Optimal Blocking by Minimizing the Maximum Within-Block Distance

Michael J. Higgins
Jasjeet Sekhon

Princeton University
University of California at Berkeley

November 14, 2013

For the Kansas State University Statistics Seminar

Goal

Our goal: Devise a method for blocking for experiments with multiple treatment categories (and possibly multiple replications of each treatment within each block) that ensures good covariate balance between treatment groups in small experiments and is efficient enough to be used in large experiments.

Covariate imbalance in randomized experiments

Randomized controlled experiment with n units and r treatment categories.

- PROBLEM:
  There may be a 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 $\longrightarrow$ Inaccurate estimates of treatment effects.
- Problem is most likely to happen in small experiments; critical to avoid if cost of additional units is large (e.g. a medical trial).

Post-randomization solutions to covariate imbalance

- Post-stratification [Miratrix et al., 2013]:
  - *After* running the experment, group similar units together when analyzing data to increase precision of treatment effect estimates.
  - Data mining concerns.
  - May not be most efficient method for small experiments.
- 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 initial balance is bad.
  - Standard errors calculated across only "valid" randomizations. These calculations can be complex—even when initial balance is good!
- Moral of the story: Preferable to design the randomization scheme to avoid covariate imbalance.

One solution: Blocking

- Blocking can reduce variance of treatment effect estimates *AND* can ensure that randomization preserves covariate balance.
- *Blocking*—grouping units according to common traits (e.g. same political party, similar health) before treatment is assigned.
- Treatment is completely randomized within each block, and is assigned independently across blocks.
- Traits used to form blocks are called *block covariates*.

Matched-pairs: Current method

Matched-pairs blocking: Pair units with "most-similar" block covariates. For each pair, randomly assign one unit to treatment, one to control.

Shortcomings:

- No efficient way to extend method to cases where more than two units within a block.
- Fixed block sizes: If units form clusters, matched-pairs design may pair units from different clusters.
- Even with two treatments, having blocks of more than two units may be helpful in estimating standard errors [Imbens, 2011].

Our approach: Mimimize the MWBD

We analyze the following blocking method:

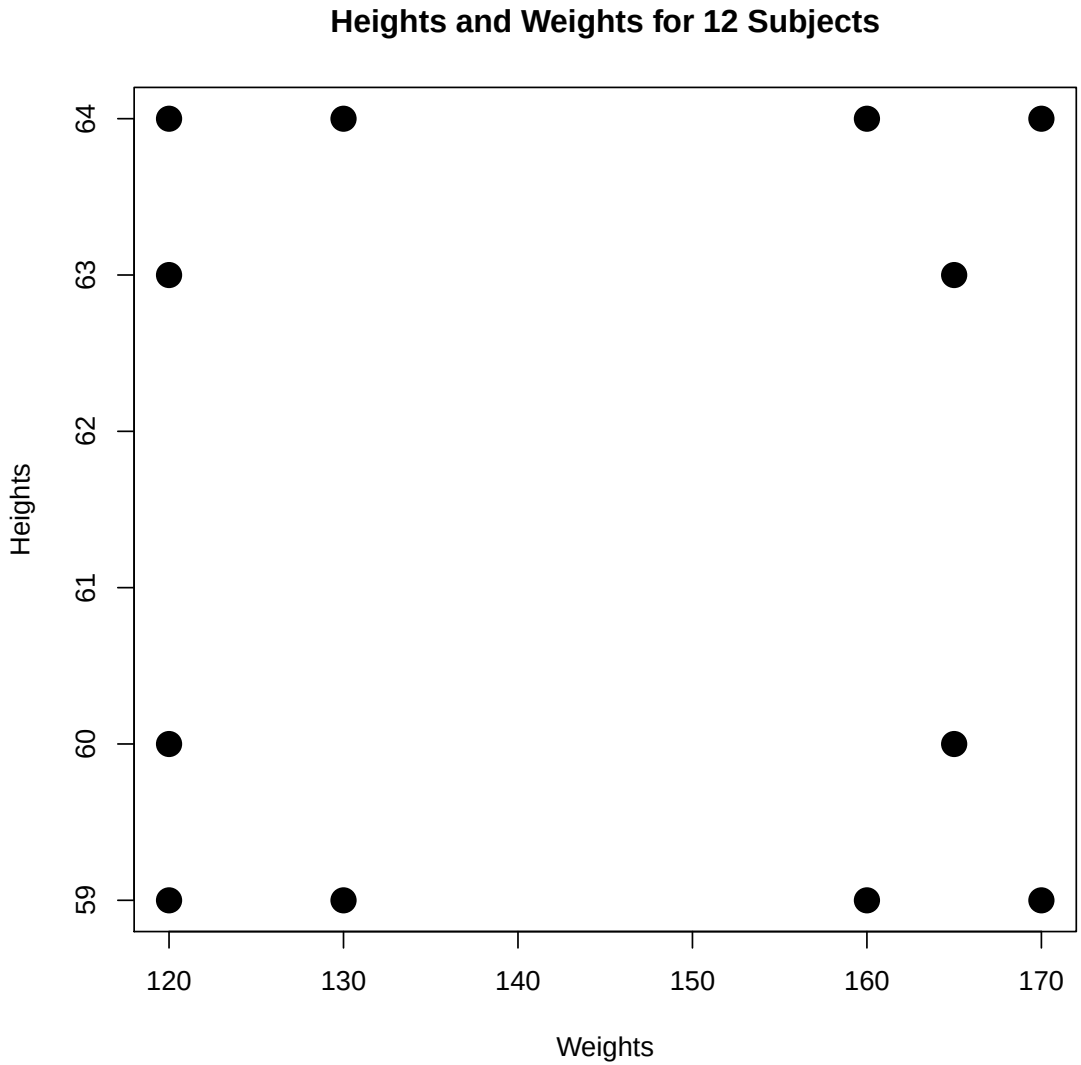
1. Choose a measure of dissimilarity or distance (e.g. Mahalanobis, standardized Euclidian) that is small when important covariates have similar values.
2. Choose a threshold $t^* \geq r$ for the minimum number of units to be contained in a block.
3. Block units so that each block contains at least t^* units, and so that the maximum distance between any two units within a block—the maximum within-block distance (MWBD)—is minimized.

Blocking by minimizing the MWBD

- Block units so that each block contains at least t^* units, and so that the maximum distance between any two units within a block—the maximum within-block distance (MWBD)—is minimized.
- Minimizing the MWBD: Can ensure covariate balance in randomization.
- Threshold t^*: Allows designs with multiple treatment categories, multiple replications of treatments within a block; blocks can preserve clustering in data.
- "Good" blocking can be found in polynomial time: applicable to large experiments.

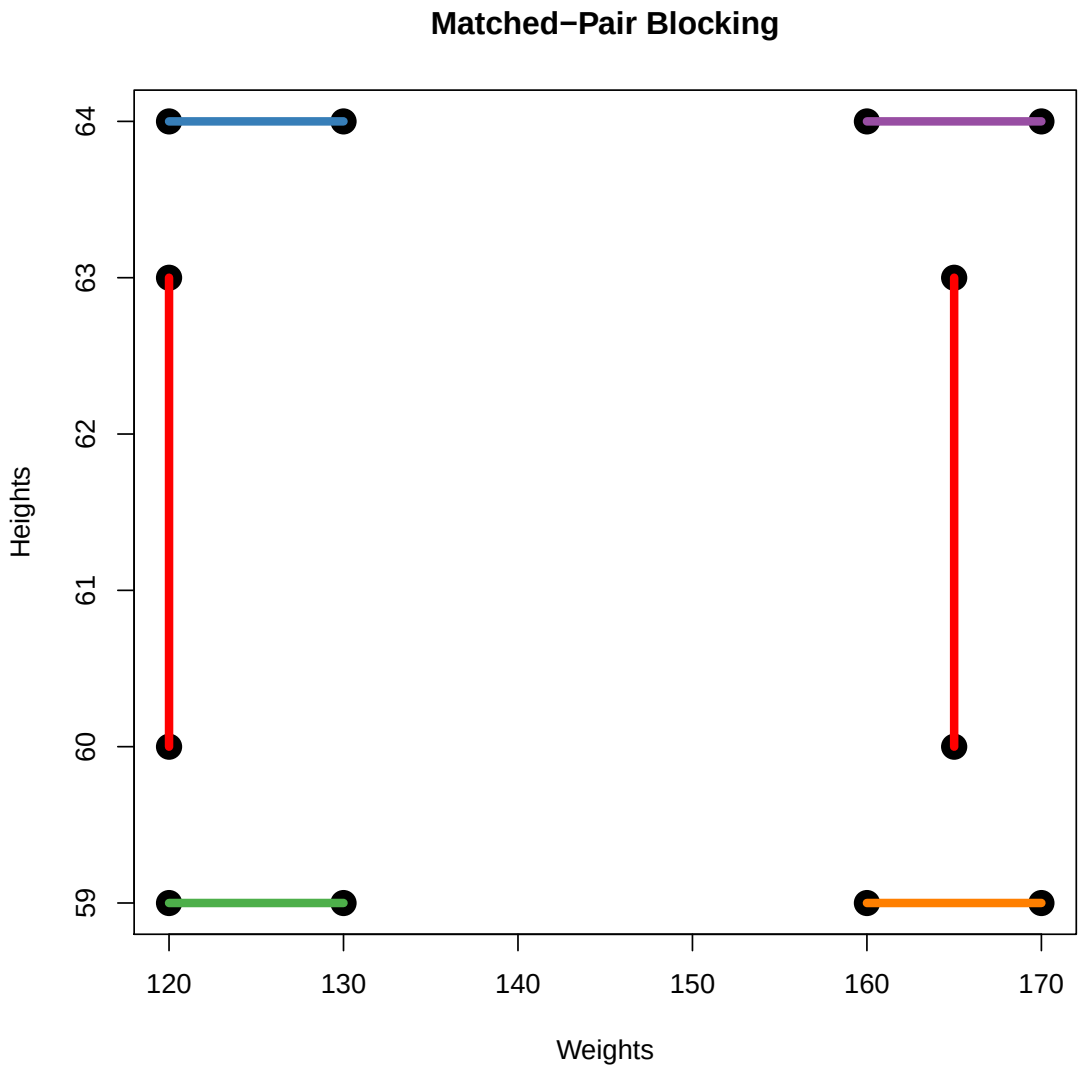
A simple example:

Threshold $t^* = 2$. Distance = Mahalanobis distance.



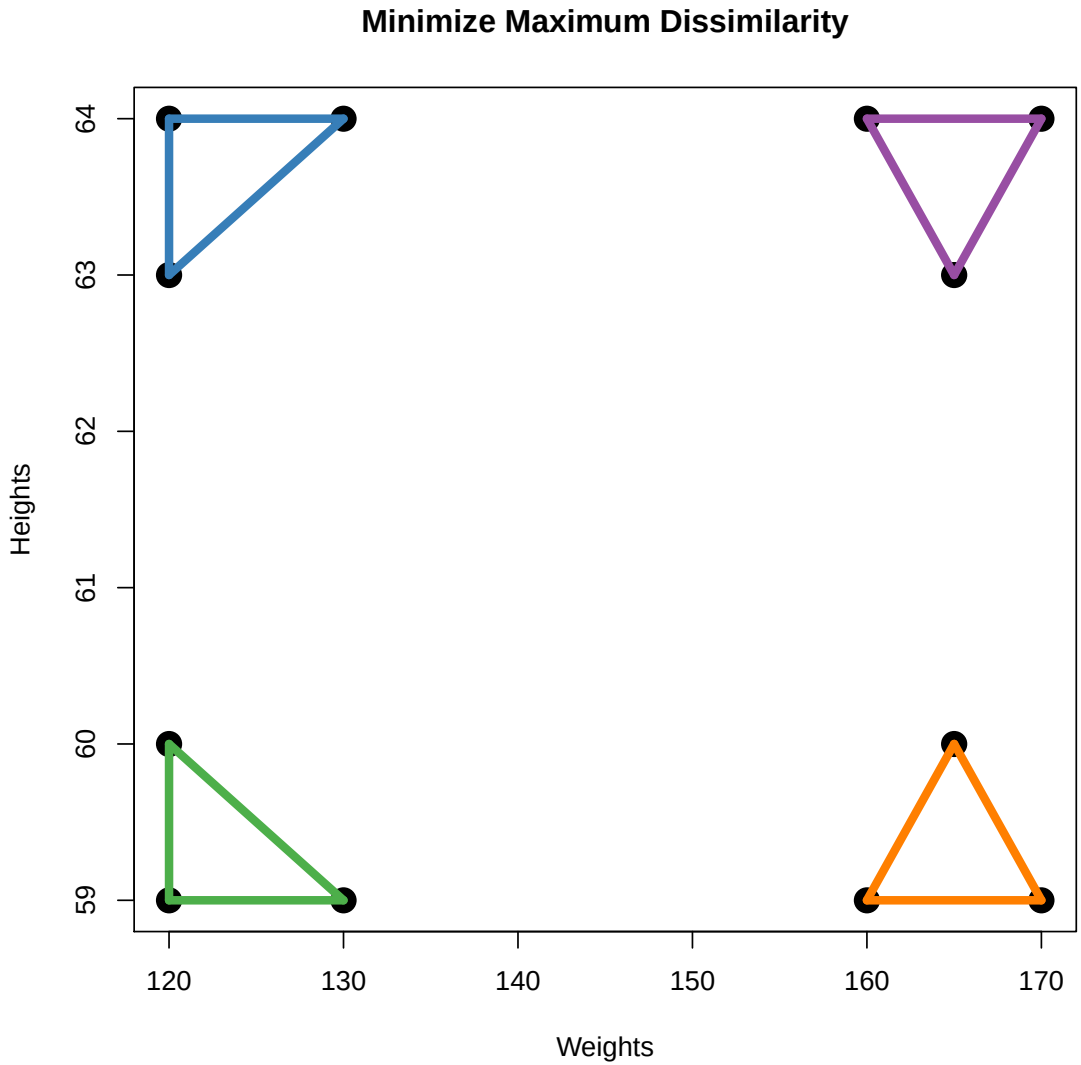
A simple example:

Threshold $t^* = 2$. Dissimilarity = Mahalanobis distance.



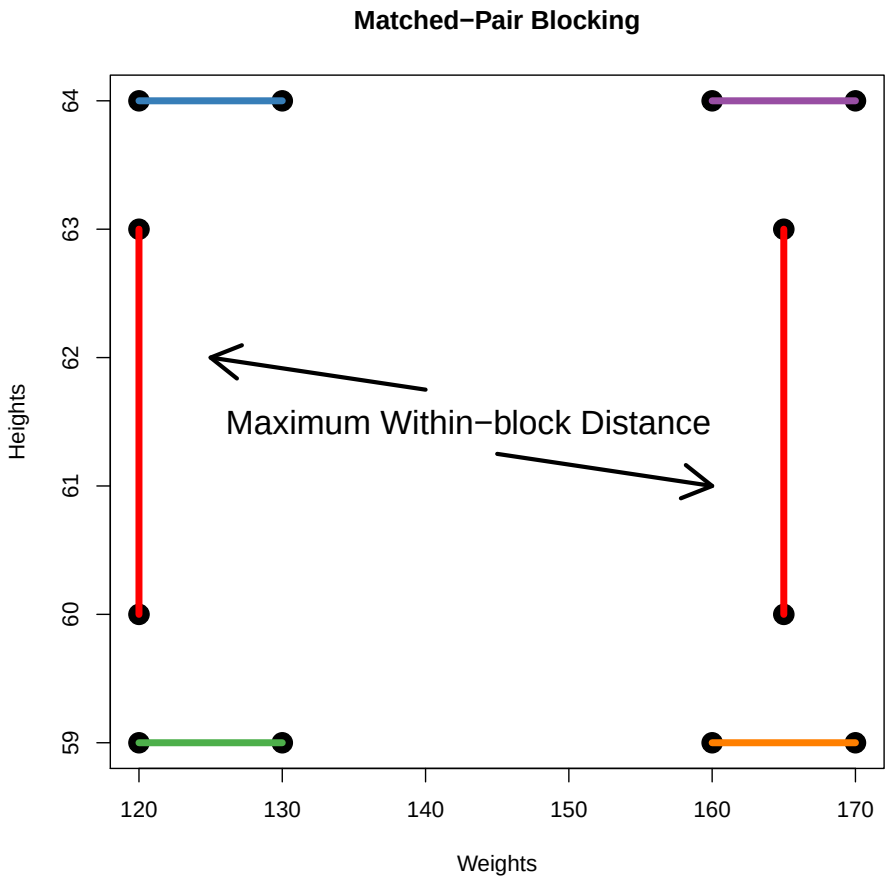
A simple example:

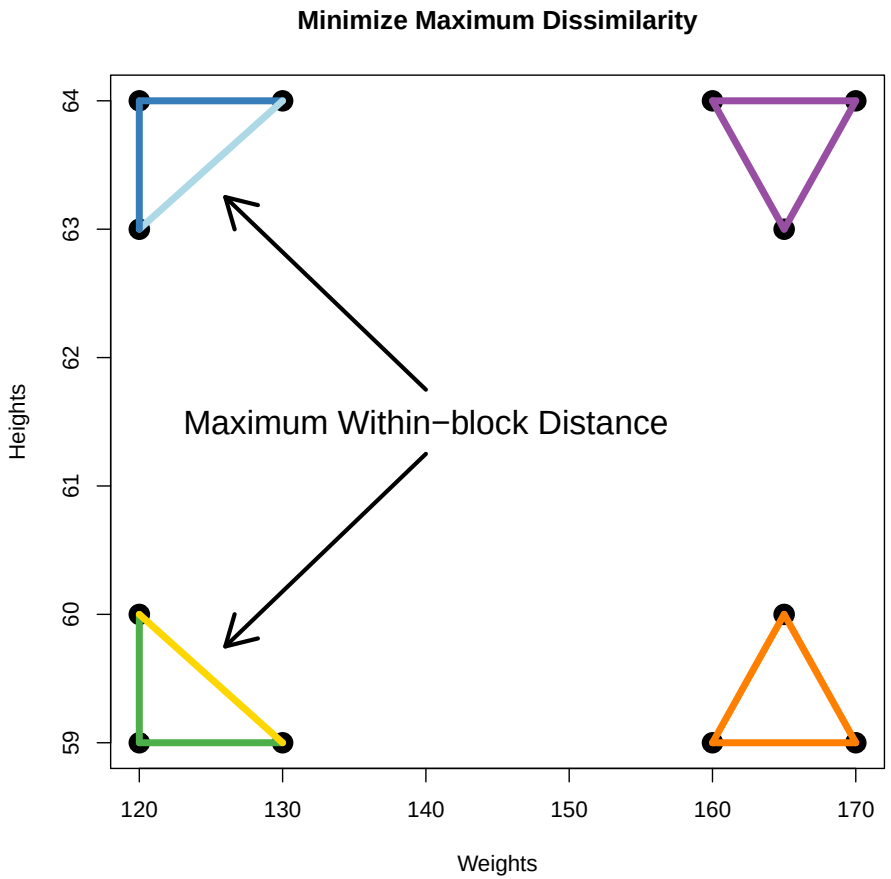
Threshold $t^* = 2$. Distance = Mahalanobis distance.



A simple example:

Threshold $t^* = 2$. Dissimilarity = Mahalanobis distance.





Optimal blocking and approximately optimal blocking

For all blockings that contain at least t^* units with each block:

- Let λ denote the smallest MWBD achievable by such a blocking—any blocking that meets this bound is called an *optimal blocking*.
- Finding an optimal blocking is an NP-hard problem—feasible to find in small experiments, may not be feasible in large experiments [Hochbaum and Shmoys, 1986].

Optimal blocking and approximately optimal blocking

For all blockings that contain at least t^* units with each block:

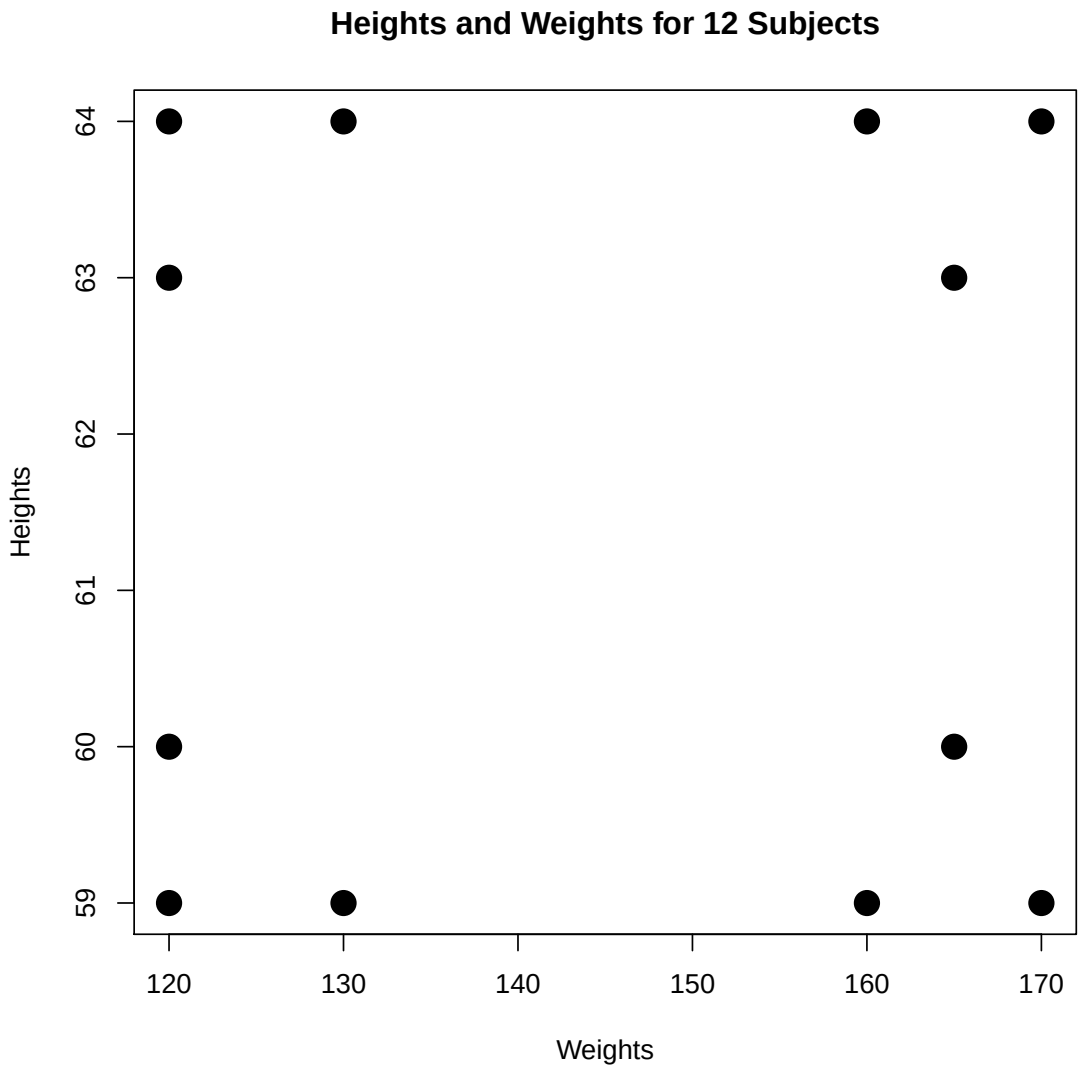
- Let λ denote the smallest MWBD achievable by such a blocking—any blocking that meets this bound is called an *optimal blocking*.
- Finding an optimal blocking is an NP-hard problem—feasible to find in small experiments, may not be feasible in large experiments [Hochbaum and Shmoys, 1986].
- We show that finding a blocking with MWBD $\leq 4\lambda$ is possible in polynomial time—finds a "good" blocking when the number of units is small *or large*.
- Denote any such blocking as an *approximately optimal blocking*.

Viewing experimental units as a graph

- Extending an idea from Paul Rosenbaum [1989]: Statistical blocking problems can be viewed as graph theory partitioning problems.
- Experimental units are vertices in a graph.
- An edge is drawn between two units if they can be placed in the same block.
- Edge-weights are a measure of distance between pretreatment covariates (e.g. Mahalanobis, standardized Euclidian).
- Use methods in graph theory to solve original blocking problem.

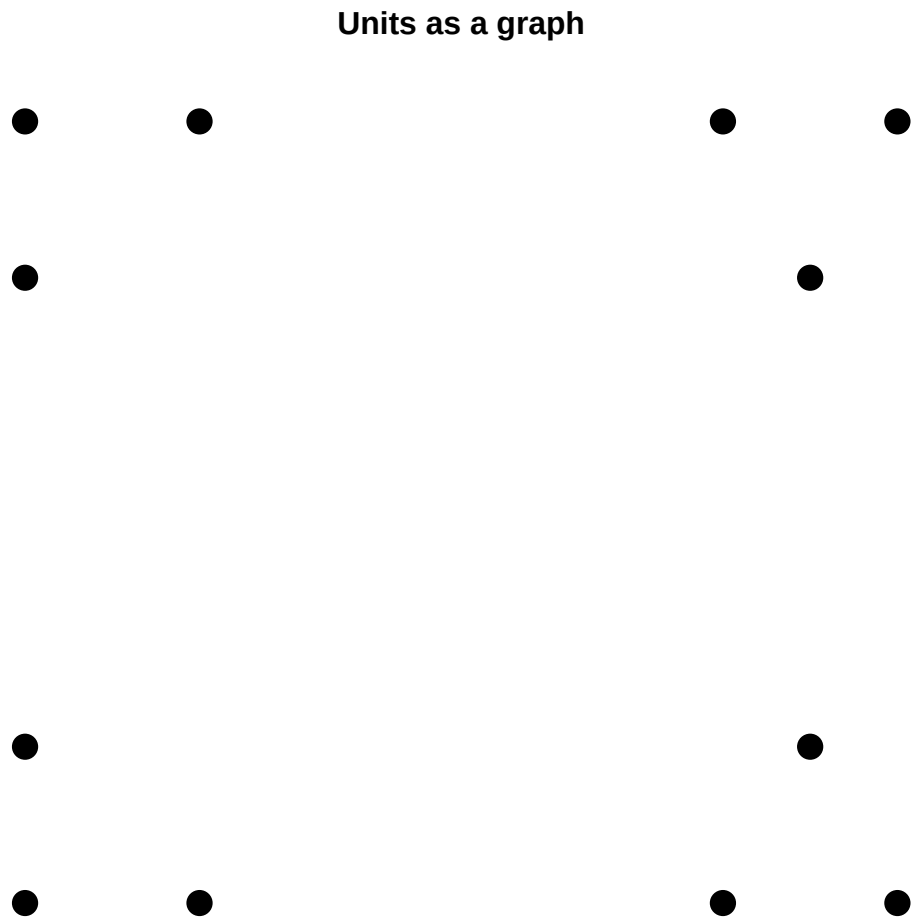
Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.



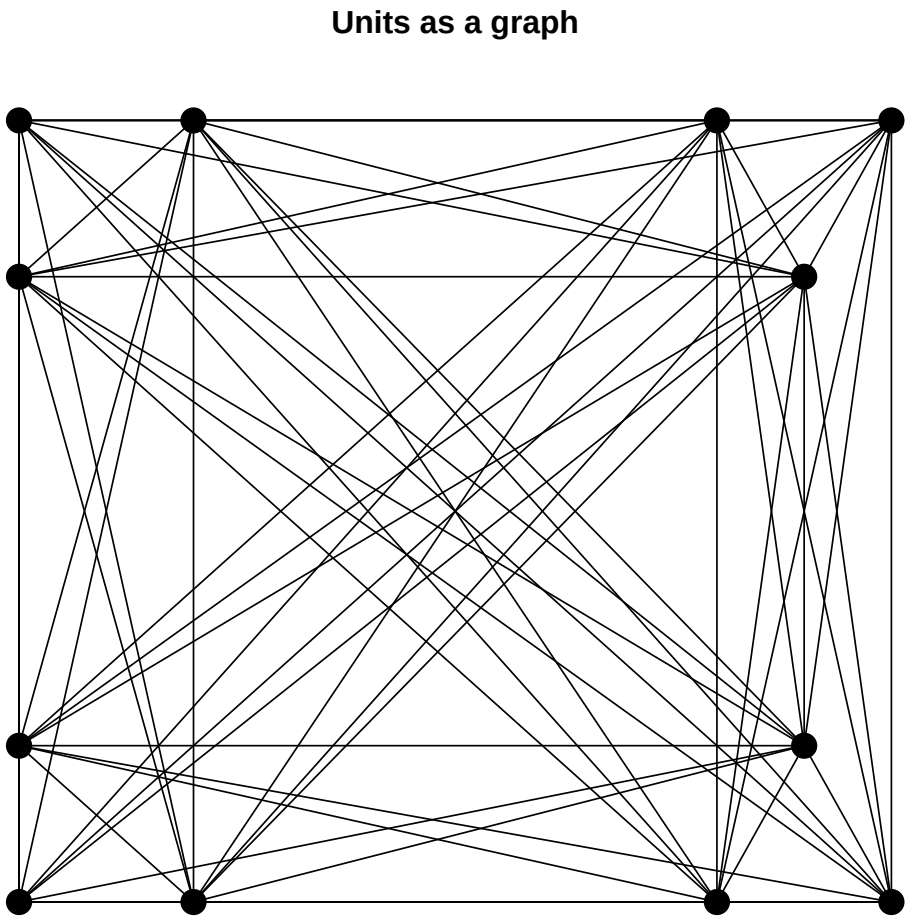
Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.



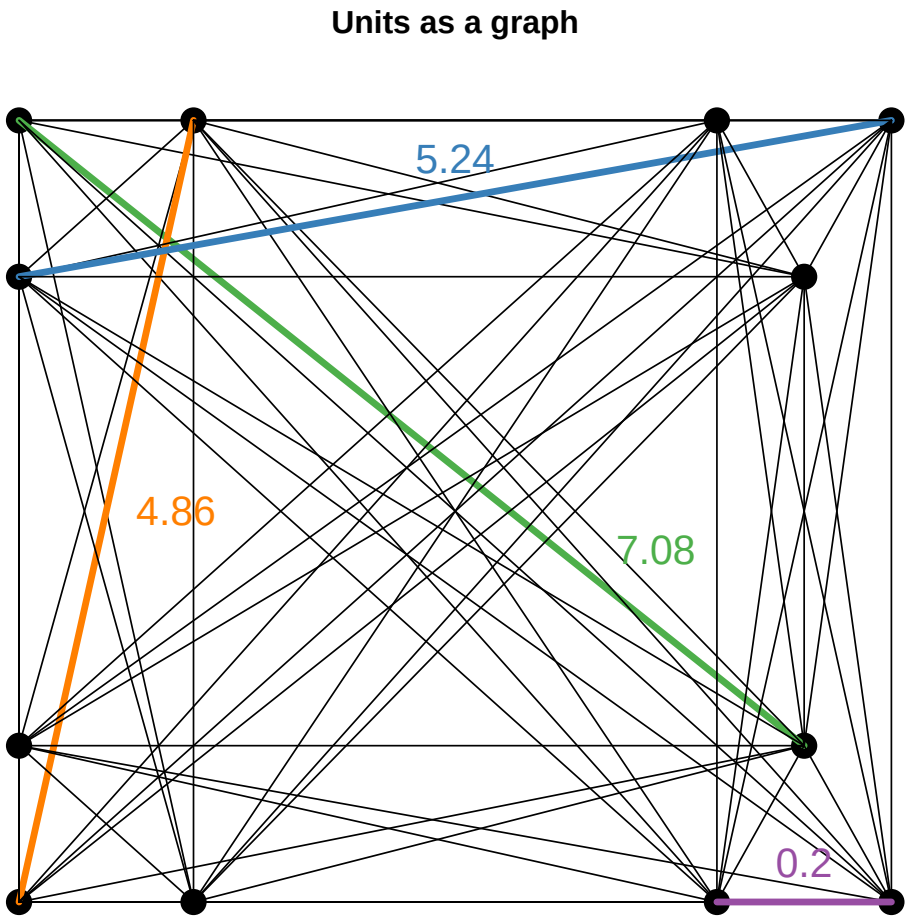
Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.



Viewing experimental units as a graph: In pictures

Dissimilarity = Mahalanobis distance.



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 $\longrightarrow$ n vertices in V.
- Edges in E are denoted by (i, j): at most $\frac{n(n-1)}{2}$ edges.
- The weight of edge $(i, j) \in E$ is denoted by w_{ij}: at most $\frac{n(n-1)}{2}$ distinct values of w_{ij}.

Note about edge weights

- We require weights to satisfy the triangle inequality: for any distinct vertices $\{i\}, \{j\}, \{k\}$,

$$w_{ik} + w_{kj} \geq w_{ij}.$$

- This holds if weights are distances, but other choices work too.
- What's important: Small weight w_{ij} if units i and j have similar values for block covariates.

Optimal blocking as a graph partitioning problem

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

Optimal blocking as a graph partitioning problem

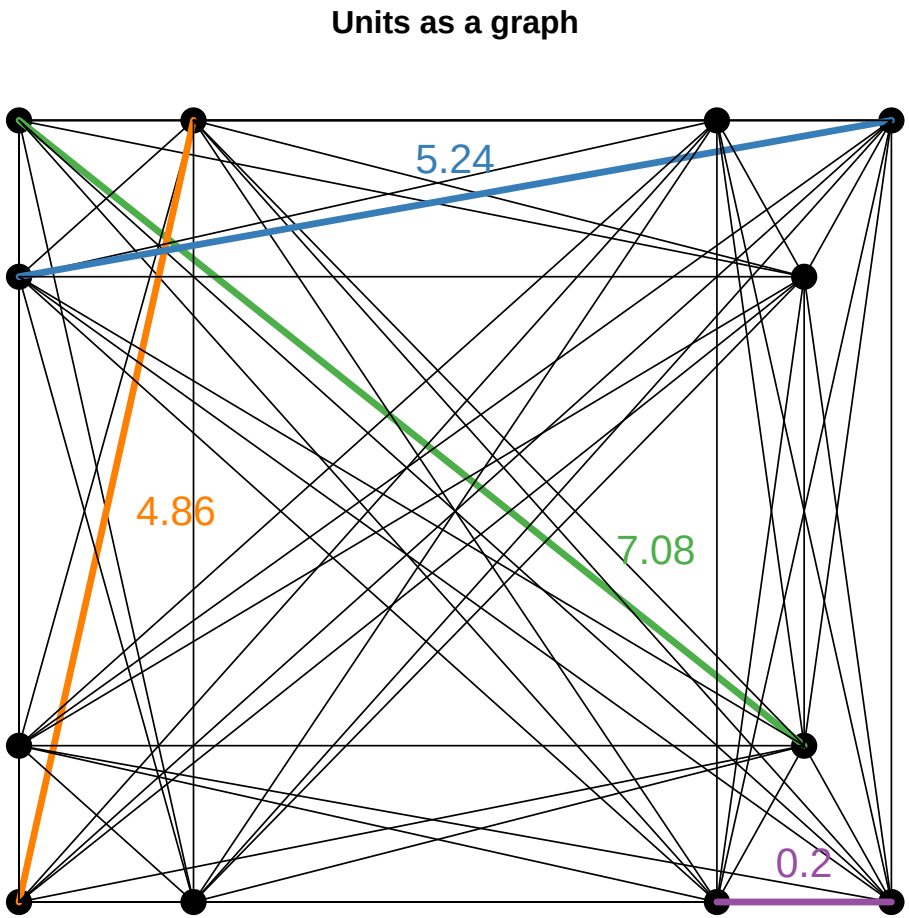
- A *partition* of V is a division of V into disjoint *blocks* of vertices $(V_1, V_2, \ldots, V_\ell)$.
- Blocking of units $\leftrightarrow$ Partition of a graph:
  Two experimental units are in the same block of the blocking if and only if their corresponding vertices are in the same block of the partition.
- Optimal blocking problems are optimal partitioning problems: we want to find a partition $(V_1^*, V_2^*, \ldots, V_{\ell^*}^*)$ with $|V_j^*| \geq t^*$ that minimizes the maximum within-block edge weight.

Bottleneck subgraphs

- Bottleneck subgraphs are a useful tool for solving graph partitioning problems.
- Define the *bottleneck subgraph of G of weight threshold w* as the graph that has an edge between vertices $\{i\}$ and $\{j\}$ if and only if $w_{ij} \leq w$.
- All points within a block of our approximately optimal blocking are connected by some path of edges in a bottleneck subgraph; used to show approximate optimality.

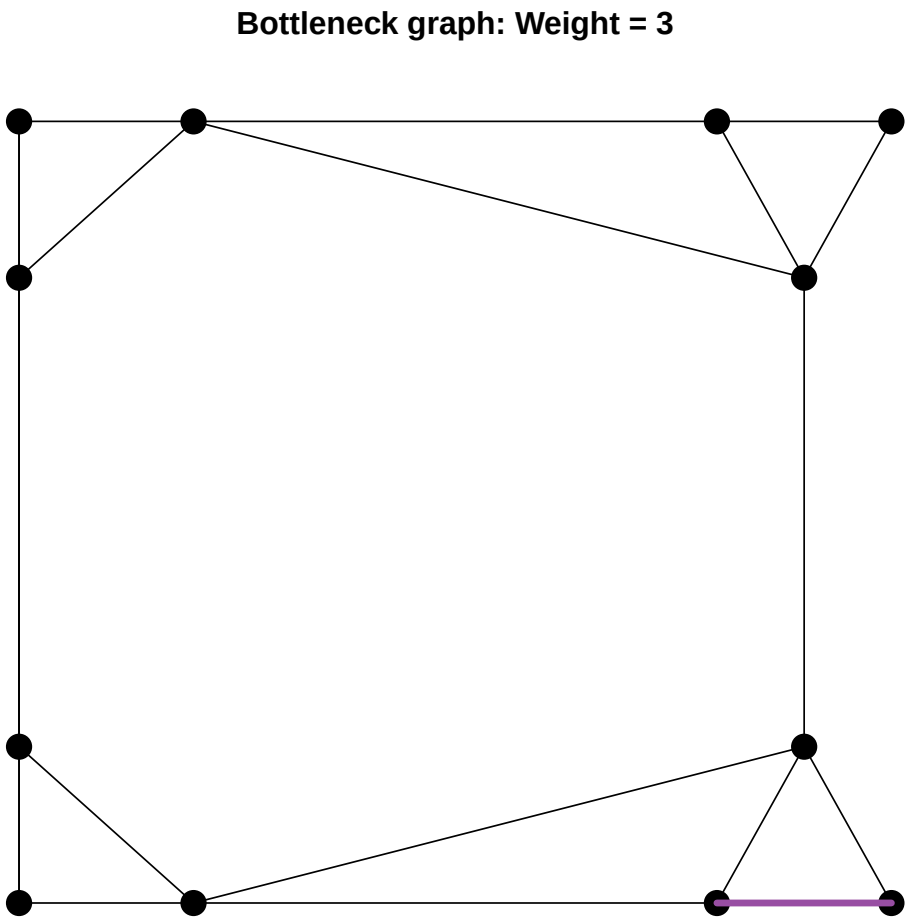
Bottleneck subgraph: In pictures

All edges



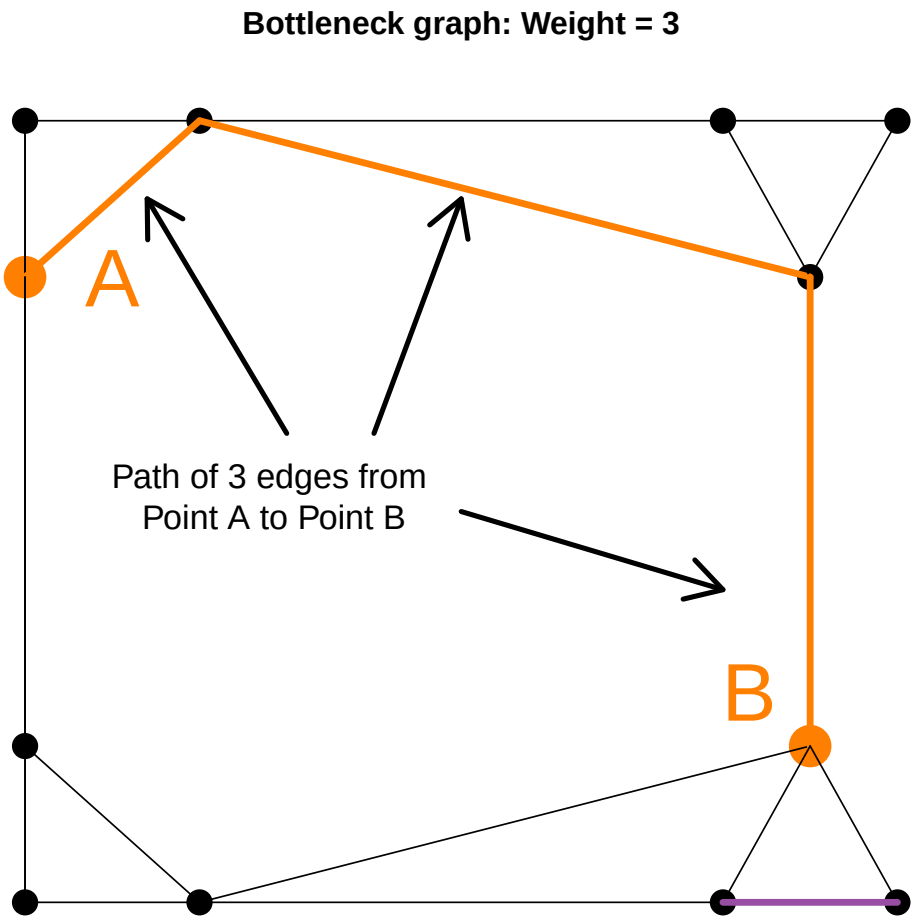
Bottleneck subgraph: In pictures

Bottleneck subgraph of weight threshold 3



Bottleneck subgraph: In pictures

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



Approximate algorithm outline:

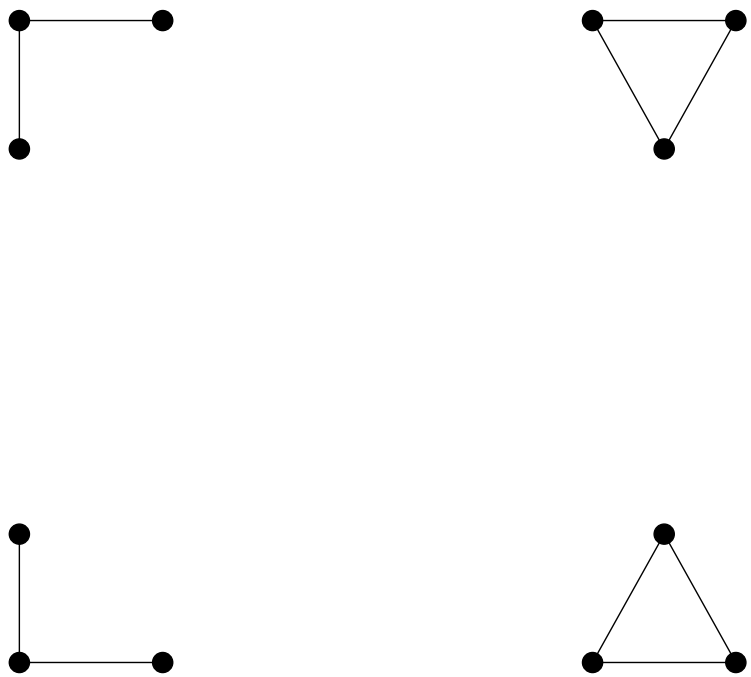
- Construct a bottleneck subgraph for a smart choice of weight threshold.
- Select block centers that are "just far enough apart."
- Grow from these block centers to obtain an approximately optimal blocking.
- Approach closely follows Hochbaum and Shmoys [1986].

Algorithm step-by-step: Find bottleneck graph

- Find smallest weight threshold λ^- such that each vertex in the corresponding bottleneck subgraph is connected to at least $t^* - 1$ edges.
- Can show that $\lambda^- \leq \lambda$, where λ is the smallest MWBD possible.
- Bottleneck subgraph can be constructed in roughly $O(n \log n)$ time.

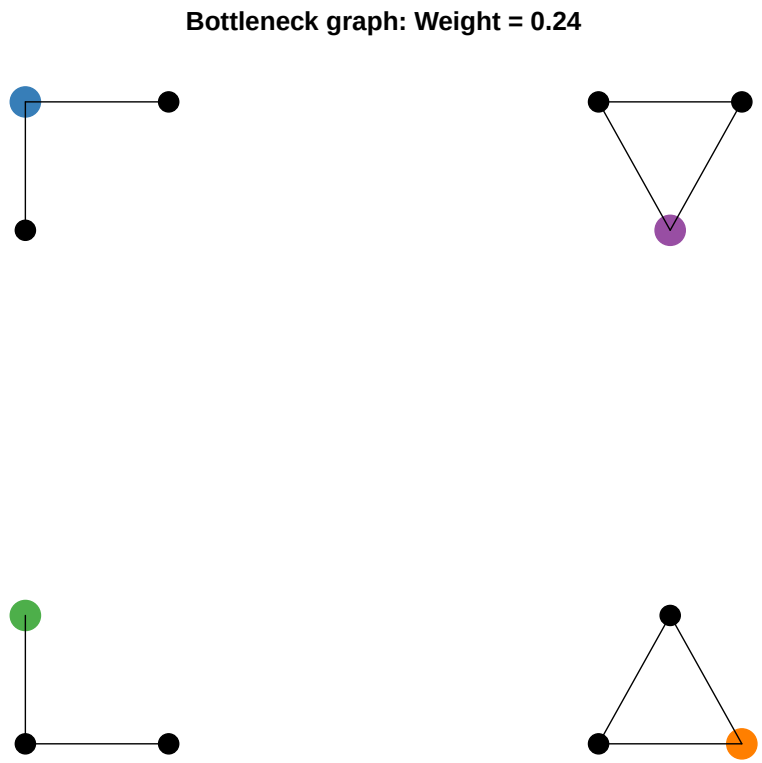
$$t^* = 2$$

Bottleneck graph: Weight = 0.24



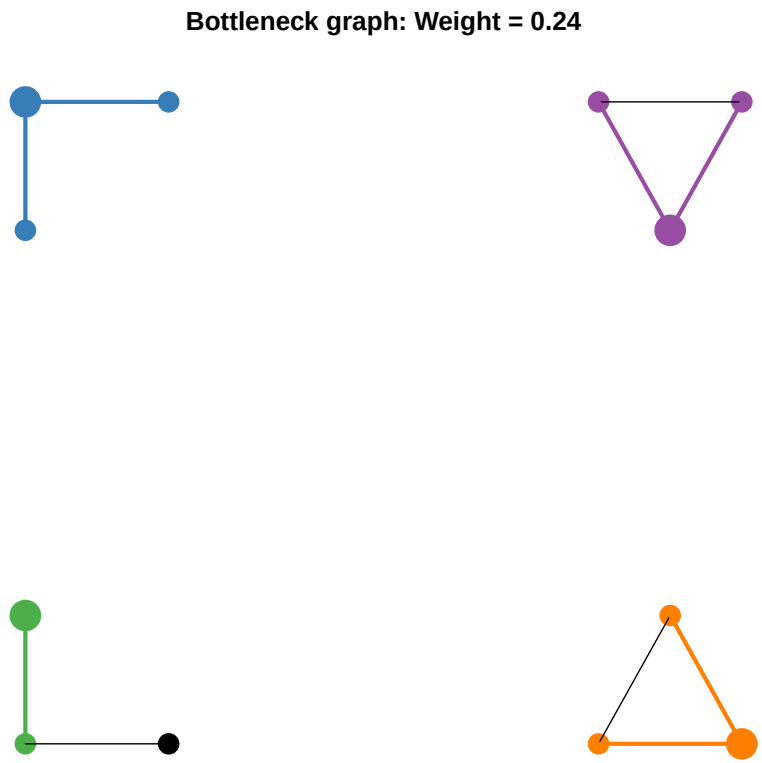
Algorithm step-by-step: Find block centers

- Find a set of vertices—*block centers*—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 will do, but some choices of centers are better.
- Takes $O(n^2)$ time.



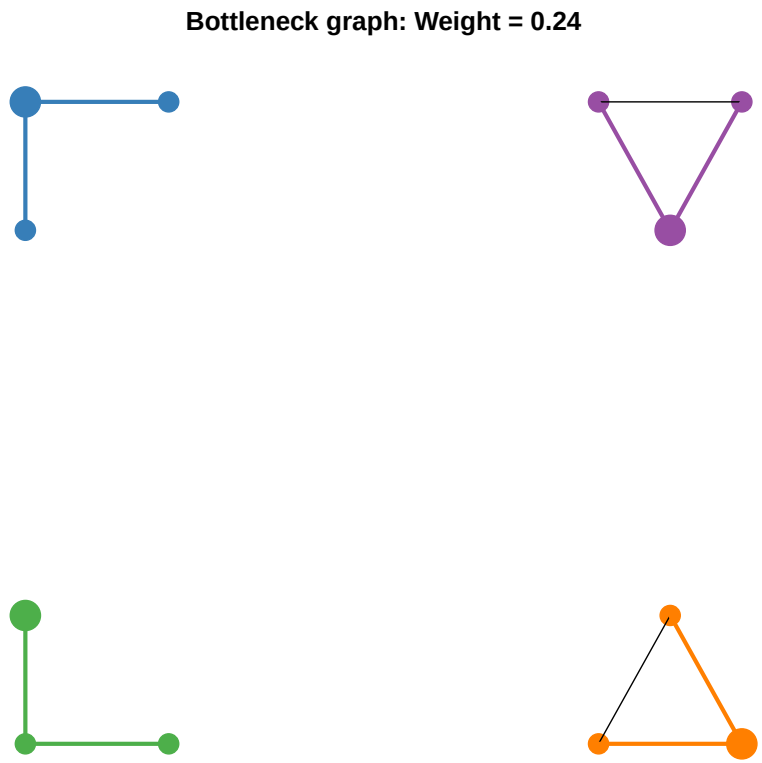
Algorithm step-by-step: Grow from block centers

- Form blocks comprised of a block center plus any vertices connected to that center by a single edge.
- The way our block centers were chosen (no path of two edges connects two block centers), these blocks will not overlap.
- At this point, these blocks contain at least t^* units (by edge connection criterion).



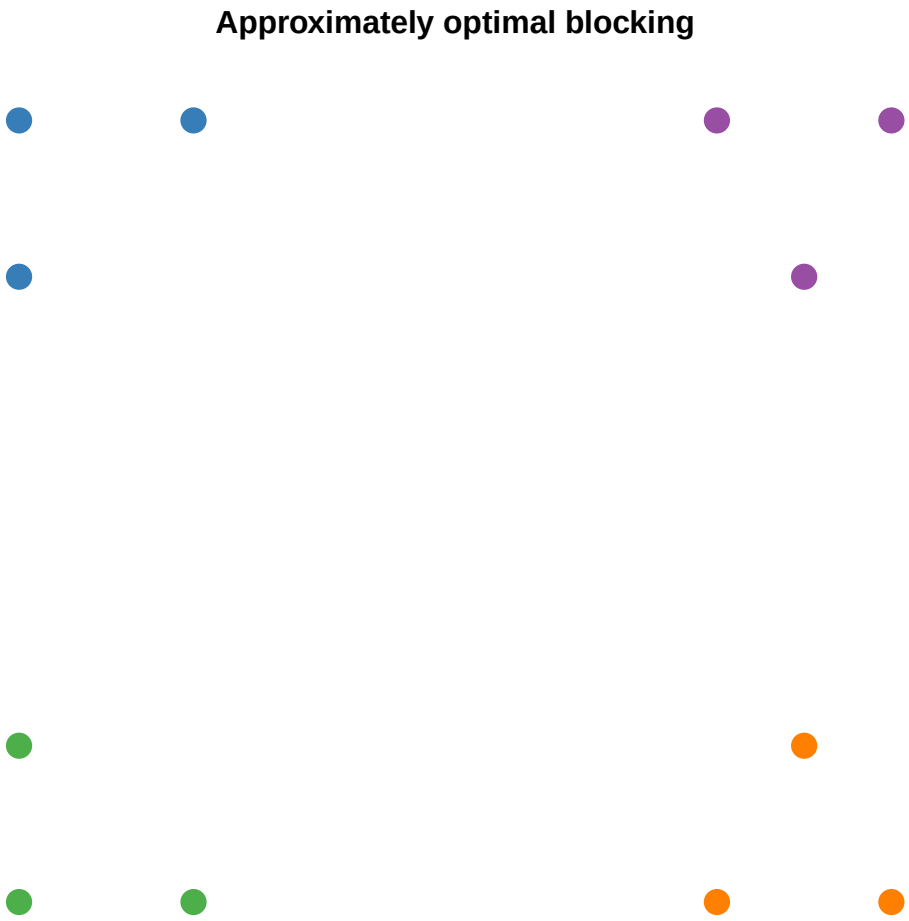
Algorithm step-by-step: Assign all unassigned vertices

- For each unassigned vertex, find the closest block center. Add that vertex to the center's corresponding block.
- The way our block centers were chosen, all unassigned vertices are at most a path of two edges away from a block center.
- This step and the previous step is completed in $O(n^2)$ time; the total runtime of the algorithm is roughly $O(n^2)$.



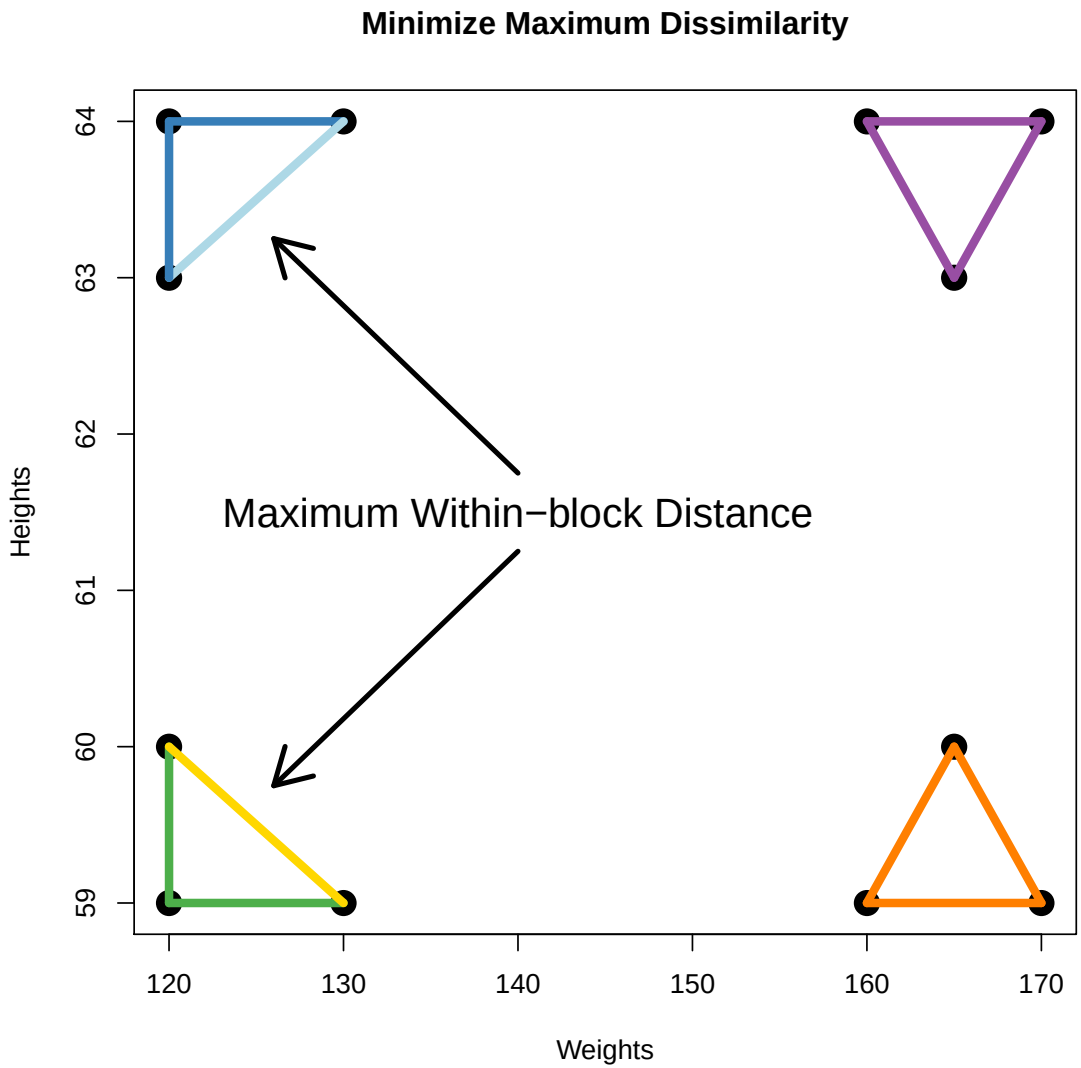
Our blocking

Our approximate algorithm came up with the following blocking:



A simple example:

Threshold $t^* = 2$. Dissimilarity = Mahalanobis distance.



Sketch of proof of approximate optimality

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

Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a blocking with MWBD $\leq 4\lambda$, 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 center $\Longrightarrow$

  In the worst case: two vertices $\{i\}, \{j\}$ in the same block can be connected by a path of four edges in the bottleneck subgraph (two from vertex $\{i\}$ to the block center, two from the block center to vertex $\{j\}$).

Sketch of proof

- Each vertex is at most a path of two edges away from a block center $\Longrightarrow$

  In the worst case: two vertices $\{i\}, \{j\}$ in the same block can be connected by a path of four edges in the bottleneck subgraph (two from vertex $\{i\}$ to the block center, two from the block center to vertex $\{j\}$).
- In the worst case: For any two vertices $\{i\}, \{j\}$ within the same block, there is a path of four edges connecting $\{i\}$ to $\{j\}$ in the bottleneck subgraph: $(i, k_1), (k_1, k_2), (k_2, k_3), (k_3, j)$
- Each edge has weight at most $\lambda^- \Longrightarrow$
  The corresponding edge weights satisfy:

$$w_{ik_1} + w_{k_1k_2} + w_{k_2k_3} + w_{k_3j} \leq 4\lambda^- \leq 4\lambda.$$

Sketch of proof cont'd

- Since edge weights satisfy the triangle inequality:

$$w_{ik} + w_{kj} \geq w_{ij}$$

it follows that

$$w_{ij} \leq w_{ik_1} + w_{k_1 k_2} + w_{k_2 k_3} + w_{k_3 j} \leq 4\lambda^{-} \leq 4\lambda.$$

- That is, every edge joining two vertices within the same block has weight $\leq 4\lambda$.
- The maximum within-block distance of the approximately optimal blocking is $\leq 4\lambda$.
- QED

Some final remarks about algorithm:

- A t^*-nearest-neighbors graph can be used instead of a bottleneck subgraph; approximate optimality still holds, and blockings tend to be closer to optimal.
- t^*-nearest-neighbors graph—an edge is drawn between $\{i\}$ and $\{j\}$ if and only if $\{j\}$ is one of the t^* closest vertices to $\{i\}$ or $\{i\}$ is one of the t^* closest vertices to $\{j\}$.

Some final remarks about algorithm:

- A t^*-nearest-neighbors graph can be used instead of a bottleneck subgraph; approximate optimality still holds, and blockings tend to be closer to optimal.
- t^*-nearest-neighbors graph—an edge is drawn between $\{i\}$ and $\{j\}$ if and only if $\{j\}$ is one of the t^* closest vertices to $\{i\}$ or $\{i\}$ is one of the t^* closest vertices to $\{j\}$.
- Quick, local changes to the approximately optimal partition may improve the partition (e.g., divide large blocks into smaller blocks, swap units between blocks).
- Algorithm does not contain any inherently random components.

Estimating treatment effects: Notation

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

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

- 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]:

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

- 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]:

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

Example:

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

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

- 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 (SATE_{st}):

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

- Two unbiased estimators of SATE_{st} are the difference-in-means estimator and the the Horvitz-Thompson estimator.

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

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

Variance of estimators

$$\mathrm{Var}(\hat{\delta}_{st,\mathrm{diff}}) = \mathrm{Var}(\hat{\delta}_{st,\mathrm{HT}})$$

$$= \sum_{c=1}^{b} \frac{n_c^2}{n^2} \left(\frac{r-1}{n_c - 1}(\sigma_{cs}^2 + \sigma_{ct}^2) + 2\frac{\gamma_{cst}}{n_c - 1} \right)$$

$$\mu_{cs} = \frac{1}{n_c} \sum_{k=1}^{n_c} y_{kcs}$$

$$\sigma_{cs}^2 = \frac{1}{n_c} \sum_{k=1}^{n_c} (y_{kcs} - \mu_{cs})^2$$

$$\gamma_{cst} = \frac{1}{n_c} \sum_{k=1}^{n_c} (y_{kcs} - \mu_{cs})(y_{kct} - \mu_{ct})$$

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

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

- Note: σ_{cs}^2 and σ_{ct}^2 are estimable, γ_{cst} not directly estimable.

Variance of estimators

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

- Note: σ_{cs}^2 and σ_{ct}^2 are estimable, γ_{cst} not directly estimable.
- Conservative estimate:

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

Results: A toy example

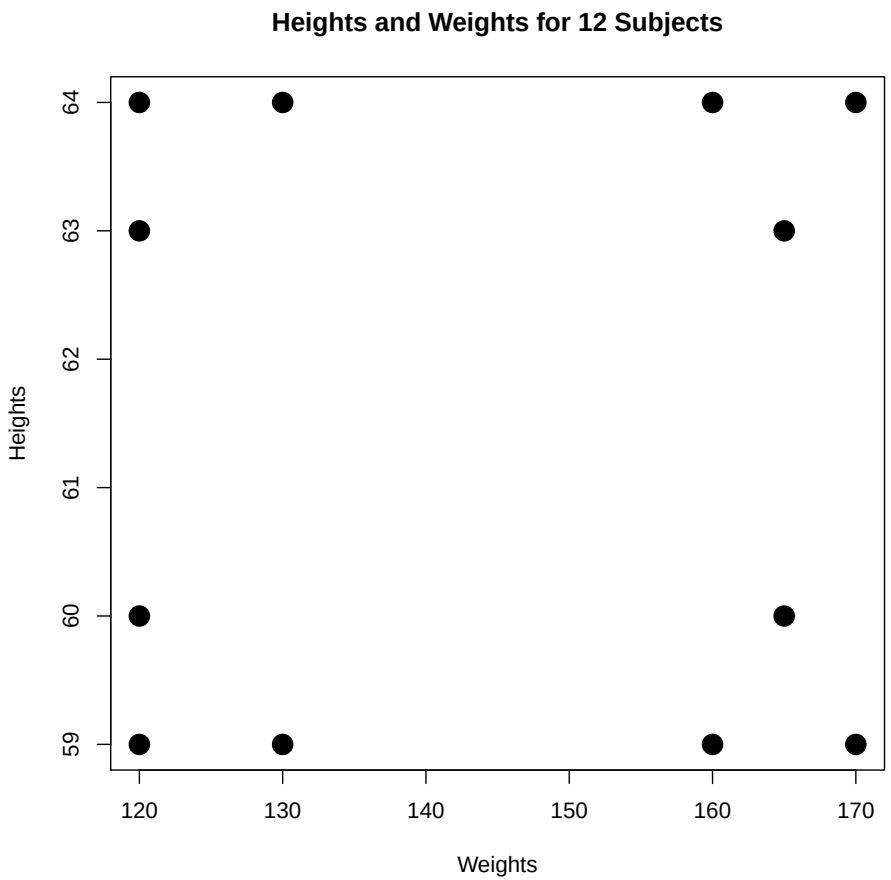
Consider the following toy example:

- "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:

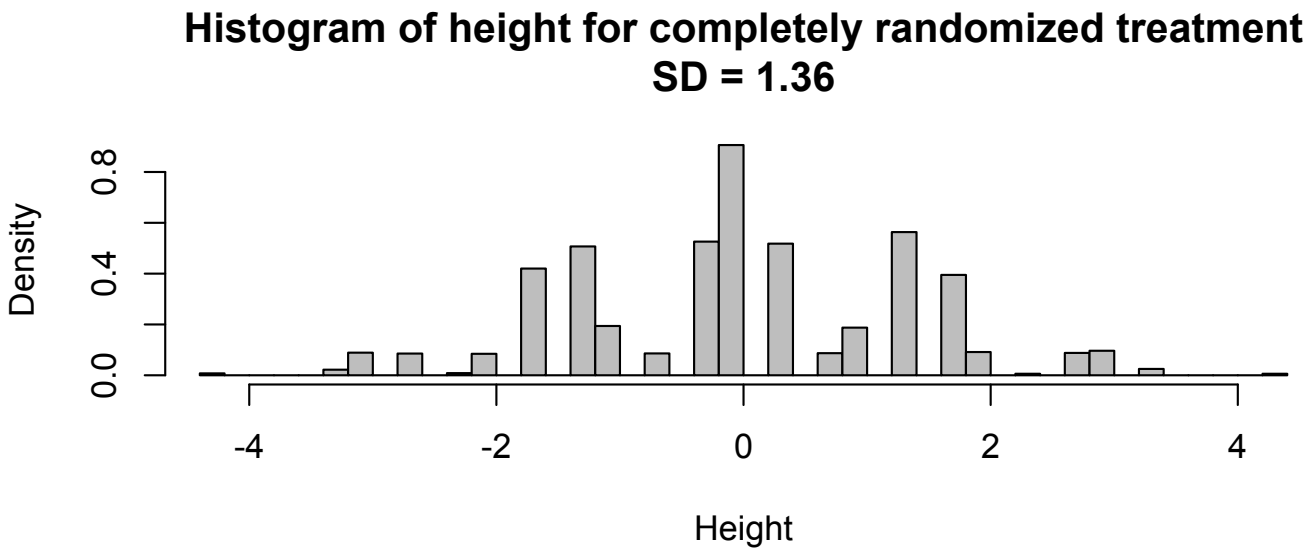
$$\begin{aligned} \text{Health score}_i &= 3(\text{height}_i) + \text{weight}_i + 1.5\sqrt{(\text{height}_i)(\text{weight}_i)} \\ &\quad +50(\text{takeVitamin}_i) \end{aligned}$$

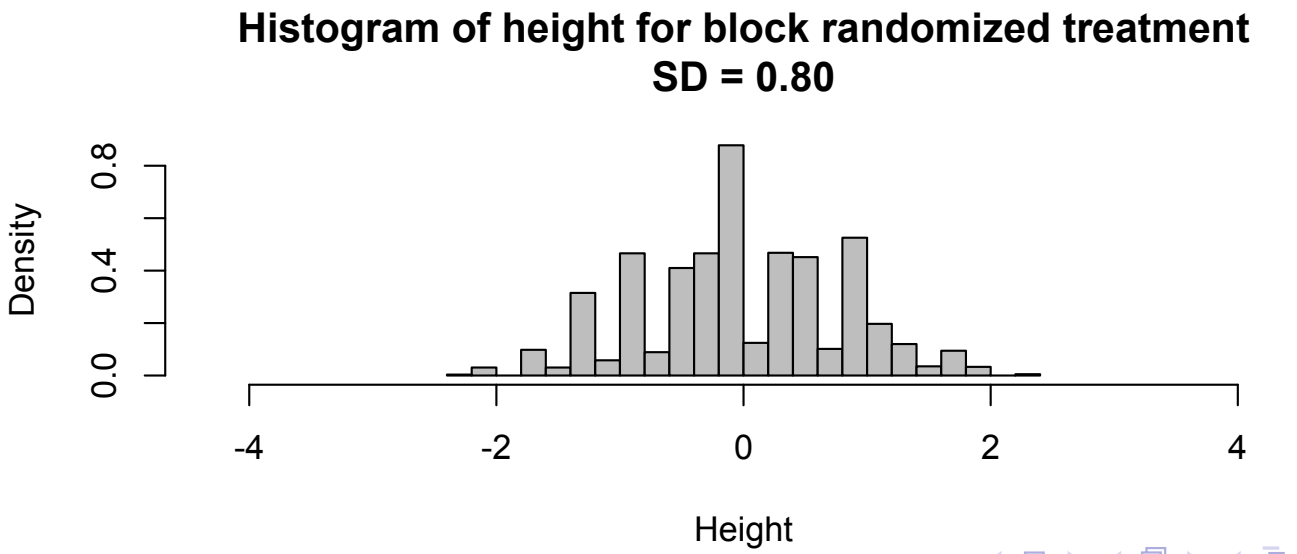
Results: A toy example

- 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.

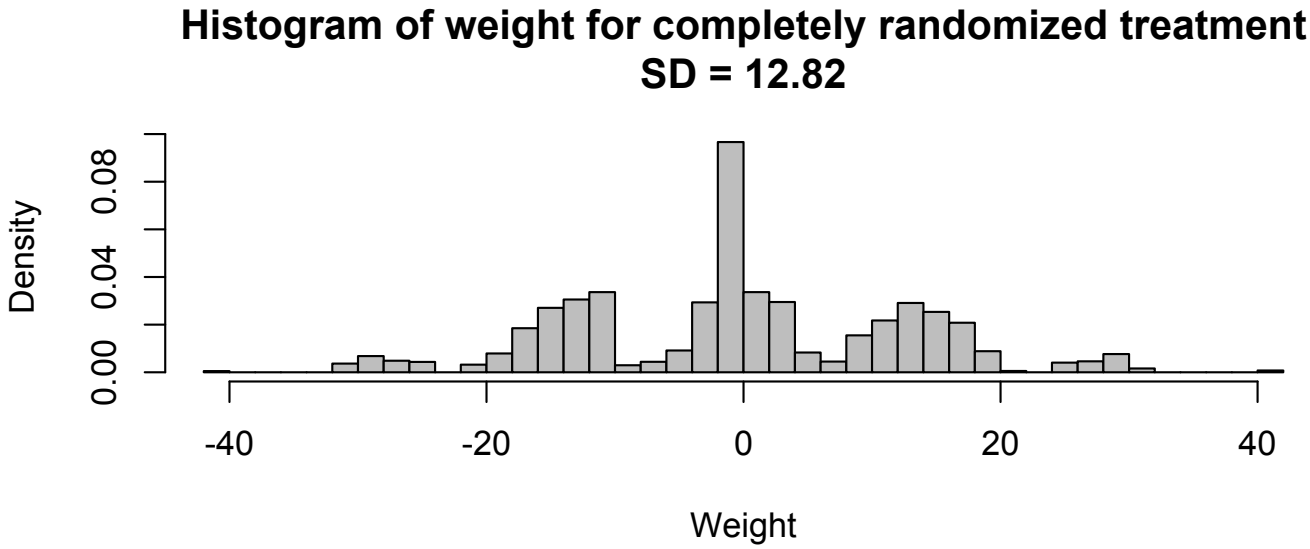


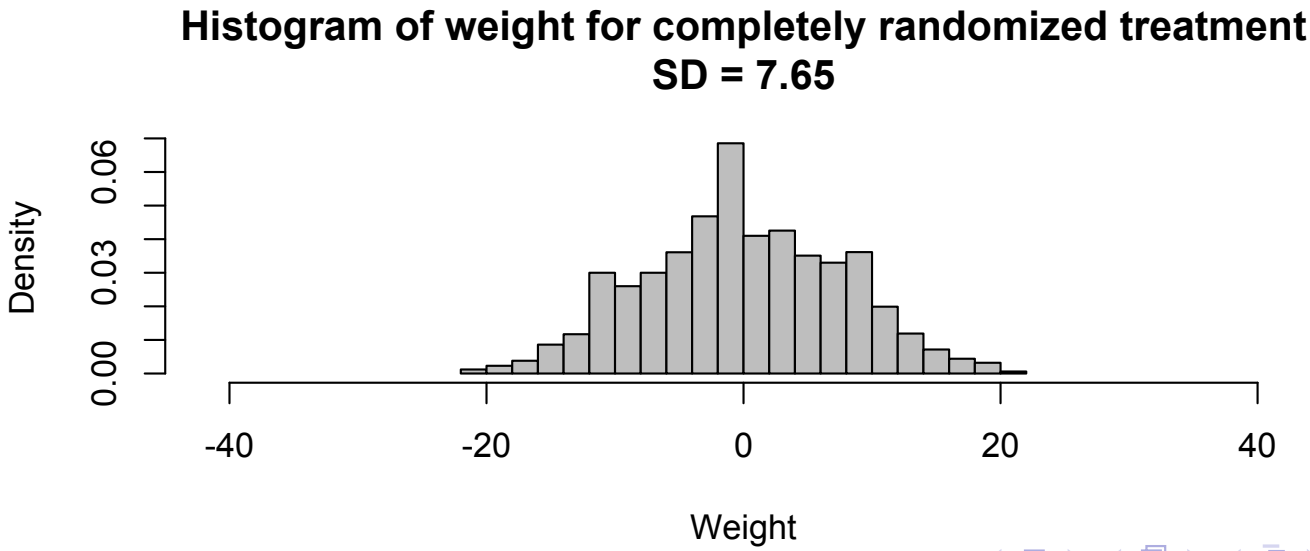
Covariate balance: Height



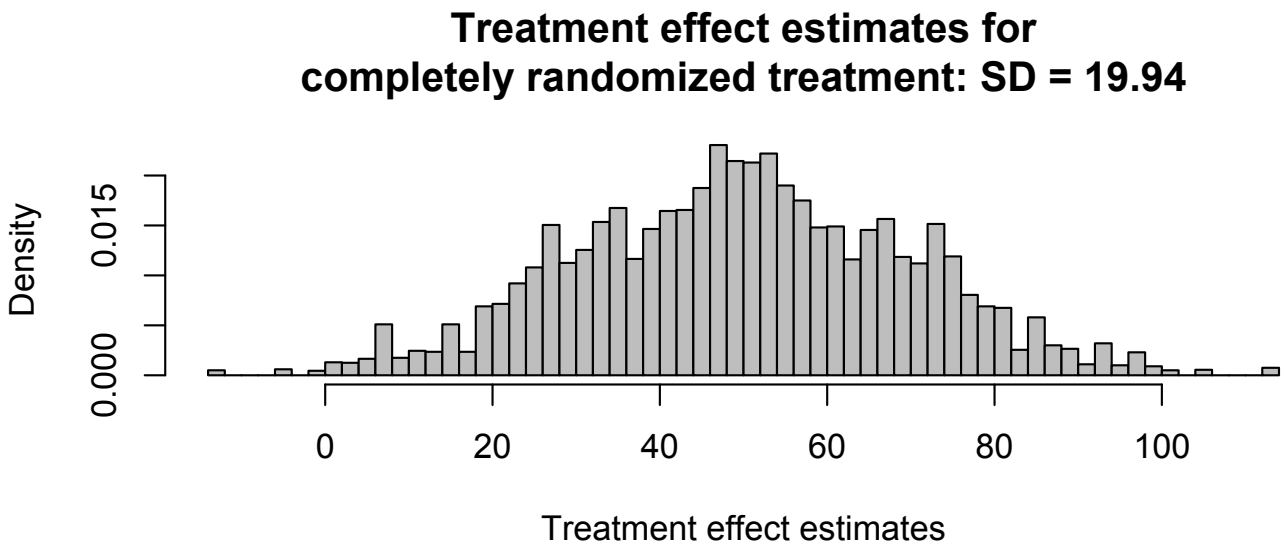


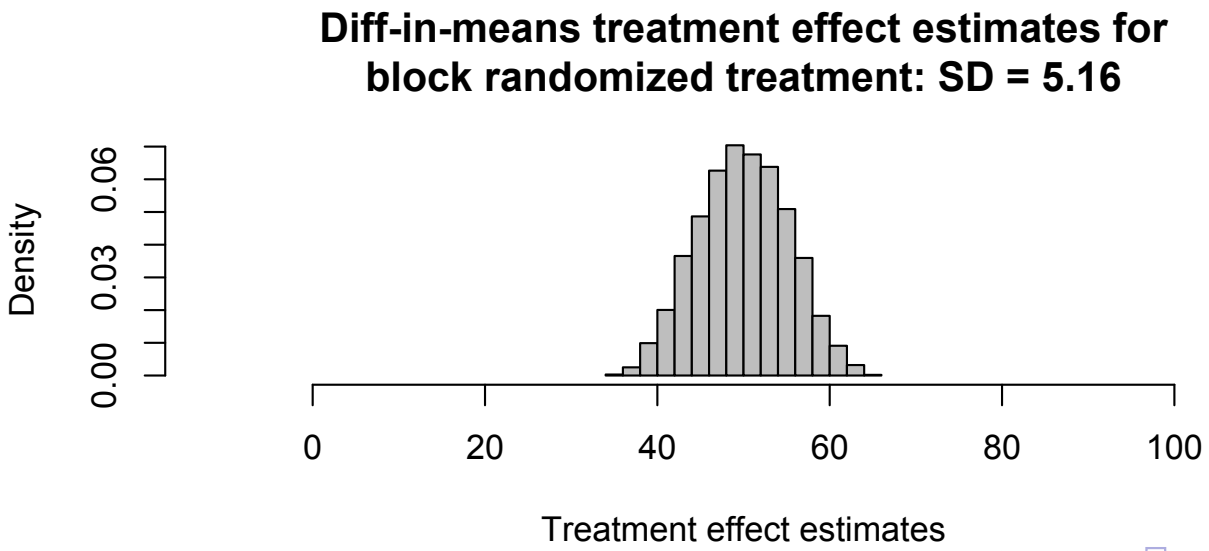
Covariate balance: Weight





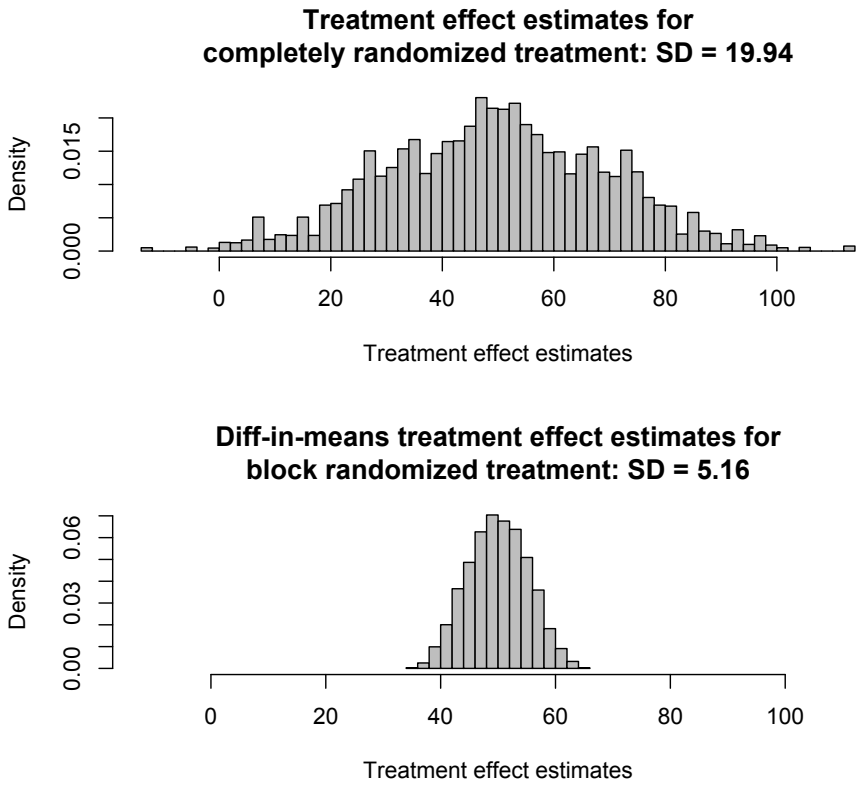
Comparison of estimates





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.



Future Work

- Apply graph partitioning techniques to other statistical problems.
  - Clustering—alternative to k-means.
  - Other problems in nonparametric statistics.
- Algorithm improvements:
  - Decrease run-time of the algorithm.
  - Add additional structure to improve partition quickly.
  - Can factor of 4 be improved?

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.

Guido W. Imbens. Experimental design for unit and cluster randomized trials. Working Paper, 2011.

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.