CESS: Heterogeneity and Machine Learning

Nuffield College Oxford

Ray Duch & Tom Robinson

July 18, 2019

Nuffield College Oxford Centre for Experimental Social Science - CESS

The Workshop

- Part I: Heterogeneous Treatment Effects
- Part II: Machine Learning
- Part III: Machine learning, heterogeneity and experimental measurement error

The Workshop

- Part 1: Heterogeneous Treatment Effects
  - Conditional Average Treatment Effects (CATEs)
  - Regression estimation of CATE
  - Illustration case study
- Part II: Machine Learning
  - How Machine Learning Works
  - Pitfalls and drawbacks of Machine Learning
  - Illustration case study
- Part III: Heterogeneity and Experimental Measurement Error
  - Multi-modes and micro-replications
  - Identifying heterogeneous mode effects
  - Assessing experimental measurement error

Part I: Heterogeneous Treatment Effects

CATE: Potential Outcome Subgroup

- Sometimes useful to refer to potential outcomes for a subset of the subjects
- Expressions of the form $Y_i(d)|X = x$ denote potential outcomes when the condition $X = x$ holds
- For example, $Y_i(0)|d_i = 1$ refers to the untreated potential outcome for a subject who actually receives the treatment

CATE Estimated Using Regression

$$
\begin{aligned}
Y_i &= Y_i(0)(1 - D_i) + Y_i(1)D_i \\
\beta_{Di} &= Y_i(0) - Y_i(1) \\
\text{ATE} &= E(Y_i(0) - Y_i(1)) = E(\beta_{D_i}) \\
Y_i &= \beta_0 + \beta_{D_i} + \epsilon_i \\
&= \beta_0 + \beta_{D_i} + [(\beta_{D_i} - \beta_D)D_i + \epsilon_i] \\
\beta_D &= E(\beta_{D_i} | D_i = 1) \\
\beta_D &= E(Y(0))
\end{aligned} \tag{1}
$$

CATE Estimated Using Regression

$$Y_i = \beta_0 + \beta_D D_i + \epsilon_i$$

$$\begin{aligned} Y_i &= \beta_0 + \beta_{D_i} D_i + \epsilon_i \\ &= \beta_0 + \beta_X X_i + (\beta_D + \beta_{DX} X_i) D_i + \epsilon_i \end{aligned} \tag{2}$$

Part II: Machine learning

How Machine Learning Works

- Prediction: produce predictions of y from x
  - Supervised – use subset of “known” y values to train model e.g. LASSO, random forest, BART etc.
  - Unsupervised – self-organised learning e.g. k-means clustering

- Discovers complex structure not specified in advance

- Fits complex and very flexible functional forms to the data
  - without simply overfitting; and
  - functions that work well out-of-sample

Performance of different machines

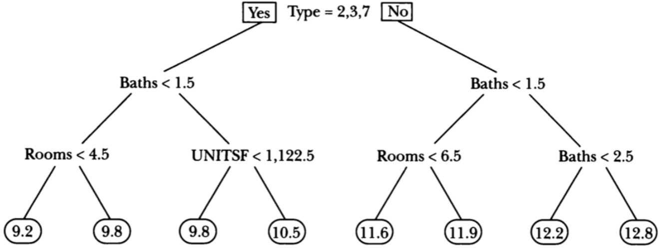
Table 1

Performance of Different Algorithms in Predicting House Values

	Prediction performance (R^2)		Relative improvement over ordinary least squares by quintile of house value				
Method	*Training sample*	*Hold-out sample*	1st	2nd	3rd	4th	5th
Ordinary least squares	47.3%	41.7% [39.7%, 43.7%]	–	–	–	–	–
Regression tree tuned by depth	39.6%	34.5% [32.6%, 36.5%]	−11.5%	10.8%	6.4%	−14.6%	−31.8%
LASSO	46.0%	43.3% [41.5%, 45.2%]	1.3%	11.9%	13.1%	10.1%	−1.9%
Random forest	85.1%	45.5% [43.6%, 47.5%]	3.5%	23.6%	27.0%	17.8%	−0.5%
Ensemble	80.4%	45.9% [44.0%, 47.9%]	4.5%	16.0%	17.9%	14.2%	7.6%

Regression tree

A Shallow Regression Tree Predicting House Values



Note: Based on a sample from the 2011 American Housing Survey metropolitan survey. House-value predictions are in log dollars.

Improving ML-estimation: ensemble methods

Bagging:

- "Bootstrap aggregation", random samples with replacement as training data
- Reduces variance, as used in *random forest* methods

Boosting:

- Sequential weak-learner models, data weighted by misclassification
- Reduces bias, as used in *gradient tree* methods

Stacking:

- Run different estimation strategies and weight super-learner by each model's predictive capacity
- Increases predictive capacity, see Grimmer et al. (2017)

Predicting counterfactual outcomes in experimental contexts

Suppose we have 8 observations of an outcome, treatment assignment and two covariates:

y	d	Gender	Education
12	1	Female	High
13	1	Female	Low
5	0	Female	High
6	0	Female	Low
7	1	Male	High
8	1	Male	Low
7	0	Male	High
6	0	Male	Low

Table 1: Observed

y	d	Gender	Education
?	0	Female	High
?	0	Female	Low
?	1	Female	High
?	1	Female	Low
?	0	Male	High
?	0	Male	Low
?	1	Male	High
?	1	Male	Low

Table 2: Unobserved counterfactual

$$\text{ATE}_{\text{Observed}} = 10 - 6 = 4$$

Random Forest Estimation

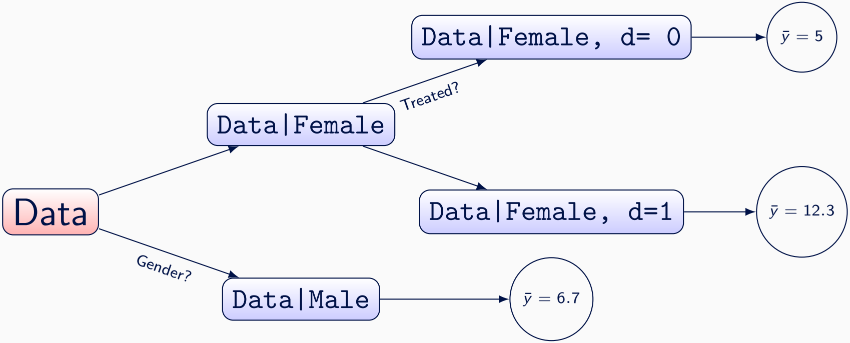
- Random sample of data *and* predictors
- Take bootstrap samples with replacement of training data:

y	d	Gender	Education
12	1	Female	High
13	1	Female	Low
5	0	Female	High
5	0	Female	High
12	1	Female	High
7	0	Male	High
7	0	Male	High
6	0	Male	Low

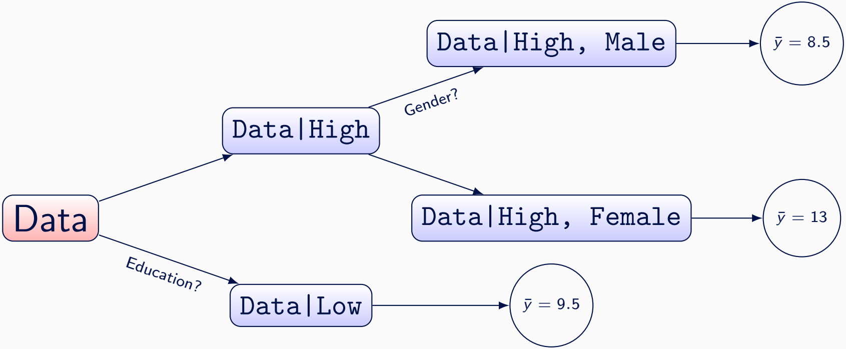
Table 3: Sampled observations (with replacement)

- Construct tree from sample, using random selection of predictor variables

Tree-based logic – Tree #1



Tree-based logic – Tree #2



Random Forest: Estimating the CATE

$$
\left(\underbrace{\begin{array}{cc}
\mathbf{\hat{y}_{i,d=1,t=1}} & \mathbf{\hat{y}_{i,d=0,t=1}} \\
12.3 & 5 \\
12.3 & 5 \\
12.3 & 5 \\
12.3 & 5 \\
6.7 & 6.7 \\
6.7 & 6.7 \\
6.7 & 6.7 \\
6.7 & 6.7
\end{array}}_{\text{Tree } \#1}\right)
\left(\begin{array}{c}
\overbrace{\mathbf{\hat{y}_{i,t=2}}}^{\text{Tree } \#2} \\
13 \\
9.5 \\
13 \\
9.5 \\
8.5 \\
9.5 \\
8.5 \\
9.5
\end{array}\right)
=
\left(\underbrace{\begin{array}{cc|c}
\hat{y}_{i,d=1} & \hat{y}_{i,d=0} & \textbf{CATE} \\
12.7 & 9 & 3.7 \\
10.9 & 7.3 & 3.6 \\
12.7 & 9. & 3.7 \\
10.9 & 7.3 & 3.6 \\
7.6 & 7.6 & 0 \\
8.1 & 8.1 & 0 \\
7.6 & 7.6 & 0 \\
8.1 & 8.1 & 0
\end{array}}_{\text{Average over trees}}\right)
$$

Treatments $d \in \{0,1\}$, trees $t \in \{0,1\}$, no. of trees $= T$

Predicted outcome given treatment assignment d

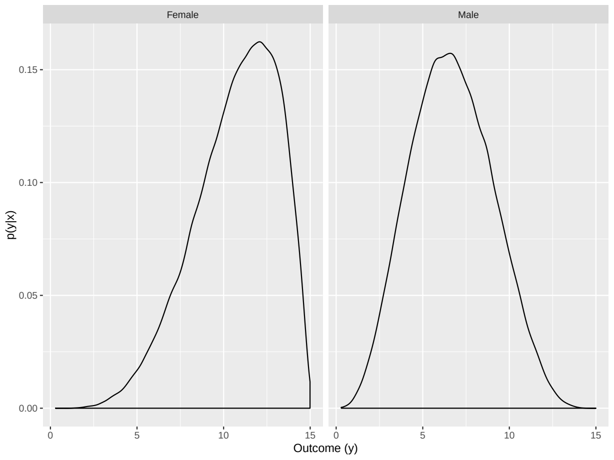
$= \hat{y}_{i,d} = \frac{1}{T}\sum_t^T \hat{y}_{i,d}$

BART Estimation strategy

- Estimate $f(x) = E(Y|x)$
- Fit a *sequence* of "weak" tree-based regression models
- Each tree contributes a "a small and different portion of f" (Chipman et al 2010)[1]
- Iterative application of sum-of-trees effectively generates a posterior probability distribution of outcomes, given covariate vector X
- From which you can recover $E(Y|x)$ and uncertainty intervals

[1] *BART: Bayesian Additive Regression Trees*, The Annals of Applied Statistics, 2010, Vo.4, No.1

Altered posterior probabilities given covariate values



$$\hat{y} = \frac{1}{K}\sum_{k=1}^{K} k \leftarrow f(x)$$

Estimating the CATE - overall strategy

- BART model estimation generates posterior function of $f(x)$
- Averaging repeat draws from posterior density generates mean outcome for each observation given its vector of predictors x_i
- x_i contains treatment assignment plus other covariates
- Predict $\hat{y}_i$ for two matrices:
  1. Actual observed treatment values (plus covariates)
  2. Counterfactual matrix of reversed treatment assignment $(1 \leftrightarrow 0)$ (plus same covariates)
- For each observation i, we recover two estimates: $y_{i,d=1}$ and $y_{i,d=0}$
- CATE $= y_{i,d=1} - y_{i,d=0}$

Estimating the CATE - generate two test matrices

- Predictions are made using two matrices[2]
- Second matrix is the test dataset in the R code
- Matrices are identical except treatment assignment is reversed in second matrix

$$
\left[\begin{array}{ccc|c}
D_{\textbf{Obs.}} & \textbf{Gender} & \textbf{Education} & y_{i,d} \\
1 & \textit{Female} & \textit{High} & 14 \\
1 & \textit{Female} & \textit{Low} & 12 \\
0 & \textit{Female} & \textit{High} & 4 \\
0 & \textit{Female} & \textit{Low} & 6 \\
1 & \textit{Male} & \textit{High} & 7 \\
1 & \textit{Male} & \textit{Low} & 7 \\
0 & \textit{Male} & \textit{High} & 8 \\
0 & \textit{Male} & \textit{Low} & 6
\end{array}\right]
\left[\begin{array}{ccc|c}
D_{\textbf{Counter.}} & \textbf{Gender} & \textbf{Education} & y_{i,d} \\
0 & \textit{Female} & \textit{High} & 7 \\
0 & \textit{Female} & \textit{Low} & 7 \\
1 & \textit{Female} & \textit{High} & 12 \\
1 & \textit{Female} & \textit{Low} & 13 \\
0 & \textit{Male} & \textit{High} & 8 \\
0 & \textit{Male} & \textit{Low} & 6 \\
1 & \textit{Male} & \textit{High} & 8 \\
1 & \textit{Male} & \textit{Low} & 6
\end{array}\right]
$$

[2] NB: The first, observed matrix is implicitly generated by BART since it is the initial training data (excluding observed outcome)

Estimating the CATE - rearrange matrices

- Matrices can be rearranged such that all observations in matrix 1 are $d = 1$ and *vice versa* for matrix 2
- Covariate information is constant across both matrices

$$\left[\begin{array}{ccc|c} D_{\textbf{Obs.}} & \textbf{Gender} & \textbf{Education} & y_{i,d=1} \\ 1 & \textit{Female} & \textit{High} & 14 \\ 1 & \textit{Female} & \textit{Low} & 12 \\ 1 & \textit{Female} & \textit{High} & 12 \\ 1 & \textit{Female} & \textit{Low} & 13 \\ 1 & \textit{Male} & \textit{High} & 7 \\ 1 & \textit{Male} & \textit{Low} & 7 \\ 1 & \textit{Male} & \textit{High} & 8 \\ 1 & \textit{Male} & \textit{Low} & 6 \end{array}\right] \left[\begin{array}{ccc|c} D_{\textbf{Counter.}} & \textbf{Gender} & \textbf{Education} & y_{i,d=0} \\ 0 & \textit{Female} & \textit{High} & 7 \\ 0 & \textit{Female} & \textit{Low} & 7 \\ 0 & \textit{Female} & \textit{High} & 4 \\ 0 & \textit{Female} & \textit{Low} & 6 \\ 0 & \textit{Male} & \textit{High} & 8 \\ 0 & \textit{Male} & \textit{Low} & 6 \\ 0 & \textit{Male} & \textit{High} & 8 \\ 0 & \textit{Male} & \textit{Low} & 6 \end{array}\right]$$

Estimating the CATE - recover CATE

- CATE $= \hat{y}_{i,d=1} - \hat{y}_{i,d=0}$
- To check for treatment heterogeneity, append covariate information since this is constant across two matrices[3]

$$\begin{pmatrix} \mathbf{\hat{y}_{i,d=1}} \\ 14 \\ 12 \\ \color{red}{12} \\ \color{red}{13} \\ 7 \\ 7 \\ \color{red}{6} \\ \color{red}{7} \end{pmatrix} - \begin{pmatrix} \mathbf{\hat{y}_{i,d=0}} \\ \color{red}{7} \\ \color{red}{7} \\ 4 \\ 6 \\ \color{red}{8} \\ \color{red}{6} \\ 8 \\ 6 \end{pmatrix} = \left(\begin{array}{c|cc} \textbf{CATE} & \textbf{Gender} & \textbf{Education} \\ 7 & \textit{Female} & \textit{High} \\ 5 & \textit{Female} & \textit{Low} \\ 8 & \textit{Female} & \textit{High} \\ 7 & \textit{Female} & \textit{Low} \\ -1 & \textit{Male} & \textit{High} \\ 1 & \textit{Male} & \textit{Low} \\ -2 & \textit{Male} & \textit{High} \\ 1 & \textit{Male} & \textit{Low} \end{array}\right)$$

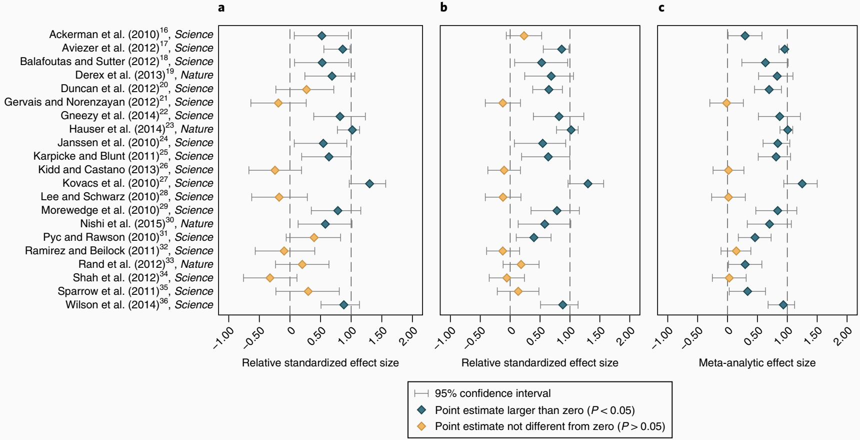
[3] NB: all observations are predicted from posterior draws; red numbers indicate predictions using counterfactual treatment assignment

Part III: Machine learning, heterogeneity and experimental measurement error

Data Generation

- Costs declining significantly
- Convenience samples are the norm
- Proliferation of data generation modes
- Democratic

There are Costs: Camerer et al 2018 Nature



Some Observations

- How do you know you have this experimental measurement error?
- You typically have no clue as to whether its an issue
- Note: this has nothing to do with external validity/representative sample/etc.

Micro-replications can help

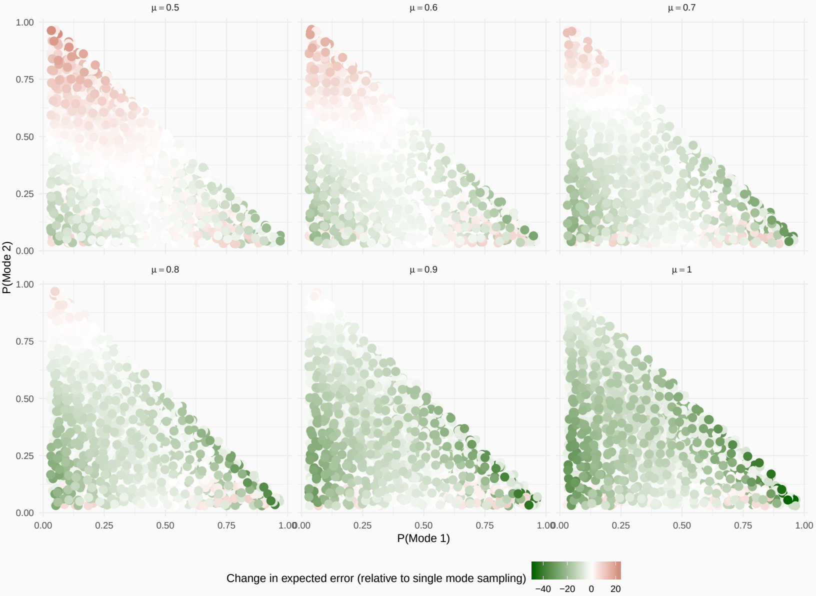
- Maybe....
- But what micro-replication?
- In which micro-replication should you invest your research dollars?
- Multi- rather than single-mode replications are more informative of experimental measurement error

The Experimental Mode or Context

Modes and Experimental Measurement Error

- do modes exaggerate measurement error, i.e., $ME_k > 0$
- resulting in $ATE_k^* = (ATE_T + ME_k)$
- multi-mode replication design may be informative when:
  - $ME_k \neq ME_{k'}$ and
  - there is a reasonably high probability the researcher can distinguish low from high error modes

Multiple-mode Replication Simulation



Illustrate: Lying Experiment (Duch Laroze Zakharov 2018)

- Outcome of interest: Lying about income from RET
- Treatment: Deduction rate that make it more expensive to lie
- Expectation: Lying declines if deduction rates rise

Lying Experiment Design (Duch Laroze Zakharov 2018

- 3 different tax rates (10%, 20% and 30%)
- Fixed at the group level
- Taxes are redistributed equally among group members
- Public good
- No excludability
- No social gains/losses
- No audits or fines
- 10 rounds
- Paid for one of them at random
- Fixed groups of 4 participants
- Random matching at the beginning

Design: each round

- RET: solve as many additions as possible in 60 sec
- two random two-digit numbers
- Information individual gross profit (before tax)
- Declare their income (to be taxed)
- Information individual net profit (after tax and redistribution)
- Differentiated by profit, tax and redistribution

Lying Experiments

CESS Oxford Lab	CESS Oxford Online	CESS Online UK	Mturk USA
CESS Oxford Subject Pool	CESS Oxford Subject Pool	CESS Online US Subject Pool	U.S. Crowd-sourced workers
1600 decisions/116 subjects	1367 Decisions/144 Subjects	696 Decisions/90 Subjects	2419 Decisions/390 Subjects
57% Rate of Lying	54% Rate of Lying	37% Rate of Lying	40% Rate of Lying

Conventional GLM Estimation

	Mode			
	Lab	Online Lab	Online UK	Mturk
Ability Rank	-0.500^{***}	-0.163^{***}	-0.163^{**}	-0.120^{***}
	(0.036)	(0.045)	(0.071)	(0.037)
20% Deduction	-0.123^{***}			
	(0.024)			
30% Deduction	-0.128^{***}	-0.184^{***}	0.042	0.018
	(0.025)	(0.025)	(0.038)	(0.021)
No Audit	-0.334^{***}	-0.127^{***}	-0.155^{***}	0.011
	(0.023)	(0.026)	(0.036)	(0.024)
Age	0.012^{***}	0.007^{**}	-0.0002	0.002^{**}
	(0.002)	(0.003)	(0.001)	(0.001)
Gender	0.002	0.100^{***}	-0.022	-0.004
	(0.022)	(0.025)	(0.035)	(0.020)
Constant	0.715^{***}	0.476^{***}	0.880^{***}	0.576^{***}
	(0.066)	(0.089)	(0.070)	(0.043)

BART Estimation

- Bayesian estimation strategy using tree-logic
- Highly flexible estimation strategy

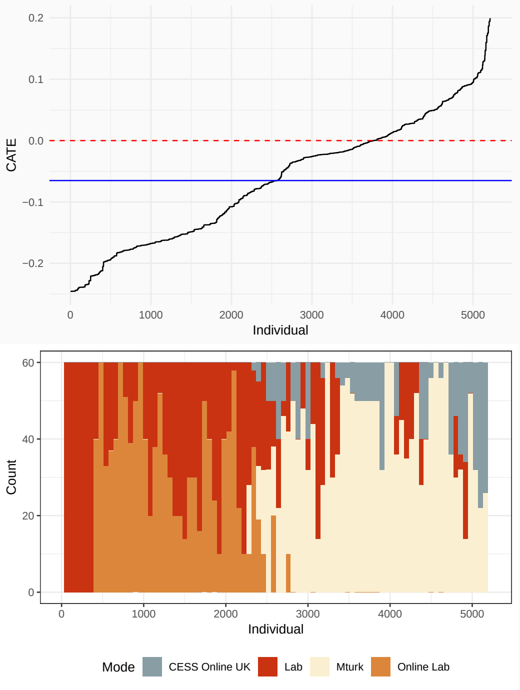
To recover individual estimates of treatment effect:

- Assume binary treatment
- Run BART on experimental data (the training set) to generate both model and predicted outcomes for observed data
- Invert treatment assignment of all observations, and pass through model (test set) to generate set of counterfactual predictions
- For each individual, i, $CATE = Y_{i,D=1} - Y_{i,D=0}$

BART: R Code

```
# Separate outcome and training data
y <- df$report.rate
train <- df[,-1]

# Gen. test data where those treated become untreated, for use in calculating ITT
test <- train
test$treat.het <- ifelse(test$treat.het == 1,0,ifelse(test$treat.het == 0,1,NA))

# Run BART for predicted values of observed and synthetic observations
bart.out <- bart(x.train = train, y.train = y, x.test = test)

# Recover CATE estimates and format into dataframe
CATE <- c(bart.out$yhat.train.mean[train$treat.het == 1] - bart.out$yhat.test.mean[test$treat.het == 0],
          bart.out$yhat.test.mean[test$treat.het == 1] - bart.out$yhat.train.mean[train$treat.het == 0])

CATE_df <- data.frame(CATE = CATE)
covars <- rbind(train[train$treat.het == 1,c(2:5)], test[test$treat.het==1,c(2:5)])

CATE_df <- cbind(CATE_df,covars)
CATE_df <- CATE_df[order(CATE_df$CATE),]
CATE_df$id <- c(1:length(CATE))
```

All replication code available at `https://github.com/rayduch/Experimental-Modes-and-Heterogeneity`


CATE
0.2
0.1
0.0
−0.1
−0.2
0
1000
2000
3000
4000
5000
Individual
Count
60
40
20
0
0
1000
2000
3000
4000
5000
Individual
Mode
CESS Online UK
Lab
Mturk
Online Lab

R Code

Ensemble methods: stacking

Stacking is a straightforward ensemble method and especially useful for the $\hat{y}$ problems raised by Mullanaithan and Spiess (2017).

- Let M be a vector of estimation strategies
- $D_{train} = \{y_{train}, X_{train}\}$, $D_{test} = \{X_{test}\}$
- Learner models h_m: $\hat{y}_m$ for each $m \in M$, using D_{train}
- Super-learner model H: $y_{train} = \hat{y}_1 + ... + \hat{y}_m$
- Predict $\hat{y}_{test}$ using H and D_{test}

Meta-regression model H provides a weighting over individual classifiers based on their individual predictive capacity!

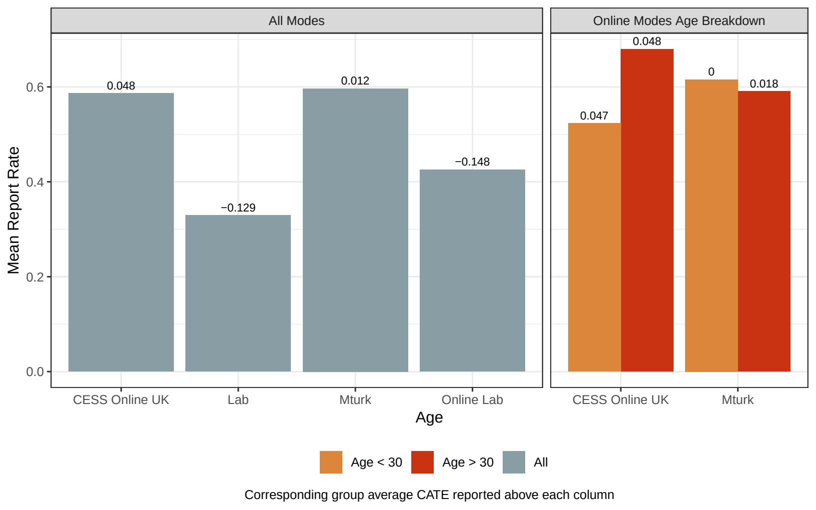
R Code

Measurement Error?

Real Effort Tasks Inter-Class Correlations Across Modes

Mode	(1)	(2)	(3)	(4)
Lab	0.768	0.768	0.636	0.85
	(0.018)	(0.018)	(0.039)	(0.047)
Lab Online	0.807	0.76	0.762	0.767
	(0.018)	(0.017)	(0.021)	(0.047)
Online UK	0.88	0.827	0.827	0.752
	(0.011)	(0.018)	(0.026)	(0.029)
MTurk	0.758	0.758	0.782	0.828
	(0.015)	(0.012)	(0.024)	(0.026)
Deduction Rate	10%	30%	10%	30%
Audited?	No	No	Yes	Yes

Comparing Percentages of Actual Earnings Reported



India Measurement Error Experiments

Coeff	S.E.	t-statistic	p	Mode	Error	Incentivised?
-0.74	0.47	-1.57	0.12	MTurk	Control	No
-0.83	0.47	-1.76	0.08	MTurk	High	No
-3.85	0.51	-7.52	0.00	CESS Online	Control	No
-3.23	0.49	-6.64	0.00	CESS Online	High	No
-1.16	0.49	-2.35	0.02	MTurk	Control	Yes
-1.00	0.33	-3.01	0.00	MTurk	High	Yes

Table 5: Induced measurement error model results