) + rnorm(N)

  return(tibble(X=X, D=D, Y=Y))
}
```

where we will use the **generate_data** function to obtain distributions on the predicted value of θ through multiple fits to many random datasets.

Naive ML

A naive approach to estimating θ would be to estimate $D\theta + g_0(X)$ using some machine learning method. In line with the demonstration in Chernozhukov et al., we will split the samples into two index sets of equal size, R (auxiliary) and S (primary). We will then use the auxiliary set generate the estimate $D\hat{\theta} + \hat{g}_0(X)$, and use the primary set to estimate θ as

$$\hat{\theta} = \left(\frac{1}{n}\sum_{i\in S} D_i^2\right)^{-1} \frac{1}{n}\sum_{i\in S} D_i(Y_i - \hat{g}_0(X_i)).$$

We do so for our simulated dataset below using random forest regressors.