Design-based problem solving

Drew Dimmery

July 16, 2024

Hertie School
Data Science Lab

What are we doing?

1. Take data as given

CausalML

1. Take data as given
2. Derive better ways to use that data

CausalML

1. Take data as given
2. Derive better ways to use that data
    - Effect estimation with empirical Bayes [Dimmery et al., 2019]

CausalML

1. Take data as given
2. Derive better ways to use that data
   - Effect estimation with empirical Bayes [Dimmery et al., 2019]
   - Balancing weights [Arbour et al., 2021b]

CausalML

1. Take data as given
2. Derive better ways to use that data
   - Effect estimation with empirical Bayes [Dimmery et al., 2019]
   - Balancing weights [Arbour et al., 2021b]
   - HTE estimation [Dimmery, 2023, Leng and Dimmery, 2024]

CausalML

1. Take data as given
2. Derive better ways to use that data
    - Effect estimation with empirical Bayes [Dimmery et al., 2019]
    - Balancing weights [Arbour et al., 2021b]
    - HTE estimation [Dimmery, 2023, Leng and Dimmery, 2024]

CausalML

1. Take data as given
2. Derive better ways to use that data
   - Effect estimation with empirical Bayes [Dimmery et al., 2019]
   - Balancing weights [Arbour et al., 2021b]
   - HTE estimation [Dimmery, 2023, Leng and Dimmery, 2024]

Problem:

> *To consult the statistician after an experiment is finished is often merely to ask him to conduct a post mortem examination. He can perhaps say what the experiment died of.*
>
> *Ronald Fisher*

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

A: Binary assignment

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

A: Binary assignment

X: Covariate vector

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

A: Binary assignment

X: Covariate vector

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

A: Binary assignment

X: Covariate vector

A is the only source of randomness

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

A: Binary assignment

X: Covariate vector

A is the only source of randomness

Analysis based entirely on the randomization distribution of A.

Design-based inference

Suppose data is n tuples of (Y, A, X)

Y: Outcome, $Y = AY(1) + (1 - A)Y(0)$

A: Binary assignment

X: Covariate vector

A is the only source of randomness

Analysis based entirely on the randomization distribution of A.

Many Examples: Abadie et al. [2022], Lin [2013], Aronow and Middleton [2013], Sävje et al. [2021], Samii and Aronow [2012]

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

- Change the data you receive!

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

- Change the data you receive!
- Focus on what we control: $p(A_1, \ldots, A_n)$

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

- Change the data you receive!
- Focus on what we control: $p(A_1, \ldots, A_n)$

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

- Change the data you receive!
- Focus on what we control: $p(A_1, \ldots, A_n)$

Quick Example:

- Problem: Interference over social networks

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

- Change the data you receive!
- Focus on what we control: $p(A_1, \ldots, A_n)$

Quick Example:

- Problem: Interference over social networks
- Solution: Partition the network so that members of a cluster are mainly connected to *other* members of that cluster, assign treatment based on clusters

Design-based problem solving

Bringing these two ideas together, how can we use the design-based way of thinking to solve important substantive problems?

- Change the data you receive!
- Focus on what we control: $p(A_1, \ldots, A_n)$

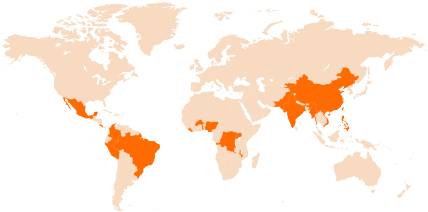
Quick Example:

- Problem: Interference over social networks
- Solution: Partition the network so that members of a cluster are mainly connected to *other* members of that cluster, assign treatment based on clusters
- "Graph-cluster randomization" Ugander et al. [2013], Eckles et al. [2017], Ugander and Yin [2023]

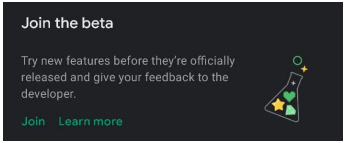
Design-based generalization

[Phan et al., 2021]

Motivation



(a) The Metaketa Initiative's study sites

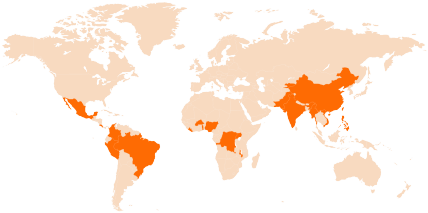


(b) Google Play's Beta program

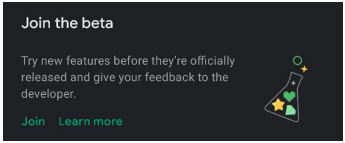
Experimental samples in practice:

- Distinct from the population of interest, e.g., tests can be performed in **one country** and we wish to generalize to the whole world

Motivation



(a) The Metaketa Initiative's study sites

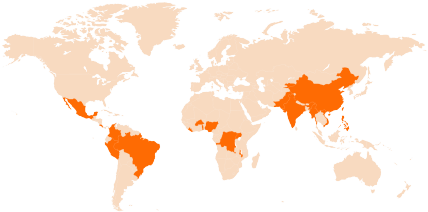


(b) Google Play's Beta program

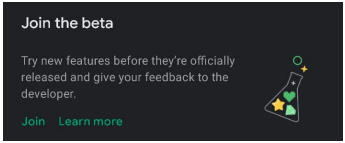
Experimental samples in practice:

- Distinct from the population of interest, e.g., tests can be performed in **one country** and we wish to generalize to the whole world
- A subpopulation of the population of interest, e.g., **beta testers** of a product

Motivation



(a) The Metaketa Initiative's study sites

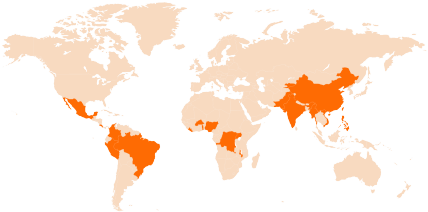


(b) Google Play's Beta program

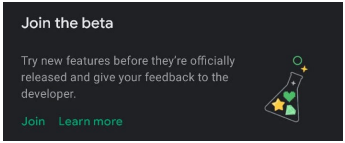
Experimental samples in practice:

- Distinct from the population of interest, e.g., tests can be performed in **one country** and we wish to generalize to the whole world
- A subpopulation of the population of interest, e.g., **beta testers** of a product

Motivation



(a) The Metaketa Initiative's study sites



(b) Google Play's Beta program

Experimental samples in practice:

- Distinct from the population of interest, e.g., tests can be performed in **one country** and we wish to generalize to the whole world
- A subpopulation of the population of interest, e.g., **beta testers** of a product

Key Question: How can we **design** experiments on our sample that will **generalize**?

Estimation via Posthoc Weighting and Target Balance

- Importance-weighted estimator $\rightarrow$ *unbiased estimator for the Target's ATE.*

Estimation via Posthoc Weighting and Target Balance

- Importance-weighted estimator $\rightarrow$ *unbiased estimator for the Target's ATE*.
- Rerandomization:

Estimation via Posthoc Weighting and Target Balance

- Importance-weighted estimator $\rightarrow$ *unbiased estimator for the Target's ATE*.
- Rerandomization:
  - Repeatedly randomize treatment assignments until a criteria (balance) is satisfied

Estimation via Posthoc Weighting and Target Balance

- Importance-weighted estimator $\rightarrow$ *unbiased estimator for the Target's ATE*.
- Rerandomization:
  - Repeatedly randomize treatment assignments until a criteria (balance) is satisfied
- Make the average observed features in the Treatment and Control similar ("balance")

Estimation via Posthoc Weighting and Target Balance

- Importance-weighted estimator $\rightarrow$ *unbiased estimator for the Target's ATE*.
- Rerandomization:
  - Repeatedly randomize treatment assignments until a criteria (balance) is satisfied
- Make the average observed features in the Treatment and Control similar ("balance")
  - **Source Balance**: make the users balance according to the Source (*status quo*)

Estimation via Posthoc Weighting and Target Balance

- Importance-weighted estimator $\rightarrow$ *unbiased estimator for the Target's ATE*.
- Rerandomization:
  - Repeatedly randomize treatment assignments until a criteria (balance) is satisfied
- Make the average observed features in the Treatment and Control similar ("balance")
  - **Source Balance**: make the users balance according to the Source (*status quo*)
  - **Target Balance**: using importance weights, make the users balance according to the Target $\rightarrow$ *reduces variance and achieves lower variance than Source Balance*.

Weighted Rerandomization-based Design

With a rejection threshold α, define the balance condition:

$$\phi^{\alpha}(\mathbf{x}, \mathbf{Z}) = \begin{cases} 1, & \text{if } M(\frac{2}{n}(\mathbf{wx})^T\mathbf{Z}) < a(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{2}{n}\sum_{Z_i=1} w_i x_i - \frac{2}{n}\sum_{Z_i=-1} w_i x_i = \frac{2}{n}(wx)^T Z.$$

Weighted Rerandomization-based Design

With a rejection threshold α, define the balance condition:

$$\phi^{\alpha}(\mathbf{x}, \mathbf{Z}) = \begin{cases} 1, & \text{if } M(\frac{2}{n}(\mathbf{wx})^T\mathbf{Z}) < a(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{2}{n}\sum_{Z_i=1} w_i x_i - \frac{2}{n}\sum_{Z_i=-1} w_i x_i = \frac{2}{n}(wx)^T Z.$$

When weights are all one, you get the standard Mahalanobis distance (i.e. Euclidean scaled by the covariance matrix).

1. Assign A randomly for each person $1, \ldots, n$ such that $\sum_i a_i = n_1$. There are $\binom{n}{n_1}$ ways to choose, each of which is equally likely.

Weighted Rerandomization-based Design

With a rejection threshold α, define the balance condition:

$$\phi^{\alpha}(\mathbf{x}, \mathbf{Z}) = \begin{cases} 1, & \text{if } M(\frac{2}{n}(\mathbf{wx})^T\mathbf{Z}) < a(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{2}{n}\sum_{Z_i=1} w_i x_i - \frac{2}{n}\sum_{Z_i=-1} w_i x_i = \frac{2}{n}(wx)^T Z.$$

When weights are all one, you get the standard Mahalanobis distance (i.e. Euclidean scaled by the covariance matrix).

1. Assign A randomly for each person $1, \ldots, n$ such that $\sum_i a_i = n_1$. There are $\binom{n}{n_1}$ ways to choose, each of which is equally likely.
2. If $\phi^{\alpha}(\mathbf{x}, \mathbf{z}) = 0$ return to step (1).

Weighted Rerandomization-based Design

With a rejection threshold α, define the balance condition:

$$\phi^{\alpha}(\mathbf{x}, \mathbf{Z}) = \begin{cases} 1, & \text{if } M(\frac{2}{n}(\mathbf{wx})^T\mathbf{Z}) < a(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{2}{n}\sum_{Z_i=1} w_i x_i - \frac{2}{n}\sum_{Z_i=-1} w_i x_i = \frac{2}{n}(wx)^T Z.$$

When weights are all one, you get the standard Mahalanobis distance (i.e. Euclidean scaled by the covariance matrix).

1. Assign A randomly for each person $1, \ldots, n$ such that $\sum_i a_i = n_1$. There are $\binom{n}{n_1}$ ways to choose, each of which is equally likely.
2. If $\phi^{\alpha}(\mathbf{x}, \mathbf{z}) = 0$ return to step (1).
3. Conduct experiment with treatment assignments, A.

Theory

- The importance-weighted estimator with Target Balance results in an unbiased estimator of the target distribution's ATE
- Target Balance reduces the variance and achieves a lower variance than Source Balance.
- Among all balance criteria with rejection probability at most α (including Source Balance), Target Balance achieves the optimal variance reduction

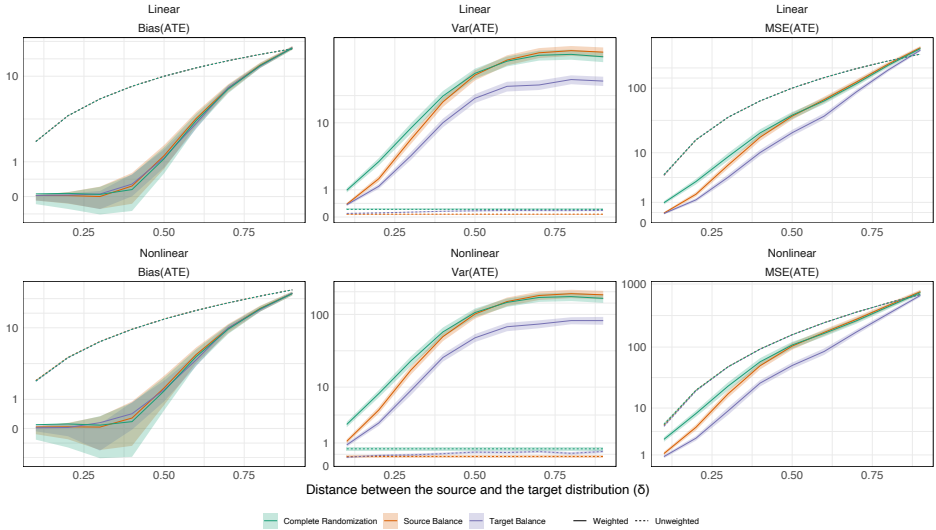
Theory

- The importance-weighted estimator with Target Balance results in an unbiased estimator of the target distribution's ATE
- Target Balance reduces the variance and achieves a lower variance than Source Balance.
- Among all balance criteria with rejection probability at most α (including Source Balance), Target Balance achieves the optimal variance reduction

Theory

- The importance-weighted estimator with Target Balance results in an unbiased estimator of the target distribution's ATE
- Target Balance reduces the variance and achieves a lower variance than Source Balance.
- Among all balance criteria with rejection probability at most α (including Source Balance), Target Balance achieves the optimal variance reduction

Simulations



Design-based HTE estimation

[Arbour et al., 2021a]

Experimental design

Approaches:

- Simple randomization (coin flipping)

Experimental design

Approaches:

- Simple randomization (coin flipping)
- Blocking [Greevy et al., 2004, Higgins et al., 2016]

Experimental design

Approaches:

- Simple randomization (coin flipping)
- Blocking [Greevy et al., 2004, Higgins et al., 2016]
- Rerandomization [Morgan et al., 2012, Li et al., 2018]

Experimental design

Approaches:

- Simple randomization (coin flipping)
- Blocking [Greevy et al., 2004, Higgins et al., 2016]
- Rerandomization [Morgan et al., 2012, Li et al., 2018]
- Optimization [Kallus, 2018, Harshaw et al., 2020]

Experimental design

Approaches:

- Simple randomization (coin flipping)
- Blocking [Greevy et al., 2004, Higgins et al., 2016]
- Rerandomization [Morgan et al., 2012, Li et al., 2018]
- **Optimization** [Kallus, 2018, Harshaw et al., 2020]

Estimation of HTE

The HTE for x is: $Y^{A=1}(x) - Y^{A=0}(x)$, $x \in X$.

We focus on the following kernel-weighted estimator:

$$\hat{f}(x_i) = \sum_{j}^{N} \frac{\mathbb{1}(a_i \neq a_j) k(x_i, x_j) y_j}{\sum_m^N \mathbb{1}(a_i \neq a_m) k(x_i, x_m)} \tag{1}$$

Estimation of HTE

The HTE for x is: $Y^{A=1}(x) - Y^{A=0}(x)$, $x \in X$.

We focus on the following kernel-weighted estimator:

$$\hat{f}(x_i) = \sum_{j}^{N} \frac{\mathbb{1}(a_i \neq a_j) k(x_i, x_j) y_j}{\sum_m^N \mathbb{1}(a_i \neq a_m) k(x_i, x_m)} \tag{1}$$

which results in the following HTE estimator:

$$\hat{\tau}_i = (2a_i - 1) \left(y_i - \sum_{j=1}^{n} w_{ij} y_j \right) \tag{2}$$

Estimation of HTE

The HTE for x is: $Y^{A=1}(x) - Y^{A=0}(x)$, $x \in X$.

We focus on the following kernel-weighted estimator:

$$\hat{f}(x_i) = \sum_{j}^{N} \frac{\mathbb{1}(a_i \neq a_j) k(x_i, x_j) y_j}{\sum_m^N \mathbb{1}(a_i \neq a_m) k(x_i, x_m)} \tag{1}$$

which results in the following HTE estimator:

$$\hat{\tau}_i = (2a_i - 1)\left(y_i - \sum_{j=1}^{n} w_{ij} y_j\right) \tag{2}$$

Fixing observed outcomes leads to the transductive inference estimator of Zhu et al. [2003].

Similar to singular kernels analyzed in benign overfitting literature [Belkin et al., 2018].

Defining the optimization problem for the HTE

Minimize the sum of absolute errors in each response surface:

$$\min_{A} \sum_{i}^{N} \left| \sum_{j} \frac{\mathbb{1}(a_i \neq a_j) w_{ij} y_j}{\sum_k \mathbb{1}(a_i \neq a_k) w_{ik}} - \tilde{y}_i \right| \tag{3}$$

Defining the optimization problem for the HTE

Minimize the sum of absolute errors in each response surface:

$$\min_{A} \sum_{i}^{N} \left| \sum_{j} \frac{\mathbb{1}(a_i \neq a_j) w_{ij} y_j}{\sum_k \mathbb{1}(a_i \neq a_k) w_{ik}} - \tilde{y}_i \right| \tag{3}$$

where $\tilde{y}_i$ is the unobserved counterfactual for i.

$$\sum_{i}^{N} \left| \frac{\sum_j \mathbb{1}(a_i \neq a_j) w_{ij} y_j}{\sum_k \mathbb{1}(a_i \neq a_k) w_{ik}} - y_i \right| \leq \sum_{i} |\tilde{y}_i - \hat{y}_i^*| + \frac{w_{\text{sum}} - \sum_{i,j} \mathbb{1}(a_i \neq a_j) w_{ij}}{d_{\text{min}}} \tag{4}$$

- w_{sum}: sum of weights
- d_{min}: minimum degree
- $\hat{y}_i^*$: weighted estimator of $\tilde{y}_i$ if all other units received treatment $1 - a_i$.

Defining the optimization problem for the HTE

Minimize the sum of absolute errors in each response surface:

$$\min_{A} \sum_{i}^{N} \left| \sum_{j} \frac{\mathbb{1}(a_i \neq a_j) w_{ij} y_j}{\sum_k \mathbb{1}(a_i \neq a_k) w_{ik}} - \tilde{y}_i \right| \tag{3}$$

where $\tilde{y}_i$ is the unobserved counterfactual for i.

$$\sum_{i}^{N} \left| \frac{\sum_j \mathbb{1}(a_i \neq a_j) w_{ij} y_j}{\sum_k \mathbb{1}(a_i \neq a_k) w_{ik}} - y_i \right| \leq \sum_{i} |\tilde{y}_i - \hat{y}_i^*| + \frac{w_{\text{sum}} - \sum_{i,j} \mathbb{1}(a_i \neq a_j) w_{ij}}{d_{\min}} \tag{4}$$

- w_{sum}: sum of weights
- $d_{\min}$: minimum degree
- $\hat{y}_i^*$: weighted estimator of $\tilde{y}_i$ if all other units received treatment $1 - a_i$.

Intuition: There is irreducible error from the counterfactual, and reducible error from assignment to treatment which is minimized by maximizing $\sum_{i,j} \mathbb{1}(a_i \neq a_j) w_{ij}$: MAXCUT

SoftBlock Algorithm

input : $X \in \mathbb{R}^D$

output: Assignments A, Spanning Tree T

1 $G \leftarrow$ Similarity matrix constructed from X

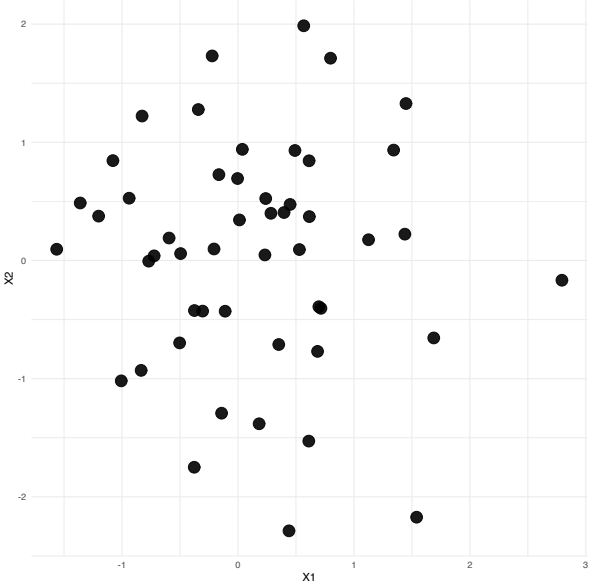
2 $T \leftarrow$ Maximum Spanning Tree(G)

3 $A \leftarrow$ MAXCUT(T)

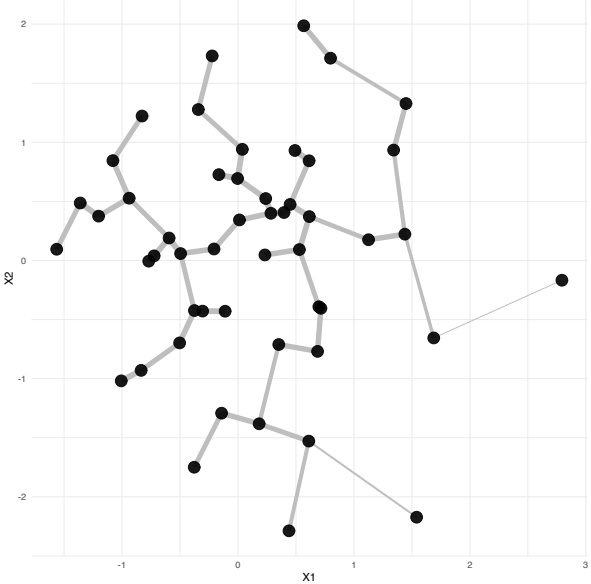
Algorithm: Friedman-Rafsky Minimizing Design

- (2) is calculated in $\mathcal{O}(n^2 \log n)$.
- (3) is calculated in $\mathcal{O}(n)$ on a tree.
- An additional algorithm (GreedyNeighbors) based on the nearest neighbor graph simply substitutes (2) for constructing the nearest neighbor graph.

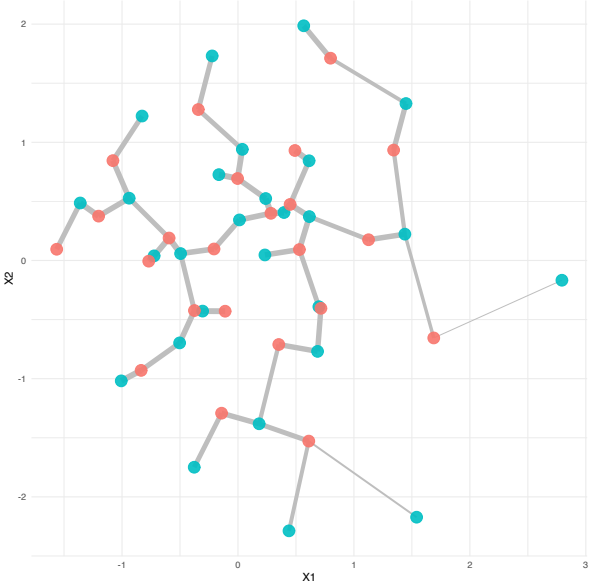
Take a point-cloud in $\mathbb{R}^2$



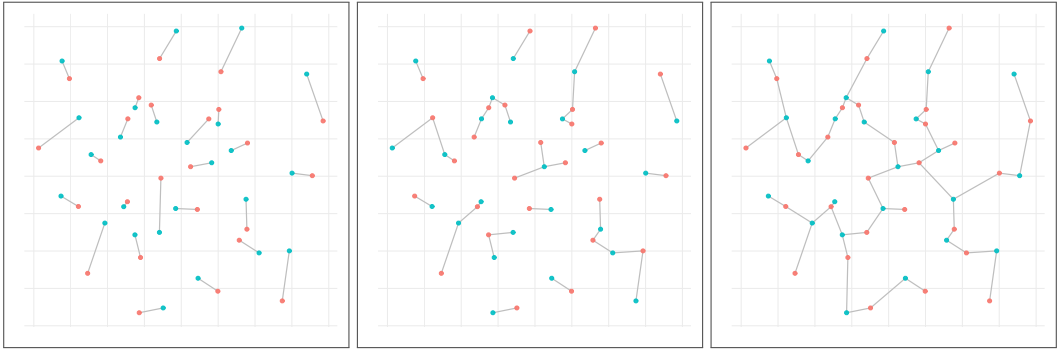
Find the maximum spanning tree



Greedily assign adjacent nodes alternating treatments



Three Designs on the Same Data in $\mathbb{R}^2$

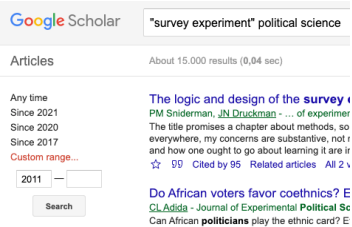


Matched Pairs GreedyNeighbors SoftBlock

Design-based sequential balance

[Arbour et al., 2022]

Motivation



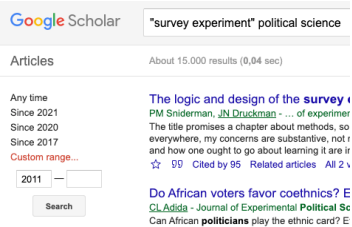
(a) We run a lot of survey experiments

(b) A criminally under-cited paper

Design of survey experiments in practice:

- Just flip a coin, it's fine!

Motivation



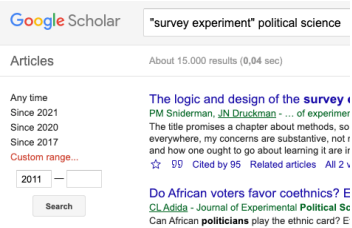
(a) We run a lot of survey experiments

(b) A criminally under-cited paper

Design of survey experiments in practice:

- Just flip a coin, it's fine!
- Slow balancing designs (either offline or with poor time complexity)

Motivation



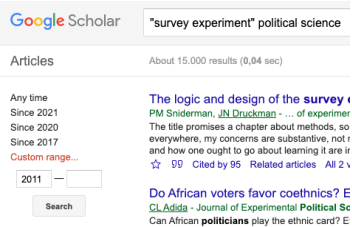
(a) We run a lot of survey experiments

(b) A criminally under-cited paper

Design of survey experiments in practice:

- Just flip a coin, it's fine!
- Slow balancing designs (either offline or with poor time complexity)

Motivation



(a) We run a lot of survey experiments

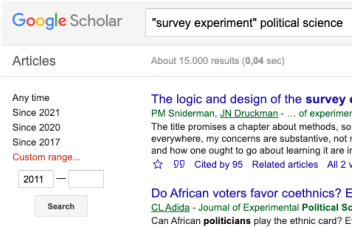
(b) A criminally under-cited paper

Design of survey experiments in practice:

- Just flip a coin, it's fine!
- Slow balancing designs (either offline or with poor time complexity)

Methods Question: How can we guarantee balance with highly efficient designs?

Motivation



(a) We run a lot of survey experiments

(b) A criminally under-cited paper

Design of survey experiments in practice:

- Just flip a coin, it's fine!
- Slow balancing designs (either offline or with poor time complexity)

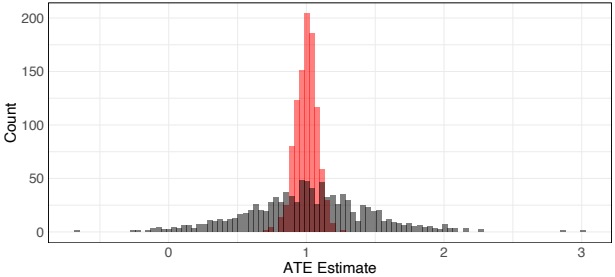
Methods Question: How can we guarantee balance with highly efficient designs?

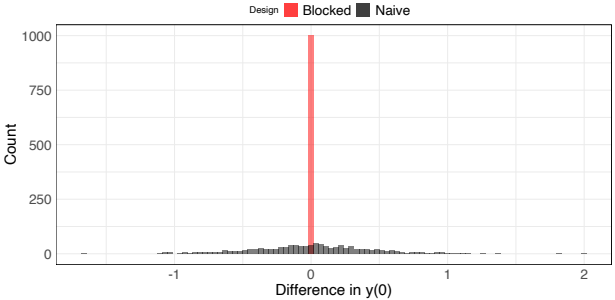
Software Question: How can we make it easy to run balancing designs on sequential data?

What makes good sequential experimental design?

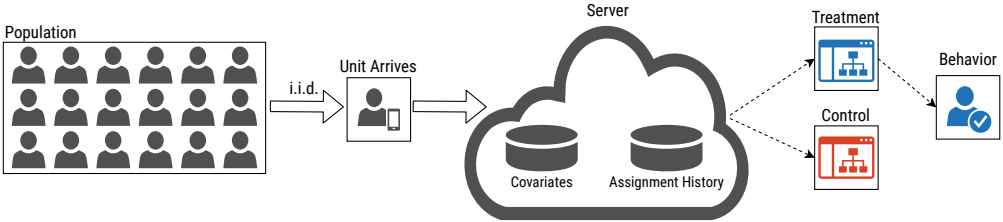
1. Balances Covariates

- Effective at minimizing imbalance
  - Better than coin-flipping
  - Ideally with theoretical guarantees



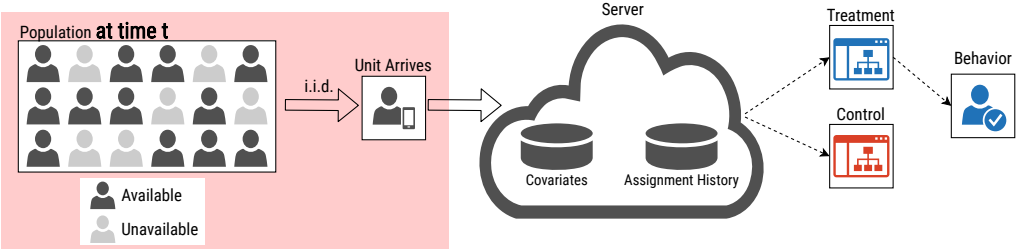


Idealized Sequential Experiment



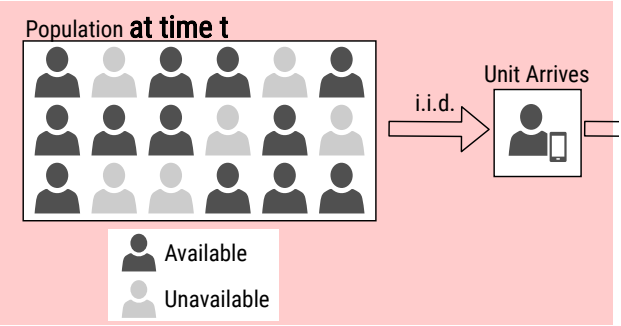
Sampling frames vary over time

More Realistic Sequential Experiment



2. Incorporates randomization to improve robustness

- Appropriate to the data generating process
- Not easily “exploitable” by an adversary
- Ideally, we would be able to turn a dial like [Harshaw et al., 2020].



How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals
  - Adversary knows the whole history of the experiment.

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals
  - Adversary knows the whole history of the experiment.
  - [Alweiss et al., 2020]: randomization is minimax optimal, a la Kallus [2018].

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals
  - Adversary knows the whole history of the experiment.
  - [Alweiss et al., 2020]: randomization is minimax optimal, a la Kallus [2018].
- Oblivious adversarial arrivals

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals
  - Adversary knows the whole history of the experiment.
  - [Alweiss et al., 2020]: randomization is minimax optimal, a la Kallus [2018].
- Oblivious adversarial arrivals
  - Adversary *does not know* historical treatment assignments.

How can we formalize this?

(At least) three approaches:

- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals
  - Adversary knows the whole history of the experiment.
  - [Alweiss et al., 2020]: randomization is minimax optimal, a la Kallus [2018].
- Oblivious adversarial arrivals
  - Adversary *does not know* historical treatment assignments.
  - Fairly common in CS literature [Alweiss et al., 2020, Dwivedi and Mackey, 2021]

How can we formalize this?

(At least) three approaches:

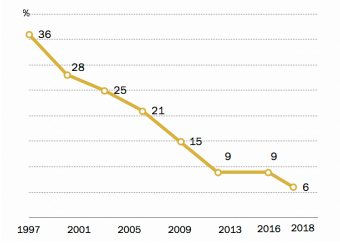
- i.i.d. Stochastic arrivals / SRS from population
  - No randomization is necessary
  - Each group-specific mean converges to the average potential outcome in the population
- Fully adversarial arrivals
  - Adversary knows the whole history of the experiment.
  - [Alweiss et al., 2020]: randomization is minimax optimal, a la Kallus [2018].
- Oblivious adversarial arrivals
  - Adversary *does not know* historical treatment assignments.
  - Fairly common in CS literature [Alweiss et al., 2020, Dwivedi and Mackey, 2021]
  - This work.

3. Doesn't assign units who won't actually show up.

- Focuses design on respondents.

After brief plateau, telephone survey response rates have fallen again

Response rate by year (%)



Note: Response rate is AAPOR RR3. Only landlines sampled 1997-2006. Rates are typical for surveys conducted in each year.
Source: Pew Research Center telephone surveys conducted 1997-2018.

PEW RESEARCH CENTER

4. Fast

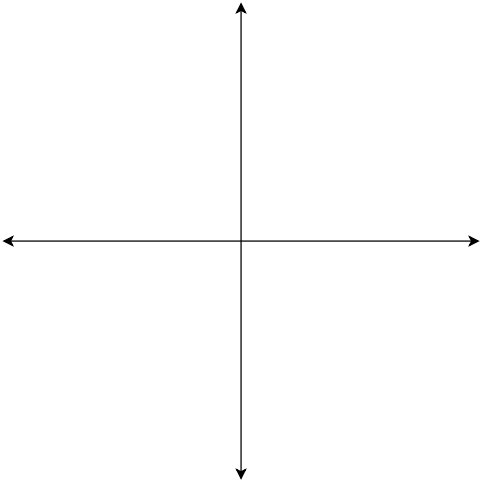
- Single-pass (i.e. must work online)
- Ideally constant-time in each iteration (i.e. overall $\mathcal{O}(n)$)

using covariate-adaptive randomization for large surveys, I have estimated
dents, the 4,000th respondent takes 5 seconds on average to assign. This es
`eqblock` on a Macbook Pro and might not be fully reflective of how the al

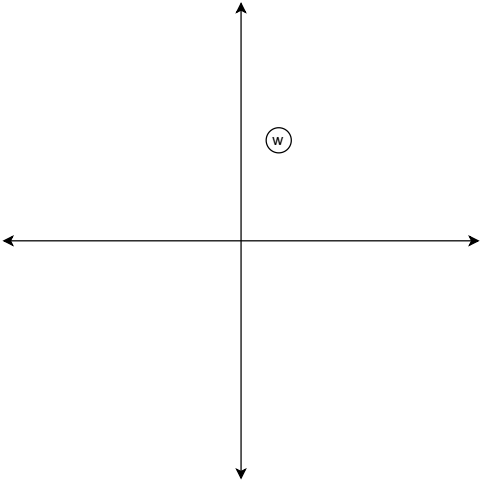
Cavaille [2018] on Moore and Moore [2017]

The algorithm

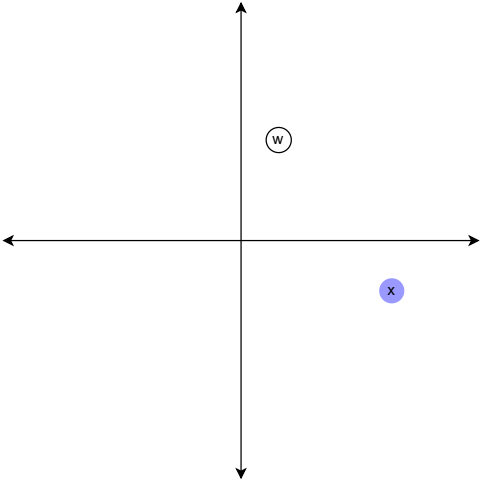
A covariate space



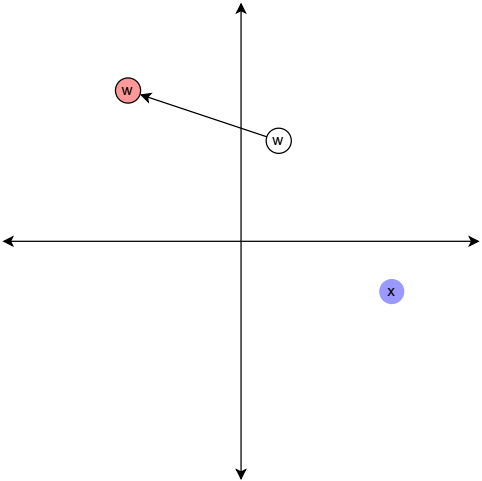
Current imbalance



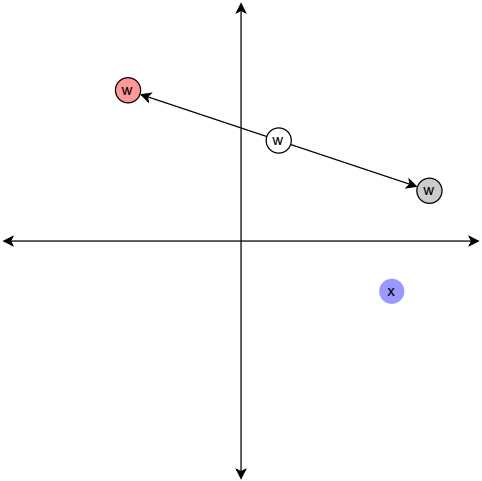
New unit arrives



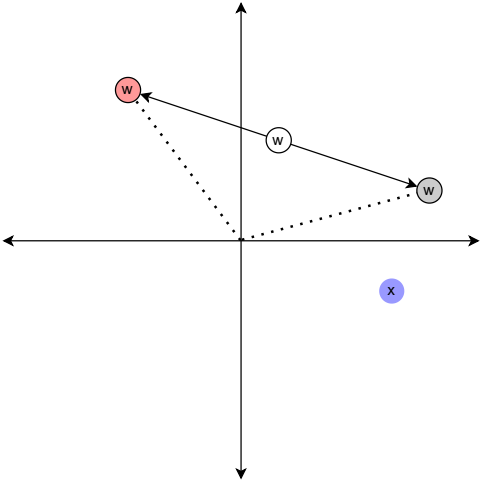
Could assign the new unit to control



Or could assign the new unit to treatment

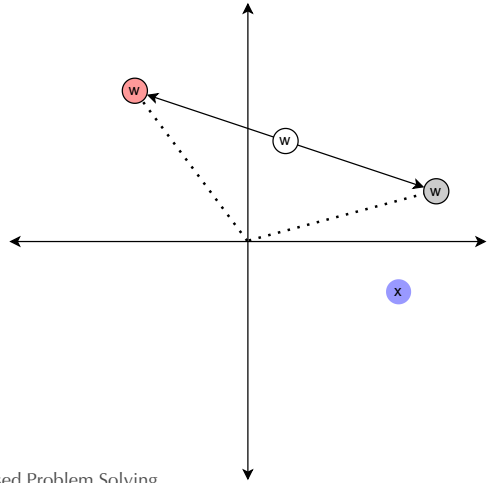


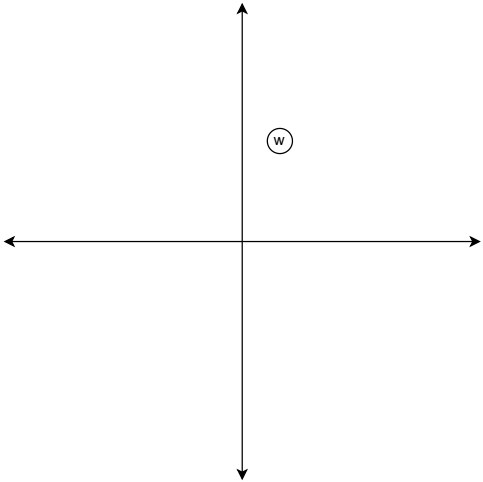
Either treatment results in similar levels imbalance



Repeating the exercise

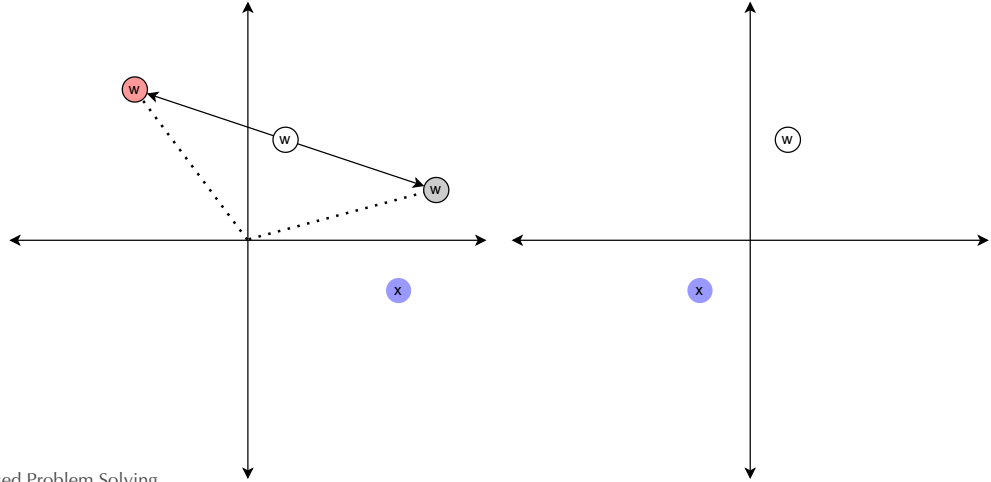
Assignment doesn't matter much





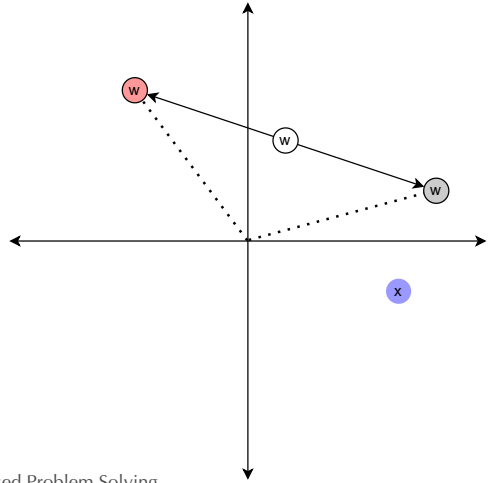
What if a different unit had arrived?

Assignment doesn't matter much

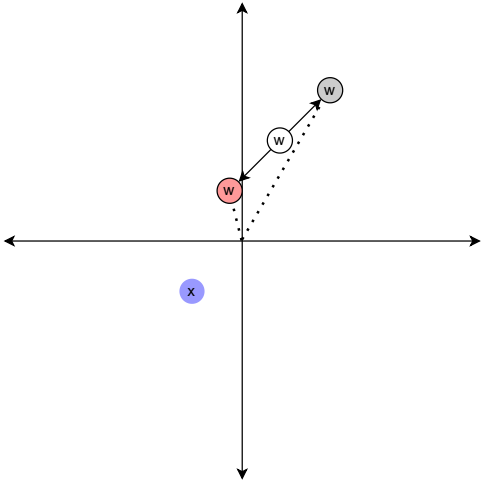


If a different unit had arrived, assignment would matter a lot

Assignment doesn't matter much

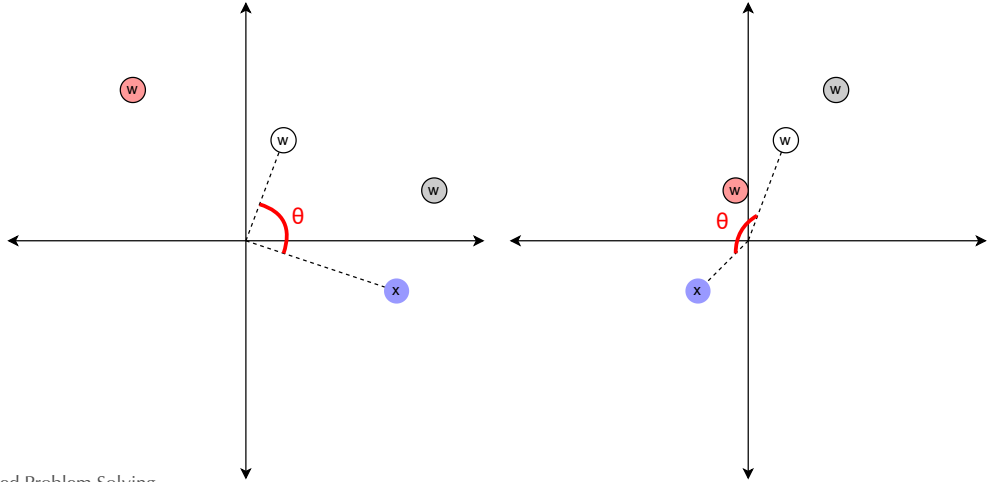


Assignment matters a lot

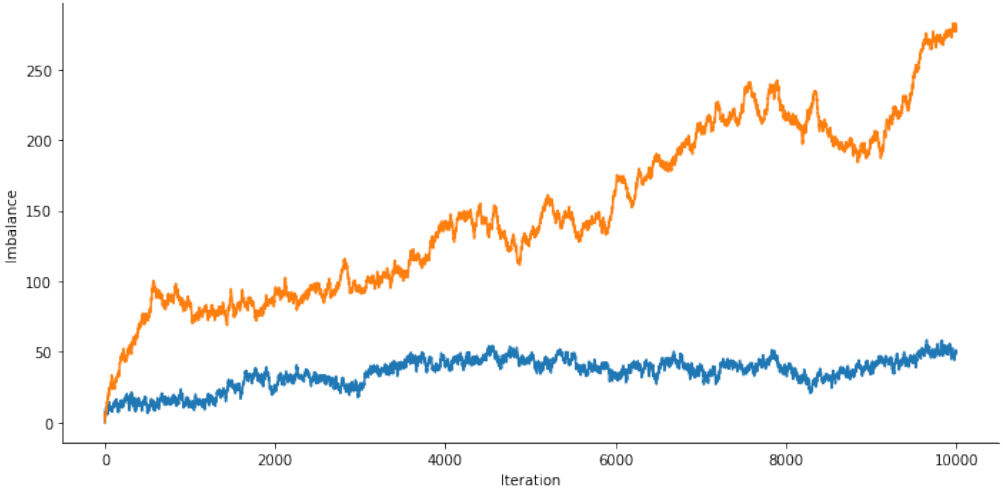


This motivates the algorithm: The Balancing Walk Design (BWD)

$\therefore$ Assign based on $\cos\theta$.

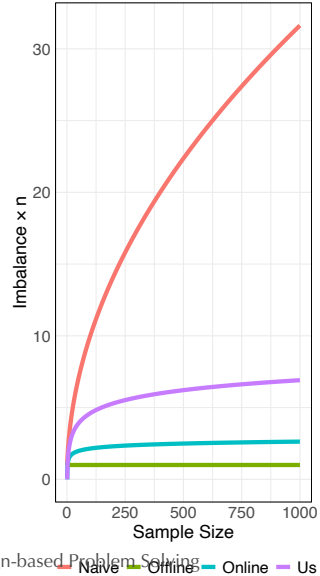


A self-balancing random walk of imbalance (drifts back towards zero)



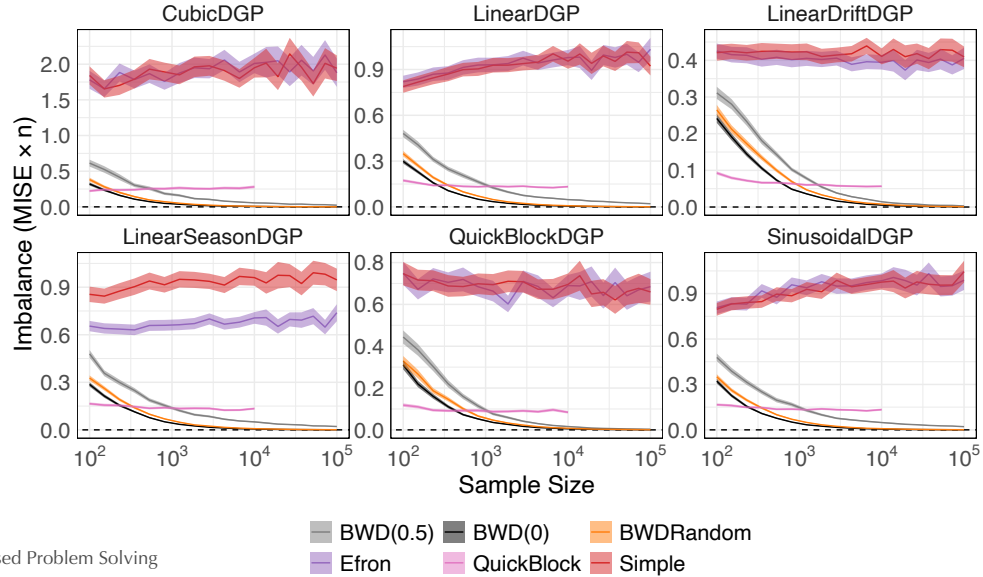
Blue: the Balancing Walk Design (BWD)

Total imbalance is small

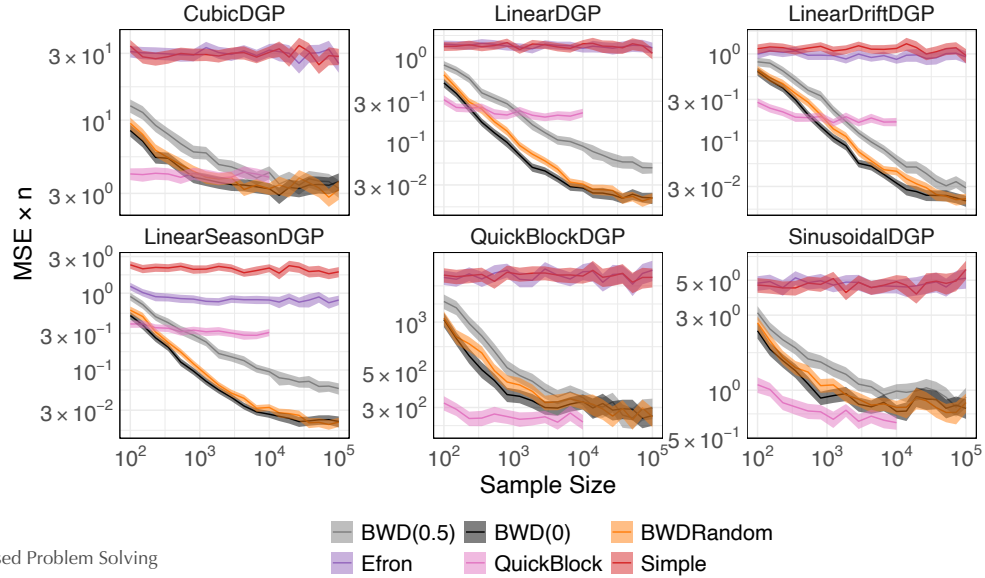


- Total imbalance: $\|\sum_i a_i \mathbf{x}_i - \sum_i (1 - a_i)\mathbf{x}_i\|_\infty$
- Our method (BWD): $\log(n/\delta)$
  - Pure randomization: $\sqrt{n}$
  - Optimal (online) rate: $\sqrt{\log n}$
  - Offline optimal rate [Harshaw et al., 2020]: 1
- A knob controls how much to prioritize imbalance versus robustness, as in Harshaw et al. [2020].
- Will be better than post-hoc covariate adjustment.
- Inference is (mostly) standard.

Two balanced treatments - L_2 norm



Two balanced treatments - causal MSE



Summary and future direction

- Better design is a huge opportunity.
- Must consider the social science context.
- But how can we change the practice of experimentation?
  - Software

Thank You!

d.dimmery@hertie-school.org

References

Alberto Abadie, Susan Athey, Guido W Imbens, and Jeffrey M Wooldridge. When Should You Adjust Standard Errors for Clustering?*. *The Quarterly Journal of Economics*, 138(1): 1–35, 10 2022. ISSN 0033-5533. doi: 10.1093/qje/qjac038. URL `https://doi.org/10.1093/qje/qjac038`.

Ryan Alweiss, Yang P Liu, and Mehtaab Sawhney. Discrepancy minimization via a self-balancing walk. *arXiv preprint arXiv:2006.14009*, 2020.

David Arbour, Drew Dimmery, and Anup Rao. Efficient balanced treatment assignments for experimentation. In *International Conference on Artificial Intelligence and Statistics*, pages 3070–3078. PMLR, 2021a.

References ii

David Arbour, Drew Dimmery, and Arjun Sondhi. Permutation weighting. In *International Conference on Machine Learning*, pages 331–341. PMLR, 2021b.

David Arbour, Drew Dimmery, Tung Mai, and Anup Rao. Online balanced experimental design. In *International Conference on Machine Learning*, pages 844–864. PMLR, 2022.

Peter M Aronow and Joel A Middleton. A class of unbiased estimators of the average treatment effect in randomized experiments. *Journal of Causal Inference*, 1(1):135–154, 2013.

Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. *Advances in neural information processing systems*, 31, 2018.

Charlotte Cavaille. Implementing blocked randomization in online survey experiments, 2018.

References iii

Drew Dimmery. *tidyhte: Tidy Estimation of Heterogeneous Treatment Effects*, 2023. https://github.com/ddimmery/tidyhte https://ddimmery.github.io/tidyhte/index.html.

Drew Dimmery, Eytan Bakshy, and Jasjeet Sekhon. Shrinkage estimators in online experiments. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2914–2922, 2019.

Raaz Dwivedi and Lester Mackey. Kernel thinning. *arXiv preprint arXiv:2105.05842*, 2021.

Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1):20150021, 2017.

Robert Greevy, Bo Lu, Jeffrey H Silber, and Paul Rosenbaum. Optimal multivariate matching before randomization. *Biostatistics*, 5(2):263–275, 2004.

References iv

Christopher Harshaw, Fredrik Sävje, Daniel Spielman, and Peng Zhang. Balancing covariates in randomized experiments using the gram-schmidt walk, 2020.

Michael J Higgins, Fredrik Sävje, and Jasjeet S Sekhon. Improving massive experiments with threshold blocking. *Proceedings of the National Academy of Sciences*, 113(27): 7369–7376, 2016.

Nathan Kallus. Optimal a priori balance in the design of controlled experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):85–112, 2018.

Yan Leng and Drew Dimmery. Calibration of heterogeneous treatment effects in randomized experiments. *Information Systems Research*, 2024.

Xinran Li, Peng Ding, and Donald B Rubin. Asymptotic theory of rerandomization in treatment–control experiments. *Proceedings of the National Academy of Sciences*, 115 (37):9157–9162, 2018.

References v

Winston Lin. Agnostic notes on regression adjustments to experimental data: Reexamining freedman's critique. *THE ANNALS of APPLIED STATISTICS*, pages 295–318, 2013.

Ryan T. Moore and Sally A. Moore. Blocking for sequential political experiments. *Political Analysis*, 21(4):507–523, 2017. doi: 10.1093/pan/mpt007.

Kari Lock Morgan, Donald B Rubin, et al. Rerandomization to improve covariate balance in experiments. *The Annals of Statistics*, 40(2):1263–1282, 2012.

My Phan, David Arbour, Drew Dimmery, and Anup Rao. Designing transportable experiments under s-admissability. In *International Conference on Artificial Intelligence and Statistics*, pages 2539–2547. PMLR, 2021.

Cyrus Samii and Peter M Aronow. On equivalencies between design-based and regression-based variance estimators for randomized experiments. *Statistics & Probability Letters*, 82(2):365–370, 2012.

References vi

Fredrik Sävje, Peter Aronow, and Michael Hudgens. Average treatment effects in the presence of unknown interference. *Annals of statistics*, 49(2):673, 2021.

Johan Ugander and Hao Yin. Randomized graph cluster randomization. *Journal of Causal Inference*, 11(1):20220014, 2023.

Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–337. ACM, 2013.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.