Machine Learning
and
Applied Econometrics

An Application: Double Machine Learning for Price Elasticity

Double Machine Learning for Price Elasticity of Demand Function

- This presentation is in part based on:
  - Alexandre Belloni, Victor Chernozhukov, and Christian Hansen, High-Dimensional Methods and Inference on Structural and Treatment Effects, Journal of Economic Perspectives 28:2 (29-50), Spring 2014.
  - Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins, Double/Debiased Machine Learning for Treatment and Structural Parameters, Econometrics Journal 21:1, 2018.

Structural and Treatment Effects

- The Model

$$Y = f(D, Z) + u, \quad E(u \mid Z, D) = 0$$

$$D = h(Z) + v, \quad E(v \mid Z) = 0$$

  - D is the target variable of interest (e.g., price) or the treatment variable (typically, D=0 or 1)
  - Z is the set of exogenous covariates or control variables (instruments, confounders), may be high-dimensional.

- Partial Linear Model: $f(D, Z) = D\theta + g(Z)$

Structural and Treatment Effects

- If D is numeric structural variable

$$\theta = \partial y / \partial D$$

- If D=1 or 0
  - Average Treatment Effect (ATE)

$$\theta = E\left[f(1,Z) - f(0,Z) \right]$$

  - Average Treatment Effect for the Treated (ATT)

$$\theta = E\left[f(1,Z) - f(0,Z) \mid D = 1 \right]$$

Structural and Treatment Effects

- Based on Partial Linear Model,
  - Frisch-Waugh-Lovel Theorem: $\hat{\theta} = \tilde{\theta}$

$$\hat{u} = Y - \hat{\theta} D - \hat{g}(Z)$$

$$\left.\begin{aligned} \tilde{u} &= Y - \tilde{g}(Z) \\ \tilde{v} &= D - \tilde{h}(Z) \end{aligned}\right\} \quad \tilde{u} = \tilde{\theta}\tilde{v} \quad \textit{if } g \textit{ and } h \textit{ are linear}$$

  - Machine Learning : $\tilde{g}(Z)$ *and* $\tilde{h}(Z)$
  - OLS: $\tilde{\theta} = \tilde{v}'\tilde{u} / \tilde{v}'\tilde{v}$
    - This estimate is biased and inefficient!
  - De-biased: $\breve{\theta} = \tilde{v}'\tilde{u} / \tilde{v}'D, \quad \textit{in general } \breve{\theta} \neq \tilde{\theta}$

Structural and Treatment Effects

- Based on Partial Linear Model,
  - Sample Splitting
    - {1,…,N} = Set of all observations
    - I_1 = main sample = set of observation numbers, of size n, is used to estimate θ; e.g., n=N/2.
    - I_2 = auxilliary sample = set of observations, of size πn = N −n, is used to estimate g;
    - I_1 and I_2 form a random partition of the set {1,...,N}
  - Cross Fitting on $\{I_1,I_2\}$ and $\{I_2,I_1\}$

Structural and Treatment Effects

- Cross Fitting on $\{I_1, I_2\}$ and $\{I_2, I_1\}$
  - Machine Learning:

$$\tilde{g}_1(Z)\ and\ \tilde{h}_1(Z)\ on\ (I_1, I_2)$$

$$\tilde{g}_2(Z)\ and\ \tilde{h}_2(Z)\ on\ (I_2, I_1)$$

  - De-Biased Estimator:

$$\left.\begin{array}{l} \breve{\theta}_2 = \breve{\theta}(I_1, I_2) \\ \breve{\theta}_1 = \breve{\theta}(I_2, I_1) \end{array}\right\} \quad \breve{\theta} = \frac{\breve{\theta}_1 + \breve{\theta}_2}{2}$$

  - $\breve{\theta}\ is\ \sqrt{N}$ consistent and approximately centered normal (Chernozhukov, et.al., 2017)

Structural and Treatment Effects

- Extensions
  - Based on sample splitting {1,…,N} = {I_1,I_2}, de-biased estimator may be obtained from pooled data and ML residuals:

    $$\breve{\theta} = \begin{bmatrix} \tilde{v}_1 & \tilde{v}_2 \end{bmatrix}' \begin{bmatrix} \tilde{u}_1 & \tilde{u}_2 \end{bmatrix} / \begin{bmatrix} \tilde{v}_1 & \tilde{v}_2 \end{bmatrix}' \begin{bmatrix} D_1 & D_2 \end{bmatrix}$$

  - Cross fitting can be k-fold, e.g. k=2, 5, 10

Example: Table Wine Sales in Vancouver BC

- Total Weekly Sales of Imported and Domestic Table Wine in Vancouver, BC, Canada from week ending April 4, 2009 to week ending May 28, 2011 (372,228 sales)
  - Irregularly-spaced time series
  - Data Source: American Association of Wine Economists

Example: Table Wine Sales in Vancouver BC

- 372,228 observations of 17 variables in an Excel spreadsheet:
  - SKU #, Product Long Name, Store Category Major Name, Store Category Sub Name, Store Category Minor Name, Current Display Price, Bottled Location Code, Bottle Location Desc, Domestic/Import Indicator, VQA Indicator, Product Sweetness Code, Product Sweetness Desc, Alcohol Percent, Julian Week No, Week Ending Date, Total Weekly Selling Unit, Total Weekly Volume Litre

Demand Function for Table Wine in Vancouver BC

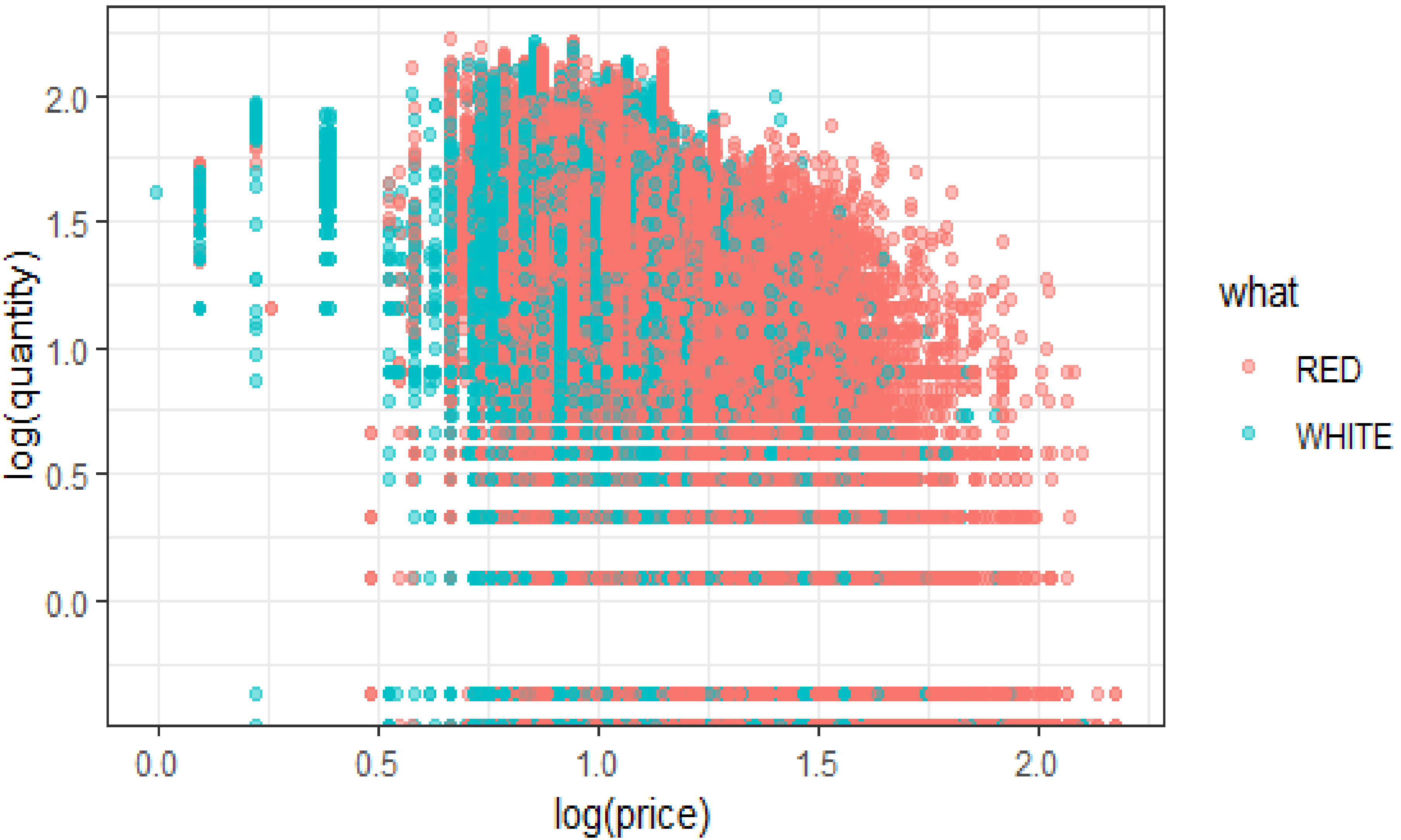

Table Wine Sales in Vancouver BC

Double Machine Learning of Price Elasticity

$$Y = \theta D + g(Z) + u, \quad E(u \mid Z, D) = 0$$

$$D = m(Z) + v, \quad E(v \mid Z) = 0$$

- Y = log of quantity (total weekly selling unit in bottles)
- D = log of price (current display price in Canadian $)
- Z = { What = Store Category Minor Name (Red/White), Where = Store Category Sub Name (Countries), Loc = Bottled Location Code, Alc = Alcohol Percent, Age = Julian Week No, …}
- θ = Price Elasticity

Table Wine Sales in Vancouver BC

Double Machine Learning of Price Elasticity

- GLM (Lasso)

K-fold CF	Y (Val. MSE)	D (Val.MSE)	θ (Price Elas.)
2	2.126	0.320	-1.238
5	2.126	0.320	-1.238
10	2.126	0.320	-1.238

- GLM (Elastic Net)

K-fold CF	Y (Val. MSE)	D (Val. MSE)	θ (Price Elas.)
2	2.129	0.321	-1.228
5	2.127	0.321	-1.232
10	2.127	0.320	-1.233

Table Wine Sales in Vancouver BC
Double Machine Learning of Price Elasticity

- DL (20,20)

K-fold CF	Y (Val. MSE)	D (Val.MSE)	θ (Price Elas.)
2	1.977	0.273	-1.261
5	1.984	0.273	-1.271
10	1.983	0.274	-1.131

- DL (20,10,5)

K-fold CF	Y (Val. MSE)	D (Val. MSE)	θ (Price Elas.)
2	1.966	0.273	-1.279
5	1.982	0.274	-1.124
10	1.973	0.273	-1.245

Table Wine Sales in Vancouver BC

Double Machine Learning of Price Elasticity

- DRF (50 trees, max depth=20)

K-fold CF	Y (Val. MSE)	D (Val.MSE)	θ (Price Elas.)
2	2.126	0.320	-1.129
5	2.130	0.318	-1.135
10	2.129	0.318	-1.136

- GBM (50 trees, max depth=5)

K-fold CF	Y (Val. MSE)	D (Val. MSE)	θ (Price Elas.)
2	1.943	0.266	-1.192
5	1.944	0.266	-1.192
10	1.941	0.265	-1.193

Table Wine Sales in Vancouver BC

Double Machine Learning of Price Elasticity

- Conclusion
  - Linear regression model may not explain and validate this set of data. Thus, the price elasticity estimate of 1.23 may not be reliable.
  - The nonparametric Deep Learning Neural Networks and Gradient Boosting Machine perform better in learning this dataset.
  - Gradient Boosting Machine as applied to a partial linear model framework in price elasticity is 1.19.
  - All computations are done with R package H2O:
    - Darren Cook, Practical Machine Learning with H2O, O'Reilly Media, Inc., 2017.