Mean-Weighted Case Specific Random Forests (MWCSRF) for Estimating Causal Effects

Linus Addae & Michael J. Higgins
laddae@ksu.edu mikehiggins@ksu.edu

Kansas State University

Aug. 6, 2020

Setting

- Suppose we have an observational study where each unit is either assigned to "treatment" or "control." There are n_t treated units and n_c control units.
- Consider the setting where there are many more control units than treated units $n_c >> n_t$.
- Covariates for treated units may comprise a small subspace of the entire covariate space.
- We are estimating the effect of treatment for the treated units.

Examples

Common settings:

- Testing effect of an experimental medical procedure.
  Treated patients: Small clinical trial.
  Control units: Large medical database.

Examples

Common settings:

- Testing effect of an experimental medical procedure.
  Treated patients: Small clinical trial.
  Control units: Large medical database.
- (Lalonde) Effect of National Supported Work (NSW) Demonstration on earnings.
  Treated units: Participants in the program.
  Control units: Responses from a national survey.

Examples

Common settings:

- Testing effect of an experimental medical procedure.
  Treated patients: Small clinical trial.
  Control units: Large medical database.
- (Lalonde) Effect of National Supported Work (NSW) Demonstration on earnings.
  Treated units: Participants in the program.
  Control units: Responses from a national survey.
- Common trait: Controls contain some people that are eligible and may benefit from the treatment. Many other control units are ineligible or inappropriate for the treatment condition.

Examples

Common settings:

- Testing effect of an experimental medical procedure.
  Treated patients: Small clinical trial.
  Control units: Large medical database.
- (Lalonde) Effect of National Supported Work (NSW) Demonstration on earnings.
  Treated units: Participants in the program.
  Control units: Responses from a national survey.
- Common trait: Controls contain some people that are eligible and may benefit from the treatment. Many other control units are ineligible or inappropriate for the treatment condition.
- Ideally, identification of "appropriate" control units to compare against treated units is automated based on measured covariates.

Machine learning for causal inference

- Recent surge in machine learning methods designed to simultaneously identify appropriate control units and perform estimation of treatment effects.

Machine learning for causal inference

- Recent surge in machine learning methods designed to simultaneously identify appropriate control units and perform estimation of treatment effects.
- Random forest methods have been developed to estimate causal effects, most notably, heterogeneous treatment effects (Causal Random Forests by Wager, Athey, and others).

Machine learning for causal inference

- Recent surge in machine learning methods designed to simultaneously identify appropriate control units and perform estimation of treatment effects.
- Random forest methods have been developed to estimate causal effects, most notably, heterogeneous treatment effects (Causal Random Forests by Wager, Athey, and others).
- However, recent methods—in particular, Case-Specific Random Forests (CSRF) by Xu and Nettleton—have shown improvement to standard random forests when finding predictions for a small subset of units.

Machine learning for causal inference

- Recent surge in machine learning methods designed to simultaneously identify appropriate control units and perform estimation of treatment effects.
- Random forest methods have been developed to estimate causal effects, most notably, heterogeneous treatment effects (Causal Random Forests by Wager, Athey, and others).
- However, recent methods—in particular, Case-Specific Random Forests (CSRF) by Xu and Nettleton—have shown improvement to standard random forests when finding predictions for a small subset of units.
- Goal: Extend CSRF to estimate causal quantities, especially in the case where $n_T << n_C$.

Model of response

We assume the Neyman-Rubin Causal Model of response:

$$Y_i = y_{i1} T_i + y_{i0}(1 - T_i)$$

- Y_i: Observed response of ith unit.
- y_{i1}, y_{i0}: Potential outcome of the unit under treatment/control.
- T_i: Treatment indicator for unit i. $T_i = 1$ if the unit receives treatment, $T_i = 0$ otherwise.

Model of response

We assume the Neyman-Rubin Causal Model of response:

$$Y_i = y_{i1} T_i + y_{i0}(1 - T_i)$$

- Y_i: Observed response of ith unit.
- y_{i1}, y_{i0}: Potential outcome of the unit under treatment/control.
- T_i: Treatment indicator for unit i. $T_i = 1$ if the unit receives treatment, $T_i = 0$ otherwise.
- SUTVA: Response of i only depends on treatment status of i.

Model of response

We assume the Neyman-Rubin Causal Model of response:

$$Y_i = y_{i1} T_i + y_{i0}(1 - T_i)$$

- Y_i: Observed response of ith unit.
- y_{i1}, y_{i0}: Potential outcome of the unit under treatment/control.
- T_i: Treatment indicator for unit i. $T_i = 1$ if the unit receives treatment, $T_i = 0$ otherwise.
- SUTVA: Response of i only depends on treatment status of i.
- Quantity of interest: Average treatment effect for the treated (ATT)

$$ATT = E(Y_{\iota 1} - Y_{\iota 0} | T_\iota = 1)$$

Random Forest

- Random forest (Breiman, 2001): Nonparametric machine learning technique for classification and regression prediction problems.
- Data split into training and test dataset.

  $\mathbf{D_{tr}} = \{(X_i, Y_i), i = 1, 2, \ldots, N_{tr}\}$ with N_{tr} instances

  $\mathbf{D_{te}} = \{(X_i, Y_i), i = N_{tr} + 1, \ldots, N_{tr} + N_{te}\}$ with N_{te} instances.

  Y_i: Response. X_i: p-dimensional covariate vector.

Random Forest

- Random forest (Breiman, 2001): Nonparametric machine learning technique for classification and regression prediction problems.
- Data split into training and test dataset.

  $\mathbf{D_{tr}} = \{(X_i, Y_i), i = 1, 2, \ldots, N_{tr}\}$ with N_{tr} instances

  $\mathbf{D_{te}} = \{(X_i, Y_i), i = N_{tr} + 1, \ldots, N_{tr} + N_{te}\}$ with N_{te} instances.

  Y_i: Response. X_i: p-dimensional covariate vector.
- Draw B independent and uniform bootstrap samples with replacement $\mathbf{D}^* = \{D_i^* = i = 1, 2, \ldots N^*\}$ of size N_{tr} from the training data set.

Random Forest

- Grow regression tree for each bootstrap sample B_i. Each unit begins with a single initial node.

Random Forest

- Grow regression tree for each bootstrap sample B_i. Each unit begins with a single initial node.
- Recursively select a subset of covariates and split each node into two subnodes K_1, K_2 based on the best covariate in the subset.
- Make split to minimize the sum of squares of the error (SSE):

$$\sum_{b=1}^{2} \sum_{i \in K_b} (Y_i^* - \overline{Y_b})^2,$$

where $\overline{Y_b}$ is the mean response of the training observations within the subnode K_b.

Random Forest

- Grow regression tree for each bootstrap sample B_i. Each unit begins with a single initial node.
- Recursively select a subset of covariates and split each node into two subnodes K_1, K_2 based on the best covariate in the subset.
- Make split to minimize the sum of squares of the error (SSE):

$$\sum_{b=1}^{2} \sum_{i \in K_b} (Y_i^* - \overline{Y_b})^2,$$

  where $\overline{Y_b}$ is the mean response of the training observations within the subnode K_b.
- Continue splitting until nodes are of sufficiently small size, last nodes in tree are called "terminal nodes."

Random Forest

- For each tree and each observation (Y_0, X_0), estimate response $\hat{Y}_0(X_0)$ by finding terminal node that contains X_0 and averaging the responses Y_i of all units in the training set that are contained in that terminal node.

Random Forest

- For each tree and each observation (Y_0, X_0), estimate response $\hat{Y}_0(X_0)$ by finding terminal node that contains X_0 and averaging the responses Y_i of all units in the training set that are contained in that terminal node.
- Aggregate trees across bootstrap samples to obtain random forest estimates.
- For a given datapoint (Y_0, X_0), the random forest estimate of Y_0, denoted $\hat{Y}_0(X_0)$ is

$$\hat{Y}_0(X_0) = \frac{1}{B} \sum_{j=1}^{B} \hat{Y}_{j0}(X_0)$$

Case-Specific Random Forests

- Case-Specific Random Forests: For a given (Y_0, X_0), weight bootstrap samples to make units (Y_i, X_i) in training set $\mathbf{D}_{tr}$ that are "closer to" (Y_0, X_0) more likely to be sampled.

Case-Specific Random Forests

- Case-Specific Random Forests: For a given (Y_0, X_0), weight bootstrap samples to make units (Y_i, X_i) in training set $\mathbf{D}_{tr}$ that are "closer to" (Y_0, X_0) more likely to be sampled.
- Perform initial bagging: Construct B^* decision trees, record how many terminal nodes containing X_0 also contain unit X_i. Denote this number as E_i.

Case-Specific Random Forests

- Case-Specific Random Forests: For a given (Y_0, X_0), weight bootstrap samples to make units (Y_i, X_i) in training set $\mathbf{D}_{tr}$ that are "closer to" (Y_0, X_0) more likely to be sampled.
- Perform initial bagging: Construct B^* decision trees, record how many terminal nodes containing X_0 also contain unit X_i. Denote this number as E_i.
- Now, when performing a regression forest, draw bootstrap samples so that the probability that unit (Y_i, X_i) is sampled is proportional to E_i.
- Each (Y_0, X_0) has its own bootstrap weights.

Mean-Weight Case-Specific Random Forest (MWCSRF)

- Mean-Weight Case-Specific Random Forests (MWCSRF): Reweight bootstrap samples, but leverage information from units contained within the same area of the covariate space.

Mean-Weight Case-Specific Random Forest (MWCSRF)

- Mean-Weight Case-Specific Random Forests (MWCSRF): Reweight bootstrap samples, but leverage information from units contained within the same area of the covariate space.
- Perform initial bagging step: For each (Y_i, X_i) in the training set $\mathbf{D_{tr}}$ and each (Y_j, X_j) in the test set $\mathbf{D_{te}}$, compute number of trees in which they share a terminal node $E_{j,i}$.

Mean-Weight Case-Specific Random Forest (MWCSRF)

- Mean-Weight Case-Specific Random Forests (MWCSRF): Reweight bootstrap samples, but leverage information from units contained within the same area of the covariate space.
- Perform initial bagging step: For each (Y_i, X_i) in the training set $\mathbf{D_{tr}}$ and each (Y_j, X_j) in the test set $\mathbf{D_{te}}$, compute number of trees in which they share a terminal node $E_{j,i}$.
- Perform random forest, but now units i in bootstrapped samples are selected with probability proportional to

$$\overline{E}_i = \sum_{j \in \mathbf{D_{tr}}} E_{j,i}.$$

- The MWCSRF reduces to CSRF when $|\mathbf{D_{te}}| = 1$.

ATT Estimation using MWCSRF

- Run MWCSRF with training units $\mathbf{D_{tr}}$ equal to all control units and test set $\mathbf{D_{te}}$ equal to all treated units.

ATT Estimation using MWCSRF

- Run MWCSRF with training units $\mathbf{D_{tr}}$ equal to all control units and test set $\mathbf{D_{te}}$ equal to all treated units.
- Define the prognostic score $\phi(X_i) = E(Y_i|X_i, T_i = 0)$ (Hansen).
- Use MWCSRF results to get prediction of the prognostic score for all treated units $\hat{\phi}(X_i)|\, T_i = 1$.

ATT Estimation using MWCSRF

- Run MWCSRF with training units $\mathbf{D_{tr}}$ equal to all control units and test set $\mathbf{D_{te}}$ equal to all treated units.
- Define the prognostic score $\phi(X_i) = E(Y_i|X_i, T_i = 0)$ (Hansen).
- Use MWCSRF results to get prediction of the prognostic score for all treated units $\hat{\phi}(X_i)|\, T_i = 1$.
- Estimate ATT by

$$\widehat{ATT} = \frac{1}{n_t} \sum_{T_i=1} Y_i - \hat{\phi}(X_i)$$

ATT Estimation using MWCSRF

- Run MWCSRF with training units $\mathbf{D_{tr}}$ equal to all control units and test set $\mathbf{D_{te}}$ equal to all treated units.
- Define the prognostic score $\phi(X_i) = E(Y_i|X_i, T_i = 0)$ (Hansen).
- Use MWCSRF results to get prediction of the prognostic score for all treated units $\hat{\phi}(X_i)|\, T_i = 1$.
- Estimate ATT by

$$\widehat{ATT} = \frac{1}{n_t} \sum_{T_i=1} Y_i - \hat{\phi}(X_i)$$

- Can show that estimate of ATT is unbiased given strong ignorability of covariates X_i and treatment T_i and unbiased estimation of the prognostic score.

MWCSRF properties

- Pros: When control units are concentrated together, reduces noise in estimation of good bootstrap sampling weights.
- When parallelization is not available, large reduction in computational cost when compared to CSRF.

MWCSRF properties

- Pros: When control units are concentrated together, reduces noise in estimation of good bootstrap sampling weights.
- When parallelization is not available, large reduction in computational cost when compared to CSRF.
- Does not use treated responses to estimate control outcomes for treated units. That is, estimates of prognostic score are "out of bag," which can reduce bias substantially.

MWCSRF properties

- Pros: When control units are concentrated together, reduces noise in estimation of good bootstrap sampling weights.
- When parallelization is not available, large reduction in computational cost when compared to CSRF.
- Does not use treated responses to estimate control outcomes for treated units. That is, estimates of prognostic score are "out of bag," which can reduce bias substantially.
- Cons: Some worry about efficiency in estimates. Not using treated units in estimation necessarily is "throwing away" data.
- Problem mitigated by "honesty" assumption in Causal Random Forests (Wager & Athey).
- However, no clear, computationally efficient way to implement both honesty and weighting of bootstrap samples.

Simulation

- Small simulation to demonstrate potential usefulness of MWCSRF.
- Sample sizes (n_c, n_t) : $(200, 10), (500, 20)$, and $(1000, 50)$.

Simulation

- Small simulation to demonstrate potential usefulness of MWCSRF.
- Sample sizes (n_c, n_t) : $(200, 10), (500, 20)$, and $(1000, 50)$.
- Control responses simulated as:

$$Y_0 = \frac{\log(|X_{i1}| + |X_{i2}|)}{X_{i3}} + \epsilon, \quad \epsilon \sim N(0, 1).$$

Homogeneous model: $Y_1 = Y_0 + 4$

Simulation

- Small simulation to demonstrate potential usefulness of MWCSRF.
- Sample sizes (n_c, n_t) : $(200, 10), (500, 20)$, and $(1000, 50)$.
- Control responses simulated as:

$$Y_0 = \frac{\log(|X_{i1}| + |X_{i2}|)}{X_{i3}} + \epsilon, \quad \epsilon \sim N(0, 1).$$

  Homogeneous model: $Y_1 = Y_0 + 4$
- Control and treated units are simulated indepedently from multivariate normal distribution with slight dependency:
- Control units: $X_c \sim MultNorm(\mathbf{0}, \Sigma)$
  $\Sigma_{ij} = 0.001^{|i-j|}, 1 \leq i, j \leq p, p = 5, 10, 500$
- Treated units: $X_t \sim MultNorm(-\mathbf{0.5}, \frac{1}{16}\Sigma)$.

Simulation

- Small simulation to demonstrate potential usefulness of MWCSRF.
- Sample sizes (n_c, n_t) : $(200, 10), (500, 20)$, and $(1000, 50)$.
- Control responses simulated as:

$$Y_0 = \frac{\log(|X_{i1}| + |X_{i2}|)}{X_{i3}} + \epsilon, \quad \epsilon \sim N(0, 1).$$

  Homogeneous model: $Y_1 = Y_0 + 4$
- Control and treated units are simulated indepedently from multivariate normal distribution with slight dependency:
- Control units: $X_c \sim MultNorm(\mathbf{0}, \Sigma)$
  $\Sigma_{ij} = 0.001^{|i-j|}, 1 \leq i, j \leq p, p = 5, 10, 500$
- Treated units: $X_t \sim MultNorm(-\mathbf{0.5}, \frac{1}{16}\Sigma)$.
- Generate 100 different datasets. For each dataset, perform 4,000 bootstrap samples each for bagging and for random forest estimation.

Plot of covariate space

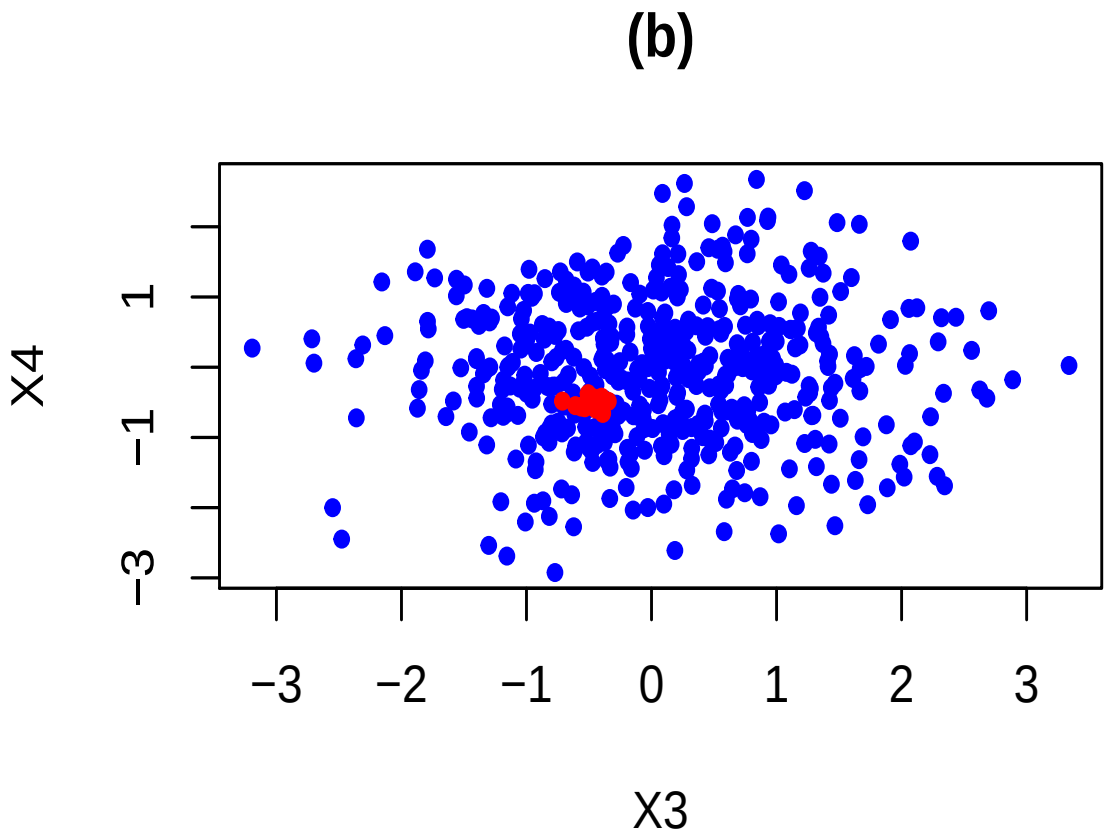

Figure: $n = 520$, $n_c = 500$, and $N_t = 20$. Red are treated units, blue are control units.

Model	P	Rep	n_c	n_t	SE	Bias
Mean weight	5	100	200	10	0.1714	0.0345
Random Forest	5	100	200	10	0.2733	0.3464
Mean weight	10	100	200	10	0.1452	0.2518
Random Forest	10	100	200	10	0.1846	0.3329
Mean weight	50	100	200	10	0.0390	0.1108
Random Forest	50	100	200	10	0.0382	0.1387
Mean weight	5	100	500	20	0.2272	0.0445
Random Forest	5	100	500	20	0.3699	0.0516
Mean weight	10	100	500	20	0.0367	0.1259
Random Forest	10	100	500	20	0.6390	0.1636
Mean weight	50	100	500	20	0.0282	0.0375
Random Forest	50	100	500	20	0.0296	0.0585
Mean weight	5	100	1000	50	0.0792	0.0665
Random Forest	5	100	1000	50	0.2927	0.1397
Mean weight	10	100	1000	50	0.0256	0.0211
Random Forest	10	100	1000	50	0.0433	0.0472
Mean weight	50	100	1000	50	0.0163	0.1222
Random Forest	50	100	1000	50	0.0172	0.1049

Conclusion

- The MWCSRF method seems to outperform standard random forest methods. Difference between methods is most significant when n_t is small.
- Similar results for heterogeneous data model.
- Still to do: Comparison with other random forest methods for causal inference.

Thank you