

# Improving Massive Experiments using Threshold Blocking: Minimizing the Within-Block Distance

Michael J. Higgins

Kansas State University

Nov. 5, 2015

Work with Fredrik Savje and Jasjeet Sekhon

- 1 Motivation
- 2 Blocking and graph theory
- 3 Approximately optimal blocking algorithm
- 4 Results
- 5 Future work

## PROBLEM 1:

- Small experiments:
- Non-negligible probability of bad covariate balance between treatment groups—a treatment group has too many Republicans, very sick people, etc.
- Bad imbalance on important covariates → Inaccurate estimates of treatment effects.
- Must prevent if cost of additional units is large (e.g. a medical trial).

## PROBLEM 2:

- Massive experiments with multiple treatment arms:
- Currently, no efficient blocking method with guaranteed performance that forms blocks with more than two units.
- In matched-pairs, non-bipartite matching may STILL be too slow.
- Needed when treatment effect sizes are very small (e.g. online advertising) or when subgroups of interest.

# Post-randomization solutions to covariate imbalance

- Post-stratification [Miratrix et al., 2013]:
  - *After* running the experiment, group similar units together to increase precision of treatment effect estimates.

# Post-randomization solutions to covariate imbalance

- Post-stratification [Miratrix et al., 2013]:
  - *After* running the experiment, group similar units together to increase precision of treatment effect estimates.
  - **Data mining,  $p$ -hacking concerns.**
  - Not as efficient as blocking (not as much of an issue).

# Post-randomization solutions to covariate imbalance

- Post-stratification [Miratrix et al., 2013]:
  - *After* running the experiment, group similar units together to increase precision of treatment effect estimates.
  - **Data mining,  $p$ -hacking concerns.**
  - Not as efficient as blocking (not as much of an issue).
- Re-randomization [Lock Morgan and Rubin, 2012]:
  - Analyze covariate balance after randomization. Repeat randomly assigning treatments until covariate balance is “acceptable.”

# Post-randomization solutions to covariate imbalance

- Post-stratification [Miratrix et al., 2013]:
  - *After* running the experiment, group similar units together to increase precision of treatment effect estimates.
  - **Data mining,  $p$ -hacking concerns.**
  - Not as efficient as blocking (not as much of an issue).
- Re-randomization [Lock Morgan and Rubin, 2012]:
  - Analyze covariate balance after randomization. Repeat randomly assigning treatments until covariate balance is “acceptable.”
  - **Not valid if decision to re-randomize is made only after bad imbalance.**
  - Standard errors calculated across only “valid” randomizations. Can be hard to compute—even when initial balance is good!



# Post-randomization solutions to covariate imbalance

- Post-stratification [Miratrix et al., 2013]:
  - *After* running the experiment, group similar units together to increase precision of treatment effect estimates.
  - **Data mining,  $p$ -hacking concerns.**
  - Not as efficient as blocking (not as much of an issue).
- Re-randomization [Lock Morgan and Rubin, 2012]:
  - Analyze covariate balance after randomization. Repeat randomly assigning treatments until covariate balance is “acceptable.”
  - **Not valid if decision to re-randomize is made only after bad imbalance.**
  - Standard errors calculated across only “valid” randomizations. Can be hard to compute—even when initial balance is good!
- **Moral of the story:** Fix these problems in the randomization scheme, not adjust after.

## Our Solution: Threshold Blocking

- *Blocking*—grouping units according to common traits (e.g. same political party, similar health) before treatment is assigned.

Completely randomize with block, independently across blocks.

- *Threshold blocking*—each block contains at least  $k$  units for some prespecified threshold  $k$ .

## Our Solution: Threshold Blocking

- *Blocking*—grouping units according to common traits (e.g. same political party, similar health) before treatment is assigned.

Completely randomize with block, independently across blocks.

- *Threshold blocking*—each block contains at least  $k$  units for some prespecified threshold  $k$ .
- Accommodates arbitrarily many treatment arms (and multiple replications within each block)
- Ensure good covariate balance in small experiments
- Efficient enough for massive experiments (100 million units)

# Our approach: Mimimize the MWBC

We analyze the following blocking method:

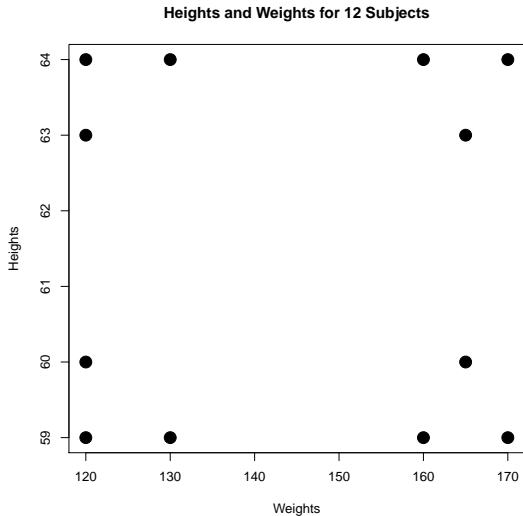
- 1 Choose a measure of dissimilarity or distance (e.g. Mahalanobis, standardized Euclidian) that is small when important covariates have similar values:  
  
*Cost:* value of this distance. **Lower cost = better match.**
- 2 Choose a threshold  $k$  for the minimum number of units to be contained in a block.
- 3 Each block contains at least  $k$  units, and the the maximum distance between any two units within a block—the maximum within-block cost (MWBC)—is minimized.

# Blocking by minimizing the MWBC

- Minimizing the MWBC: Ensures covariate balance in randomization.
- Threshold  $k$ : Allows designs with multiple treatment categories, multiple replications of treatments within a block; blocks can preserve clustering in data.
- “Good” blocking can be found very quickly: can be used in massive experiments.

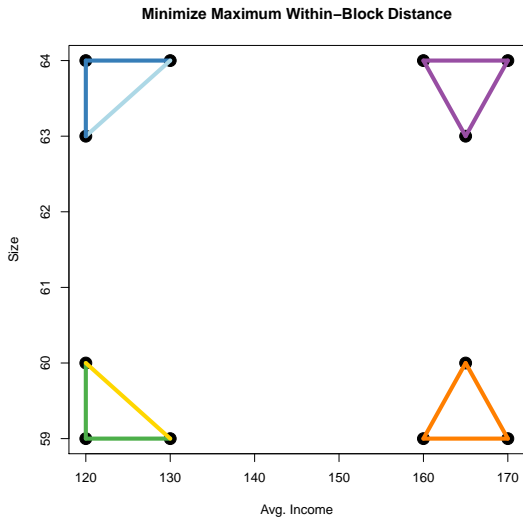
# A simple example:

Threshold  $k = 2$ . Distance = Mahalanobis distance.



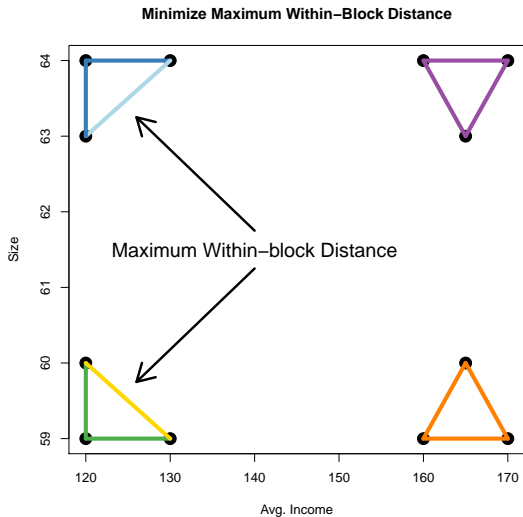
# A simple example:

Threshold  $k = 2$ . Distance = Mahalanobis distance.



# A simple example:

Threshold  $k = 2$ . Distance = Mahalanobis distance.

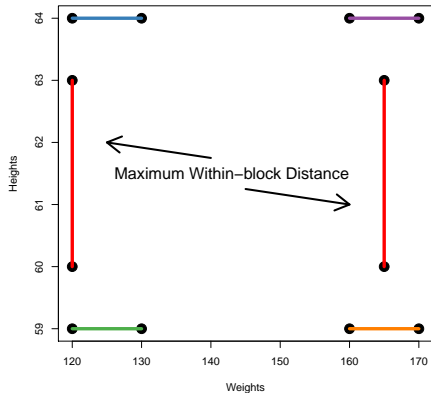




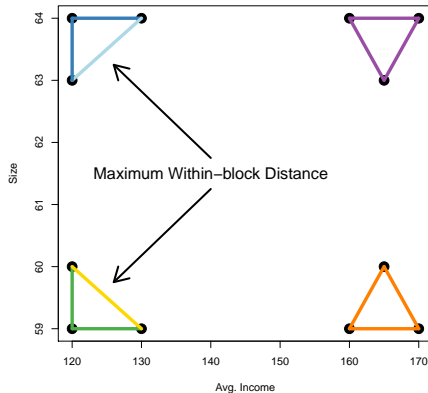
# A simple example:

Threshold  $k = 2$ . Dissimilarity = Mahalanobis distance.

Matched-Pair Blocking



Minimize Maximum Within-Block Distance



# Optimal blocking and approximately optimal blocking

*For all blockings that contain at least  $k$  units with each block:*

- Let  $\lambda$  denote the smallest MWBC achievable by such a blocking—any blocking that meets this bound is called an *optimal blocking*.
- Finding optimal blocking is NP-hard—feasible to find in small experiments, may not be in large experiments (Follows from [Kirkpatrick and Hell, 1983]).

# Optimal blocking and approximately optimal blocking

*For all blockings that contain at least  $k$  units with each block:*

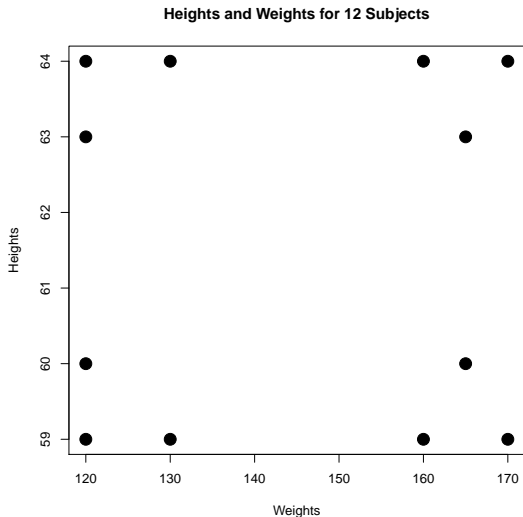
- Let  $\lambda$  denote the smallest MWBC achievable by such a blocking—any blocking that meets this bound is called an *optimal blocking*.
- Finding optimal blocking is NP-hard—feasible to find in small experiments, may not be in large experiments (Follows from [Kirkpatrick and Hell, 1983]).
- We show blocking with  $\text{MWBC} \leq 4\lambda$  is constructable in  $O(kn)$  time and space, outside of forming nearest neighbor graph  
Find “good” blocking when number of units is small *or massive*.
- Denote any such blocking as an *approximately optimal blocking*.

# Viewing experimental units as a graph

- Extend ideas from Rosenbaum [1989] and Greevy *et. al.* [2004]: Statistical blocking problems can be viewed as graph theory partitioning problems.
- Experimental units are vertices in a graph.
- Edges signify that two units can be placed in the same block.
- Edge costs are a measure of dissimilarity between pretreatment covariates (e.g. Mahalanobis, Euclidian).
- Use methods in graph theory to solve original blocking problem.

# Viewing experimental units as a graph: In pictures

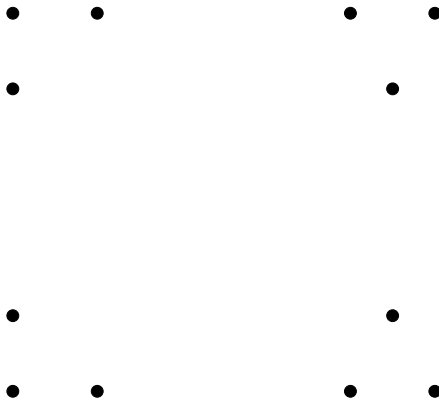
Dissimilarity = Mahalanobis distance.



# Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.

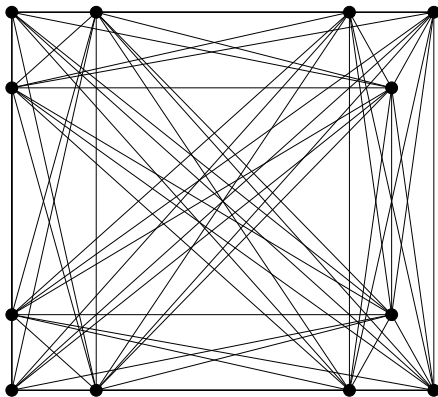
Units as a graph



# Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.

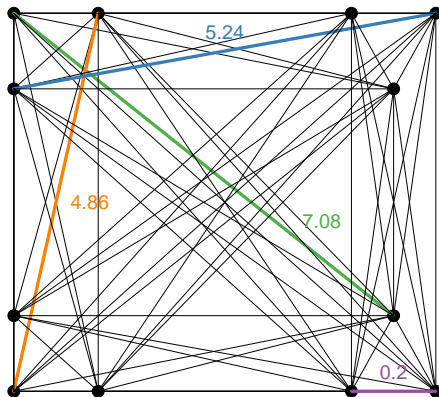
Units as a graph



# Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.

Units as a graph





# Notation:

- A graph  $G$  is defined by its vertex set  $V$  and its edge set  $E$ :  
 $G = (V, E)$ .
- Vertices in  $V$  denoted by  $i$ ;  $n$  units  $\rightarrow n$  vertices in  $V$ .
- Edges in  $E$  are denoted by  $ij$ : at most  $\frac{n(n-1)}{2}$  edges.
- The cost of edge  $ij \in E$  is denoted by  $c_{ij} \geq 0$ .

- We require costs to satisfy the triangle inequality: for any distinct vertices  $i, j, k$ ,

$$c_{ij} \leq c_{ik} + c_{kj}.$$

- This holds if costs are distances, but other choices work too.
- Small cost  $c_{ij}$  if units  $i$  and  $j$  have similar values for block covariates.

# Nearest neighbor subgraph

- An edge  $ij$  is in the  $k$ -nearest neighbor subgraph if and only if

$$c_{ij} \leq c_{i(k)} \quad \text{or} \quad c_{ji} \leq c_{j(k)}.$$

- $c_{i(k)}$ :  $k^{\text{th}}$ -largest cost of edge connected to  $i$ .
- Well studied—most solutions in  $O(n \log n)$  time and  $O(n)$  space.

Depends on dimension of covariates, graph sparseness, and measure of dissimilarity.

# Optimal blocking as a graph partitioning problem

- A *partition* of  $V$  is a division of  $V$  into disjoint *blocks* of vertices  $\{V_1, V_2, \dots, V_\ell\}$ .
- Blocking of units  $\leftrightarrow$  Partition of a graph:  
Two experimental units are in the same block of the blocking if corresponding vertices are in the same block of the partition.

# Optimal blocking as a graph partitioning problem

- A *partition* of  $V$  is a division of  $V$  into disjoint *blocks* of vertices  $\{V_1, V_2, \dots, V_\ell\}$ .
- Blocking of units  $\leftrightarrow$  Partition of a graph:  
Two experimental units are in the same block of the blocking if corresponding vertices are in the same block of the partition.
- Approximately solve the *bottleneck threshold blocking problem*:  
Find a partition  $\{V_1^*, V_2^*, \dots, V_{\ell^*}^*\}$  with  $|V_j^*| \geq k$  and MWBC at most  $4\lambda$ .

# Approximate algorithm outline:

- Construct a  $(k - 1)$ -nearest neighbor subgraph.
- Select block seeds that are “just far enough apart.”
- Grow from these block centers to obtain an approximately optimal blocking.
- Approach extends from Hochbaum and Shmoys [1986].

# Algorithm step-by-step: Find nearest neighbor graph

- Construct a  $(k - 1)$ -nearest-neighbors graph
- Can show that edge costs are, at most,  $\lambda$ .

$$k = 2$$



# Algorithm step-by-step: Find block centers

- Find a set of vertices—*block seeds*—such that:
  - There is no path of two edges or less connecting any of the vertices in the set.
  - For any vertex not in the set, there is a path of two edges or less that connects that vertex to one in the set.
- Any set works, but some choices of seeds are better.
- Takes  $O(kn)$  time.





# Algorithm step-by-step: Grow from block centers

- Form blocks comprised of a block seed and any vertices adjacent to the seed.
- The way we choose seeds (no path of two edges connects two seeds), these blocks will not overlap.
- By nearest neighbors, these blocks contain at least  $k$  units.
- Takes  $O(n)$  time.



# Algorithm step-by-step: Assign all unassigned vertices

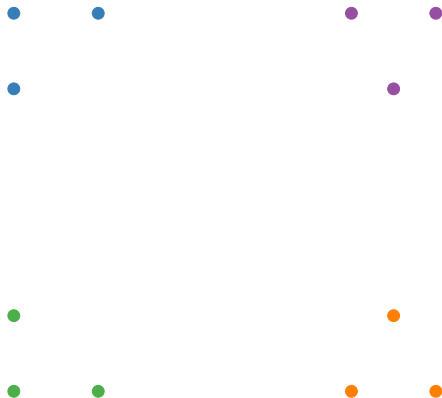
- For each unassigned vertex, find its closest seed in the nearest neighbor graph. Add that vertex to the seed's corresponding block.
- We choose seeds so that unassigned vertices are at most a path of two edges away from a block seed.
- Takes  $O(n)$  time.
- Since steps are sequential, total runtime is  $O(kn)$  outside of nearest neighbor graph construction.



# Our blocking

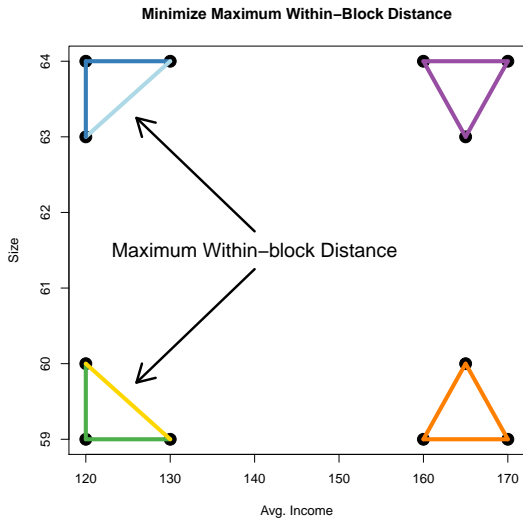
Our approximate algorithm came up with the following blocking:

Approximately optimal blocking



# A simple example:

Threshold  $k = 2$ . Dissimilarity = Mahalanobis distance.



# Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a blocking with  $MWBC \leq 4\lambda$ , though does much better than that in practice.

# Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a blocking with  $MWBC \leq 4\lambda$ , though does much better than that in practice.
- Sketch of proof:
- Each vertex is at most a path of two edges away from a block seed

$\implies$

Worst case: two vertices  $i, j$  in the same block can be connected by a path of four edges in the nearest neighbors graph:

Two from  $i$  to block seed, two from seed to  $j$ .

# Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a blocking with  $MWBC \leq 4\lambda$ , though does much better than that in practice.
- Sketch of proof:
- Each vertex is at most a path of two edges away from a block seed

$\implies$

Worst case: two vertices  $i, j$  in the same block can be connected by a path of four edges in the nearest neighbors graph:

Two from  $i$  to block seed, two from seed to  $j$ .

- Worst case: there are vertices  $l_1, l_2, l_3$  that form a path of 4 edges connecting  $i$  to  $j$ :

$$i l_1, l_1 l_2, l_2 l_3, l_3 j \quad (1)$$

# Sketch of proof

- Each edge has cost at most  $\lambda \implies$   
The corresponding edge costs satisfy:

$$c_{il_1} + c_{l_1l_2} + c_{l_2l_3} + c_{l_3j} \leq 4\lambda.$$



# Sketch of proof

- Each edge has cost at most  $\lambda \implies$   
The corresponding edge costs satisfy:

$$c_{il_1} + c_{l_1l_2} + c_{l_2l_3} + c_{l_3j} \leq 4\lambda.$$

- Since, edge costs satisfy the triangle inequality:

$$c_{ij} \leq c_{il_1} + c_{l_1l_2} + c_{l_2l_3} + c_{l_3j} \leq 4\lambda.$$

- That is, every edge joining two vertices within the same block has cost  $\leq 4\lambda$ .
- Hence, MWBC of the approximately optimal blocking is  $\leq 4\lambda$ .
- QED

# Heuristic Improvements:

Some quick adjustments can improve performance of algorithm:

- Use a directed nearest neighbor graph:
  - Improve sparseness  $\implies$  Smaller block sizes
- Heuristics for improving selection of block seeds.
- Subdivide blocks with more than  $2k$  units
- Local search (e.g. Kernighan–Lin)

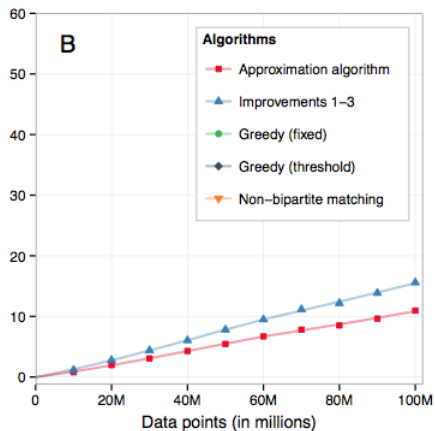
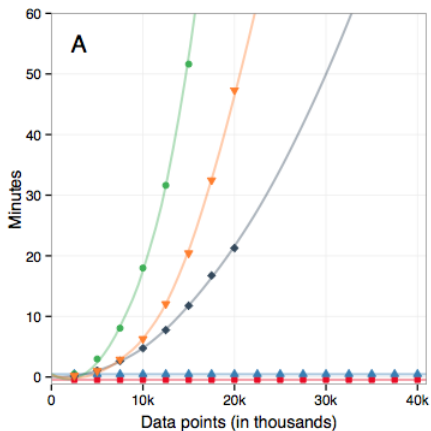
# Results: Simulation study

Simulation study to compare our method to current methods.

- Repeatedly simulate covariates  $x_1, x_2 \sim \text{Uniform}(0, 10)$
- Compare our blocking method (with and without heuristic improvements) when  $k = 2$  to commonly used implementations of greedy blocking and non-bipartite matching.
- Sample sizes from 100 units to 100 million units: Our method with improvements **blocks 100 million units in about 17 minutes.**

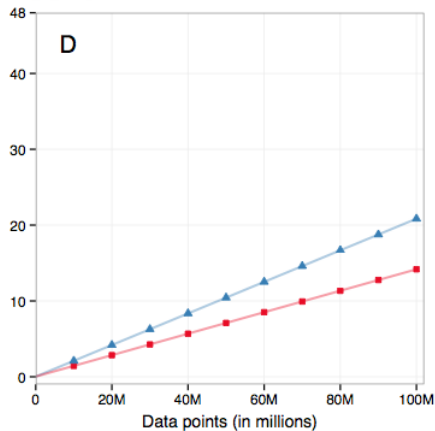
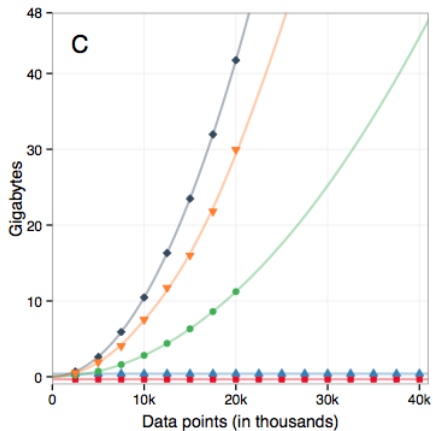
# Time and space comparison

Run time in minutes



# Time and space comparison

## Memory usage in gigabytes



# Results: Simulation study

- We consider the response schedule:

$$y_i = x_{1i}x_{2i} + \epsilon_i, \quad \epsilon_i \sim N(0, 1)$$

- Compare our blocking method to implementations of greedy blocking and non-bipartite matching, and estimates using complete randomization.
- Elaborate simulation study currently in progress.

# Results: RMSE

| Method                  | $10^2$ | $10^3$ | $10^4$ |
|-------------------------|--------|--------|--------|
| Approximation algorithm | 1.000  | 1.000  | 1.000  |
| Improvements            | 0.931  | 0.960  | 0.992  |
| Fixed greedy            | 1.609  | 1.598  | 1.152  |
| Threshold greedy        | 1.207  | 1.146  | 1.041  |
| Non-bipartite matching  | 0.952  | 0.949  | 0.983  |
| Unadjusted              | 6.092  | 15.158 | 20.710 |
| OLS adjustment          | 2.352  | 5.776  | 7.900  |

Table: RMSE relative to approximation algorithm by sample size

# Future Work

- Extend method to other statistical problems
  - Post-stratification—alternative to coarsened exact matching.
  - Clustering—alternative to  $k$ -means.
- Algorithm improvements:
  - Decrease runtime and improve performance of the algorithm.
  - Under what circumstances can factor of 4 be improved?
- Software coming soon.



# Bibliography I

- D.S. Hochbaum and D.B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. Journal of the ACM (JACM), 33(3):533–550, 1986.
- Paul W Holland. Statistics and causal inference. Journal of the American statistical Association, 81(396):945–960, 1986.
- David G. Kirkpatrick and Pavol Hell. On the complexity of general graph factor problems. SIAM Journal on Computing, 12(3):601–609, 1983.
- Kari Lock Morgan and Donald B Rubin. Rerandomization to improve covariate balance in experiments. Annals of Statistics, 40(2):1263–1282, 2012.
- Luke W. Miratrix, Jasjeet S. Sekhon, and Bin Yu. Adjusting treatment effect estimates by post-stratification in randomized experiments. Journal of the Royal Statistical Society, Series B, 75(2):369–396, 2013.

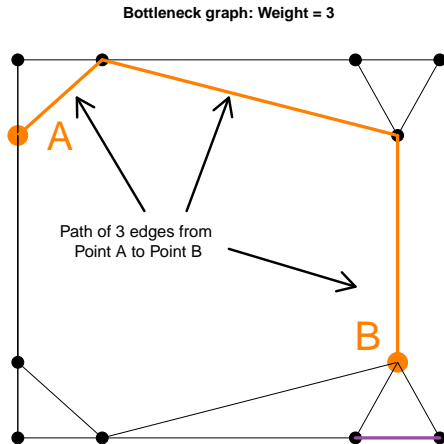
# Bibliography II

- P.R. Rosenbaum. Optimal matching for observational studies. Journal of the American Statistical Association, 84(408):1024–1032, 1989.
- Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of Educational Psychology; Journal of Educational Psychology, 66(5):688, 1974.
- Jerzy Splawa-Neyman, DM Dabrowska, and TP Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. Statistical Science, 5(4):465–472, 1990.

Thank you.

# Bottleneck subgraph: In pictures

$B_3(G)$ : Bottleneck subgraph of weight 3



# Estimating treatment effects: Notation

- Recall, there are  $n$  units and  $r$  treatment categories.
- There are  $b$  blocks, with  $n_c$  units within each block  $c = 1, \dots, b$ .
- Units within each block are ordered in some way, let  $(k, c)$  denote the  $k$ th unit in block  $c$ .

For tractability:

- Assume treatment assignment is *balanced* within each block: Each treatment is replicated the same number of times (up to remainder).
- Assume  $r$  divides each  $n_c$ .

# Neyman-Rubin potential outcomes model

- We assume responses follow the Neyman-Rubin potential outcomes model [Splawa-Neyman et al., 1990, Rubin, 1974, Holland, 1986]:

$$Y_{kc} = y_{kc1} T_{kc1} + y_{kc2} T_{kc2} + \dots + y_{kcr} T_{kcr}.$$

- Frequently used in causal inference.

# Neyman-Rubin potential outcomes model

- We assume responses follow the Neyman-Rubin potential outcomes model [Splawa-Neyman et al., 1990, Rubin, 1974, Holland, 1986]:

$$Y_{kc} = y_{kc1} T_{kc1} + y_{kc2} T_{kc2} + \dots + y_{kcr} T_{kcr}.$$

- Frequently used in causal inference.
- $T_{kcs}$  denotes a treatment indicator:  $T_{kcs} = 1$  if unit  $(k, c)$  receives treatment  $s$ ; otherwise,  $T_{kcs} = 0$ .
- $y_{kcs}$  denotes the *potential outcome* for unit  $(k, c)$  under treatment  $s$ —the response of  $(k, c)$  we would observe had that unit received treatment  $s$ . Potential outcomes are non-random, and  $y_{kcs}$  is unknown unless unit  $(k, c)$  receives treatment  $s$ .
- $Y_{kc}$  denotes the observed response of  $(k, c)$ . Randomness of  $Y_{kc}$  due entirely to randomness in treatment assignment.

# Neyman-Rubin potential outcomes model

- We assume responses follow the Neyman-Rubin potential outcomes model [Splawa-Neyman et al., 1990, Rubin, 1974, Holland, 1986]:

$$Y_{kc} = y_{kc1} T_{kc1} + y_{kc2} T_{kc2} + \dots + y_{kcr} T_{kcr}.$$

Example:

- Medical trial: Suppose testing a experimental procedure that may improve health outcomes.
- $T_{kc1}, T_{kc2}$ : Indicates whether the patient receives/does not receive the procedure.
- $y_{kc1}, y_{kc2}$ : Whether the patient is alive five years from today if the patient receives/does not receive the procedure.
- $Y_{kc}$ : Whether the patient is alive five years from today.



# Neyman-Rubin potential outcomes model

- We assume responses follow the Neyman-Rubin potential outcomes model [Splawa-Neyman et al., 1990, Rubin, 1974, Holland, 1986]:

$$Y_{kc} = y_{kc1} T_{kc1} + y_{kc2} T_{kc2} + \dots + y_{kcr} T_{kcr}.$$

- Model makes the *stable-unit treatment value assumption* (SUTVA): the observed  $Y_{kc}$  only depends on which treatment is assigned to unit  $(k, c)$ , and is not affected by the treatment assignment of any other unit  $(k', c')$ .

# Parameter of interest and estimators

- Parameter of interest: Sample average treatment effect of treatment  $s$  relative to treatment  $t$  ( $\text{SATE}_{st}$ ):

$$\text{SATE}_{st} = \sum_{c=1}^b \sum_{k=1}^{n_c} \frac{y_{kcs} - y_{kct}}{n}$$

- Two unbiased estimators of  $\text{SATE}_{st}$  are the difference-in-means estimator and the the Horvitz-Thompson estimator.

$$\hat{\delta}_{st,\text{diff}} \equiv \sum_{c=1}^b \frac{n_c}{n} \sum_{k=1}^{n_c} \left( \frac{y_{kcs} T_{kcs}}{\# T_{cs}} - \frac{y_{kct} T_{kct}}{\# T_{ct}} \right),$$

$$\hat{\delta}_{st,\text{HT}} \equiv \sum_{c=1}^b \frac{n_c}{n} \sum_{k=1}^{n_c} \left( \frac{y_{kcs} T_{kcs}}{n_c/r} - \frac{y_{kct} T_{kct}}{n_c/r} \right).$$

- These estimators are the same when treatment assignment is balanced and  $r$  divides each  $n_c$ .

# Variance of estimators

$$\begin{aligned}\text{Var}(\hat{\delta}_{st,\text{diff}}) &= \text{Var}(\hat{\delta}_{st,\text{HT}}) \\ &= \sum_{c=1}^b \frac{n_c^2}{n^2} \left( \frac{r-1}{n_c-1} (\sigma_{cs}^2 + \sigma_{ct}^2) + 2 \frac{\gamma_{cst}}{n_c-1} \right) \\ \mu_{cs} &= \frac{1}{n_c} \sum_{k=1}^{n_c} y_{kcs} \\ \sigma_{cs}^2 &= \frac{1}{n_c} \sum_{k=1}^{n_c} (y_{kcs} - \mu_{cs})^2 \\ \gamma_{cst} &= \frac{1}{n_c} \sum_{k=1}^{n_c} (y_{kcs} - \mu_{cs})(y_{kct} - \mu_{ct})\end{aligned}$$

Small differences in formulas for more general treatment assignments.  
Diff-in-means tends to have smaller variance when block sizes are small.

# Variance of estimators

$$\begin{aligned}\text{Var}(\hat{\delta}_{st,\text{diff}}) &= \text{Var}(\hat{\delta}_{st,\text{HT}}) \\ &= \sum_{c=1}^b \frac{n_c^2}{n^2} \left( \frac{r-1}{n_c-1} (\sigma_{cs}^2 + \sigma_{ct}^2) + 2 \frac{\gamma_{cst}}{n_c-1} \right)\end{aligned}$$

- Note:  $\sigma_{cs}^2$  and  $\sigma_{ct}^2$  are estimable,  $\gamma_{cst}$  not directly estimable.

# Variance of estimators

$$\begin{aligned}\text{Var}(\hat{\delta}_{st,\text{diff}}) &= \text{Var}(\hat{\delta}_{st,\text{HT}}) \\ &= \sum_{c=1}^b \frac{n_c^2}{n^2} \left( \frac{r-1}{n_c-1} (\sigma_{cs}^2 + \sigma_{ct}^2) + 2 \frac{\gamma_{cst}}{n_c-1} \right)\end{aligned}$$

- Note:  $\sigma_{cs}^2$  and  $\sigma_{ct}^2$  are estimable,  $\gamma_{cst}$  not directly estimable.
- Conservative estimate:

$$\widehat{\text{Var}} = \sum_{c=1}^b \frac{n_c^2}{n^2} \left( \frac{r}{n_c-1} (\hat{\sigma}_{cs}^2 + \hat{\sigma}_{ct}^2) \right)$$

# When does blocking help?

- Blocking vs. completely randomized treatment assignment (no blocking): which estimates of  $SATE_{st}$  have lower variance?
- Can show that blocking helps if and only if:

$$\sum_{c=1}^b n_c^2 \left[ \left( \frac{(r-1)(\sigma_s^2 + \sigma_t^2) + 2\gamma_{st}}{\sum n_c^2(n-1)} \right) - \left( \frac{(r-1)(\sigma_{cs}^2 + \sigma_{ct}^2) + 2\gamma_{cst}}{n^2(n_c-1)} \right) \right] \geq 0$$

- Intuitive to make block-level variances small w.r.t. domain-level variances, but other blocking designs may also improve treatment effect estimates.

# Can blocking hurt?

- When assignments of units to blocks completely randomized:

$$\begin{aligned} & \mathbb{E} \left[ \sum_{c=1}^b n_c^2 \left( \frac{(r-1)(\sigma_{cs}^2 + \sigma_{ct}^2) + 2\gamma_{cst}}{n^2(n_c - 1)} \right) \right] \\ &= \sum_{c=1}^b n_c^2 \left( \frac{(r-1)(\sigma_s^2 + \sigma_t^2) + 2\gamma_{st}}{\sum n_c^2(n-1)} \right) \end{aligned}$$

Expected variance when blocking = Completely randomized variance

- Blocking better than “at random” → Reduced variance in treatment effect estimates.

# Results: A toy example

Consider the following toy example:

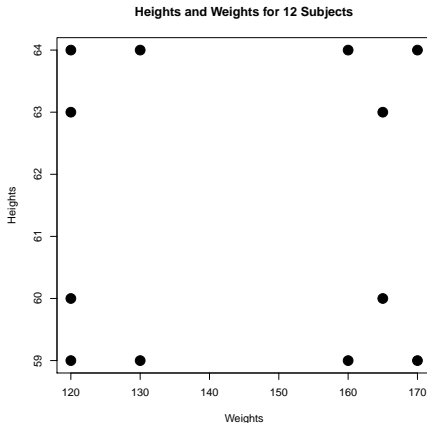
- “Health score” is a variable that is well-known to be affected by a person’s height and weight.
- Scientists claim that taking a vitamin will improve the health score.
- Unbeknownst to the researchers, the true relationship between height, weight, and vitamin intake on health score is:

$$\text{Health score}_i = 3(\text{height}_i) + \text{weight}_i + 1.5\sqrt{(\text{height}_i)(\text{weight}_i)} + 50(\text{takeVitamin}_i)$$



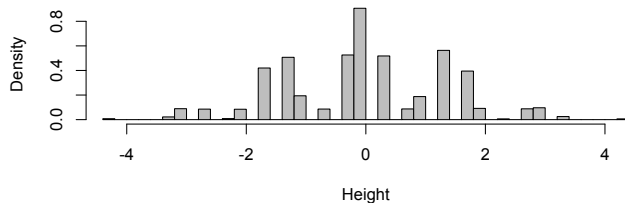
# Results: A toy example

- Suppose the scientists are able to perform an experiment on 12 subjects to determine the effect of the vitamin.
- We analyze results of this experiment when blocking on height and weight using our blocking method ( $t^* = 2$ , Mahalanobis distance) and when completely randomizing treatment.
- Compare both covariate balance and precision of treatment effect estimates.

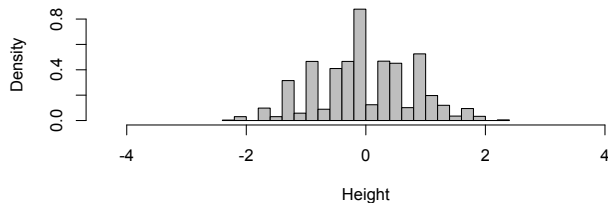


# Covariate balance: Height

**Histogram of height for completely randomized treatment**  
**SD = 1.36**

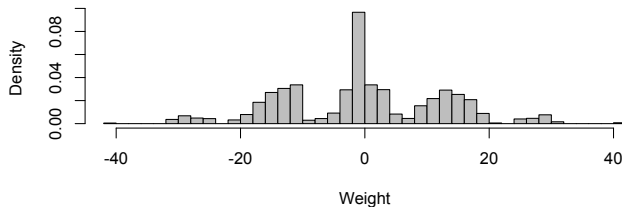


**Histogram of height for block randomized treatment**  
**SD = 0.80**

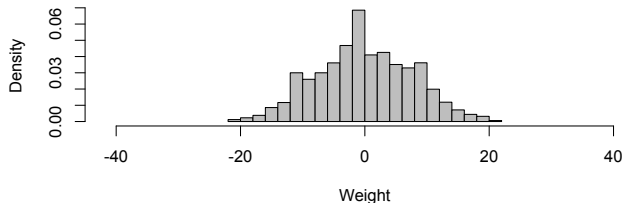


# Covariate balance: Weight

**Histogram of weight for completely randomized treatment**  
**SD = 12.82**

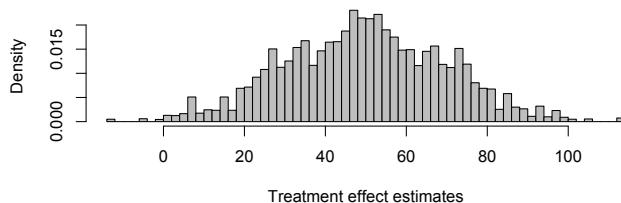


**Histogram of weight for completely randomized treatment**  
**SD = 7.65**

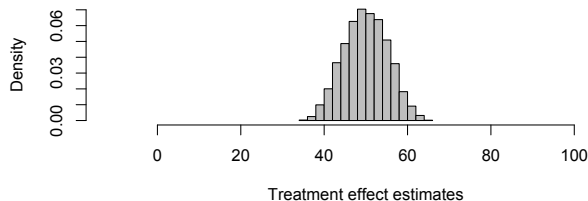


# Comparison of estimates

**Treatment effect estimates for completely randomized treatment: SD = 19.94**



**Diff-in-means treatment effect estimates for block randomized treatment: SD = 5.16**



# Results: Comparison of estimates

- For this toy example, our blocking method dramatically reduces the potential for large covariate imbalance.
- Blocking yields a much more precise estimate of the treatment effect.

