Threshold Partitioning Problems and Applications in Statistics

Michael J. Higgins

Kansas State University

Oct. 1, 2016

Work with Fredrik Savje and Jasjeet Sekhon

Effect of treatments

- Causal inference—a subfield of statistics
- Common goal in causal inference: interested in the effect of a "treatment" on a "response."

Effect of treatments

- Causal inference—a subfield of statistics
- Common goal in causal inference: interested in the effect of a "treatment" on a "response."
- E.g.: Medical trial.
  Treatment: Experimental procedure.
  Response: Recovery time.

  Would someone with a slow recovery have healed faster had they received the experimental procedure?

Effect of treatments

- Causal inference—a subfield of statistics
- Common goal in causal inference: interested in the effect of a "treatment" on a "response."
- E.g.: Medical trial.
  Treatment: Experimental procedure.
  Response: Recovery time.

  Would someone with a slow recovery have healed faster had they received the experimental procedure?
- E.g: Get out the vote.
  Treatment: Receive a mailer.
  Response: Voting in an election.

  Would a non-voter have voted had they received a flyer encouraging them to vote?

Prognostically important covariates

- There may be additional observable variables—*covariates*—that may be correlated with response.

Prognostically important covariates

- There may be additional observable variables—*covariates*—that may be correlated with response.
- E.g.: Medical trial.
  Pretreatment covariate: Initial health.

  Patients with poor health initially have slower recovery times regardless if they receive or do not receive the experimental procedure.

Prognostically important covariates

- There may be additional observable variables—*covariates*—that may be correlated with response.
- E.g.: Medical trial.
  Pretreatment covariate: Initial health.

  Patients with poor health initially have slower recovery times regardless if they receive or do not receive the experimental procedure.
- Factoring these *prognostically important* covariates into the design of a statistical analysis is much better than ignoring these covariates.

Experiments vs. observational studies

Two settings: *Experiments* and *observational studies*.

- Experiments: Researcher (randomly) assigns treatments to units.
  E.g.: Sending mailers to households.

  Can account for covariates before assigning treatments.

Experiments vs. observational studies

Two settings: *Experiments* and *observational studies*.

- Experiments: Researcher (randomly) assigns treatments to units. E.g.: Sending mailers to households.

  Can account for covariates before assigning treatments.

- *Statistical blocking*—Group units with similar values of prognostically important covariates. Assign treatment within each group and independently across groups.

  Ensures similar covariate distributions across different treatment groups.

Experiments vs. observational studies

- Observational studies: Researcher does not control treatment assignments.
  E.g.: Effect of smoking on heart disease.
- Prognostically important covariates often correlated both with treatment and response.
  E.g.: People who smoke often have worse diets, which may also increase incidence of heart disease.

Experiments vs. observational studies

- Observational studies: Researcher does not control treatment assignments.
  E.g.: Effect of smoking on heart disease.
- Prognostically important covariates often correlated both with treatment and response.
  E.g.: People who smoke often have worse diets, which may also increase incidence of heart disease.
- *Statistical matching*—For each treated unit, find a control unit that has similar values of prognostically important covariates. Analyze differences of these units.

  May ensure similar covariate distributions across treatment groups.

Blocking, matching, and graph partitioning

Idea: View blocking and matching as graph partitioning problems [Rosenbaum, 1989, Greevy *et. al.* 2004].

- Units are vertices in a graph.
- Edges signify that two units can be blocked or matched together.
- Edge costs are a measure of dissimilarity on prognostically important covariates (e.g. Mahalanobis distance, Euclidian distance).

  Dissimilarity between two units is small if they have similar values of covariates.

  Required to satisfy triangle inequality.

Blocking, matching, and graph partitioning

Idea: View blocking and matching as graph partitioning problems [Rosenbaum, 1989, Greevy *et. al.* 2004].

- Units are vertices in a graph.
- Edges signify that two units can be blocked or matched together.
- Edge costs are a measure of dissimilarity on prognostically important covariates (e.g. Mahalanobis distance, Euclidian distance).

  Dissimilarity between two units is small if they have similar values of covariates.

  Required to satisfy triangle inequality.
- Goal: Find a partition with small within-block costs. Helps ensure that all units in the same block are similar.

Threshold partitioning

Our Solution: Threshold partitioning with a bottleneck objective.

- *Threshold partitioning*—each block of the partition contains at least k vertices for some prespecified threshold k.

  For observational studies: Require each block to contain at least k_r vertices corresponding to treatment r.

- *Bottleneck objective*—Minimize the maximum within-block cost (MWBC). Forces all units in a block to be "similar" to each other.

Threshold partitioning

Our Solution: Threshold partitioning with a bottleneck objective.

- *Threshold partitioning*—each block of the partition contains at least k vertices for some prespecified threshold k.

  For observational studies: Require each block to contain at least k_r vertices corresponding to treatment r.
- *Bottleneck objective*—Minimize the maximum within-block cost (MWBC). Forces all units in a block to be "similar" to each other.
- Accommodates arbitrarily many treatment arms and multiple replications of each treatment within each block.
- Ensures good covariate balance in small experiments and observational studies.
- Efficient enough for massive datasets (100's of millions of units).

Optimal blocking and approximately optimal blocking

For all partitions that contain at least k units within each block:

- Let λ denote the smallest MWBC achievable by such a partition—any partition that meets this bound is called an *optimal partition*.
- Finding optimal partition is NP-hard—feasible to find in small datasets, may not be in large datasets.

Optimal blocking and approximately optimal blocking

For all partitions that contain at least k units within each block:

- Let λ denote the smallest MWBC achievable by such a partition—any partition that meets this bound is called an *optimal partition*.
- Finding optimal partition is NP-hard—feasible to find in small datasets, may not be in large datasets.
- We show a partition with MWBC $\leq 4\lambda$ is constructable in $O(kn)$ time and space, outside of forming a nearest neighbor graph

  Find "good" partition when number of units is small *or massive*.

Optimal blocking and approximately optimal blocking

For all partitions that contain at least k units within each block:

- Let λ denote the smallest MWBC achievable by such a partition—any partition that meets this bound is called an *optimal partition*.
- Finding optimal partition is NP-hard—feasible to find in small datasets, may not be in large datasets.
- We show a partition with MWBC $\leq 4\lambda$ is constructable in $O(kn)$ time and space, outside of forming a nearest neighbor graph

  Find "good" partition when number of units is small *or massive*.
- Similar results when adding k_r size requirement for treatments r.

Optimal blocking and approximately optimal blocking

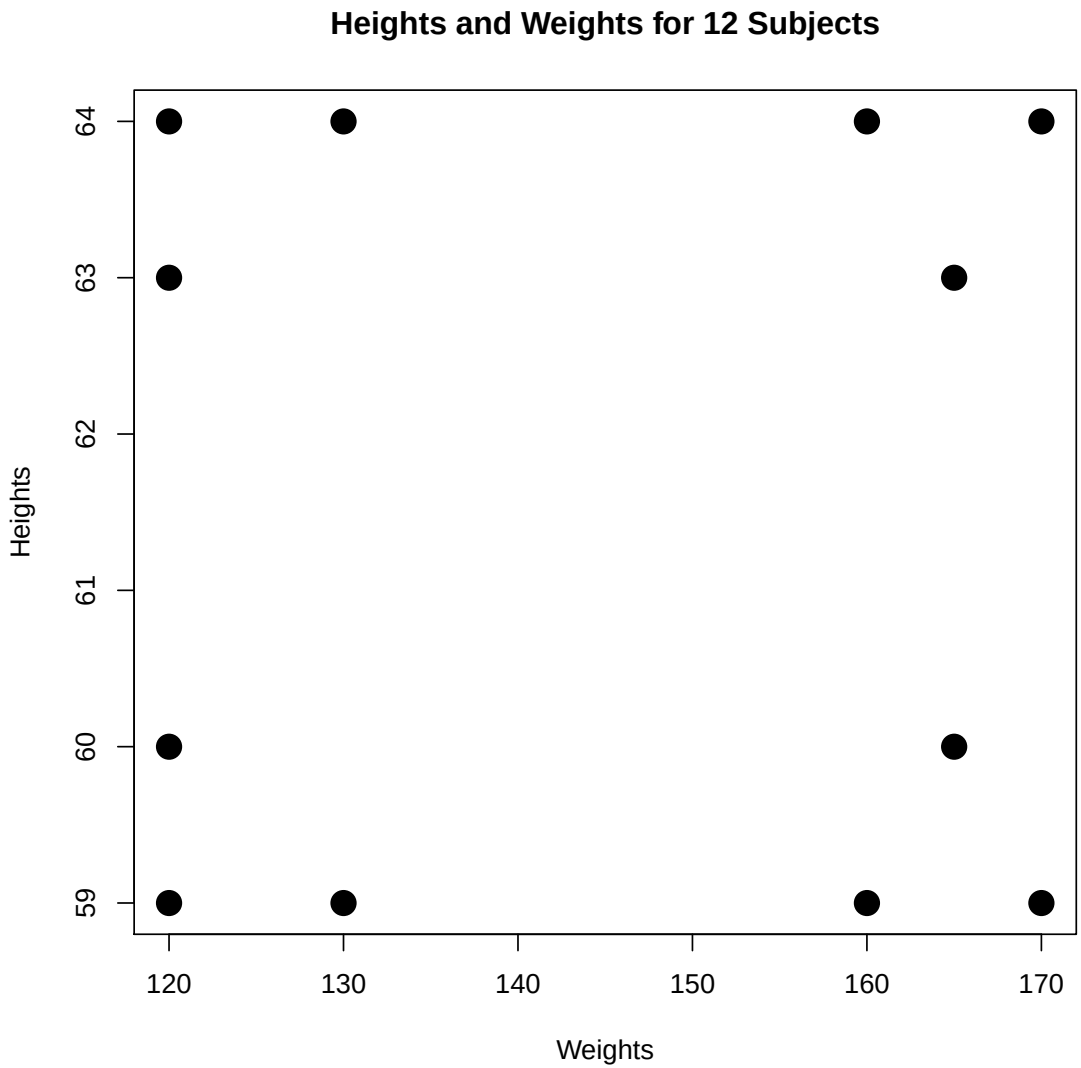
For all partitions that contain at least k units within each block:

- Let λ denote the smallest MWBC achievable by such a partition—any partition that meets this bound is called an *optimal partition*.
- Finding optimal partition is NP-hard—feasible to find in small datasets, may not be in large datasets.
- We show a partition with MWBC $\leq 4\lambda$ is constructable in $O(kn)$ time and space, outside of forming a nearest neighbor graph

  Find "good" partition when number of units is small *or massive*.
- Similar results when adding k_r size requirement for treatments r.
- Denote any such partition as an *approximately optimal partition*.

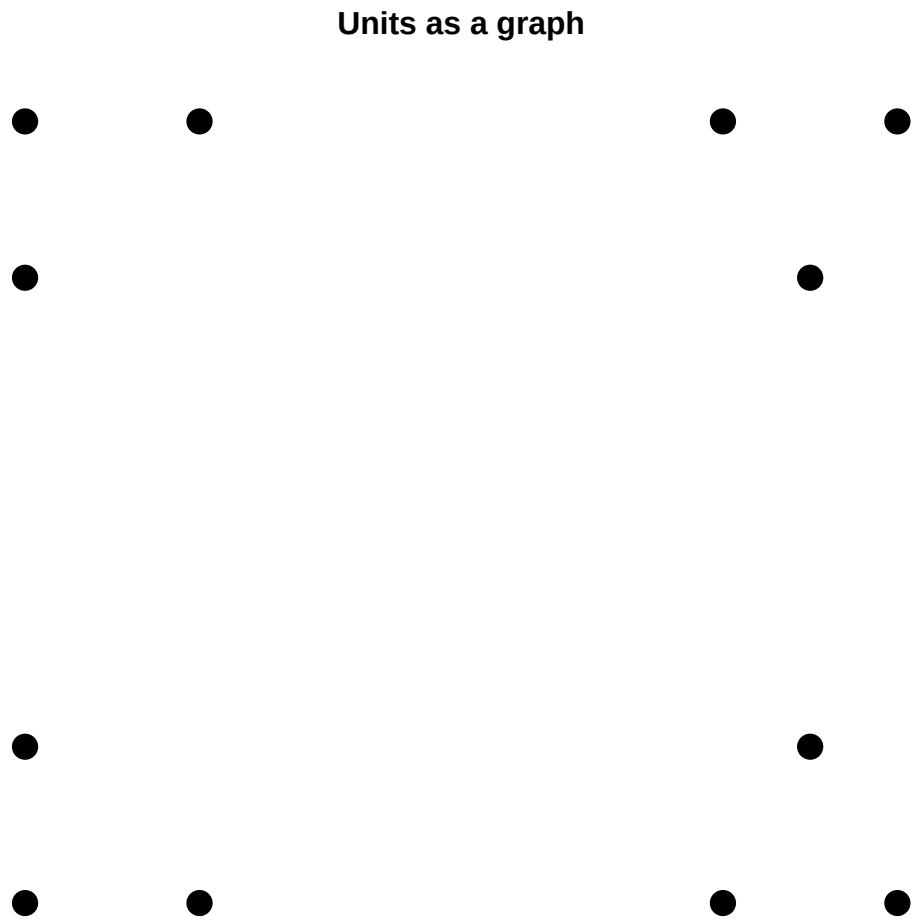
Blocking of an experiment as a graph: In pictures

Dissimilarity = Mahalanobis distance.



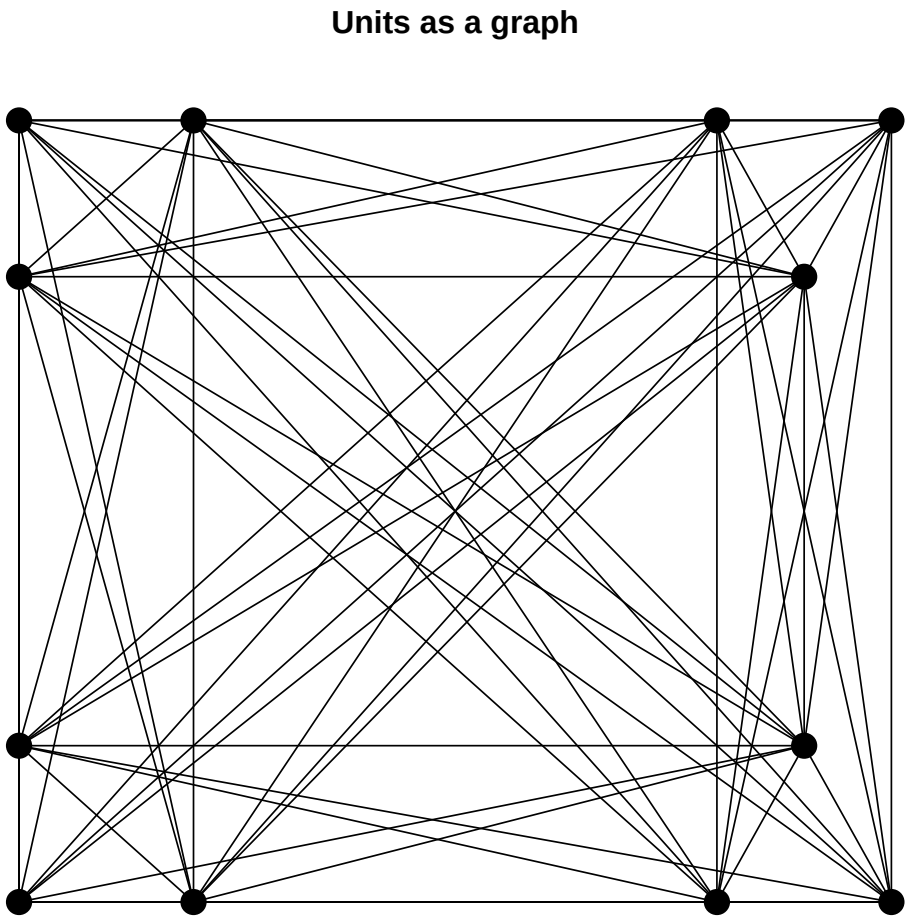
Blocking of an experiment as a graph: In pictures

Dissimilarity = Mahalanobis distance.



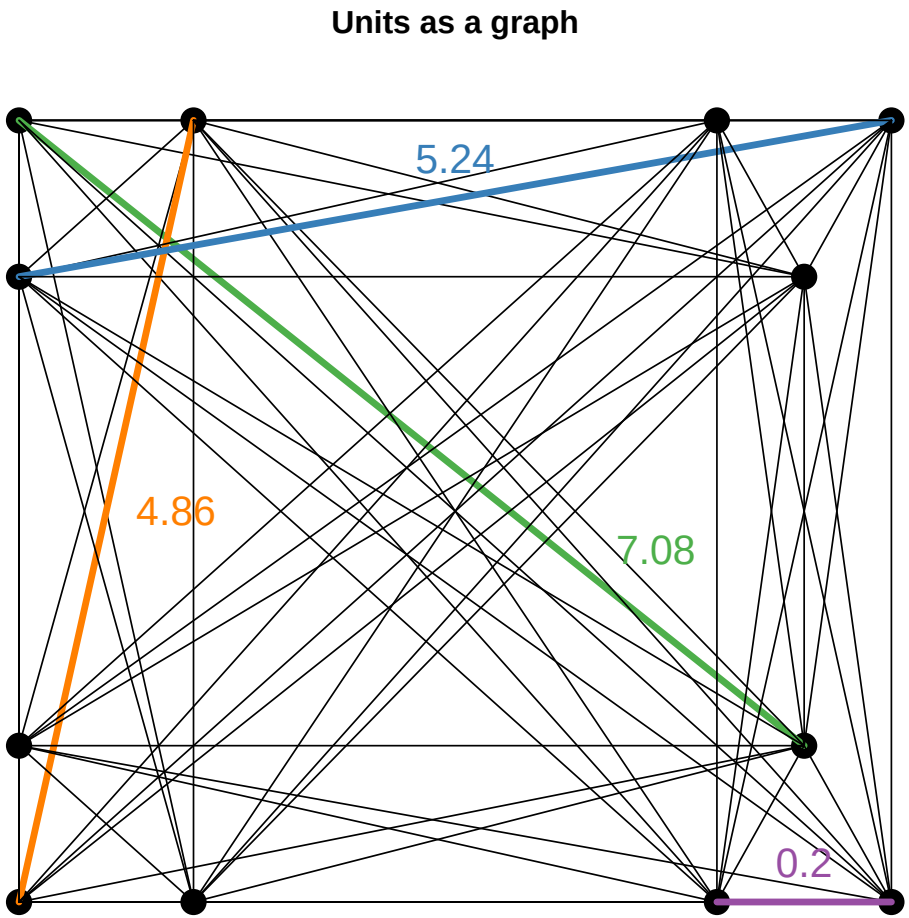
Blocking of an experiment as a graph: In pictures

Dissimilarity = Mahalanobis distance.



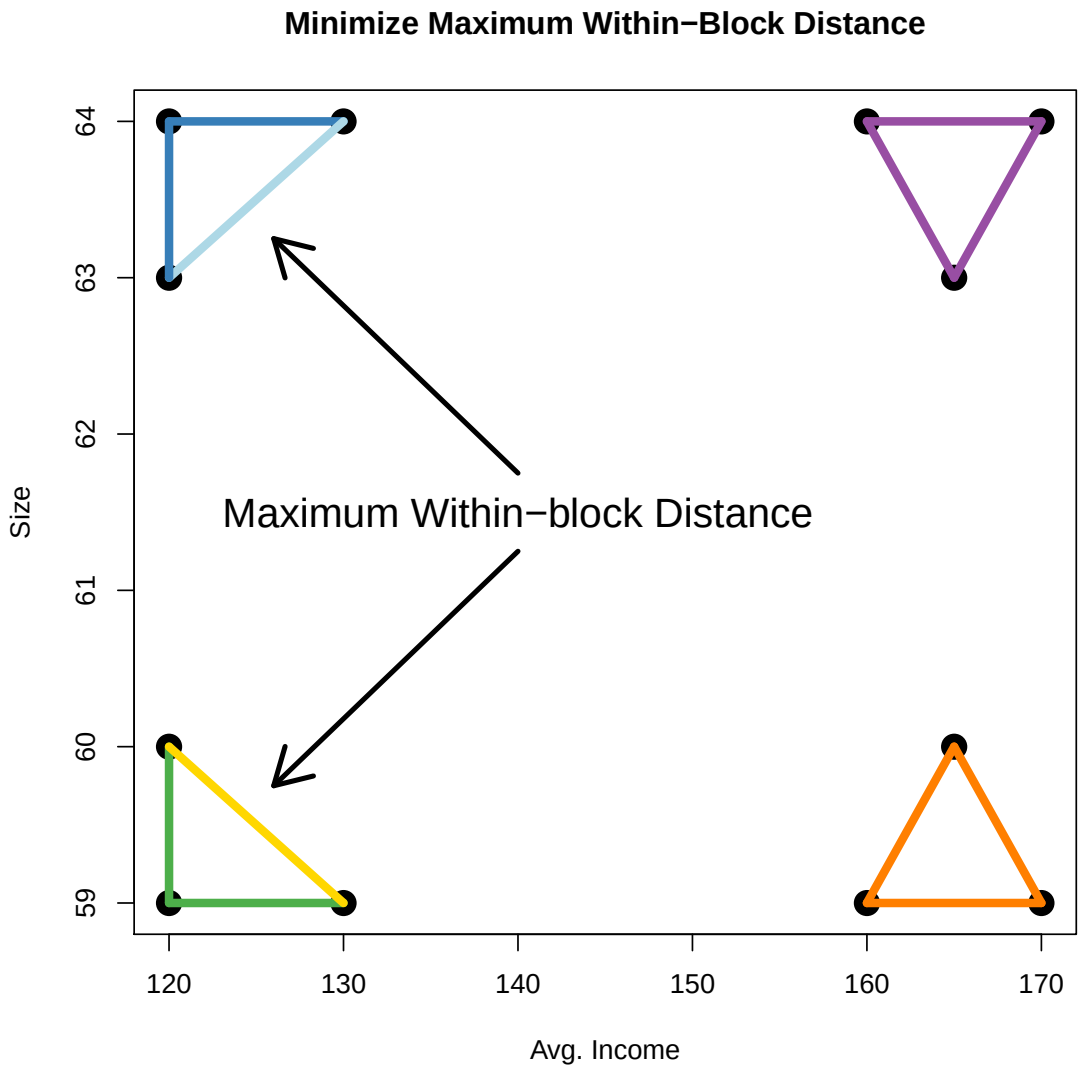
Blocking of an experiment as a graph: In pictures

Dissimilarity = Mahalanobis distance.



A simple example:

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



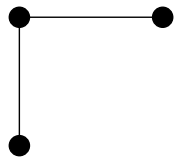
Approximate algorithm outline:

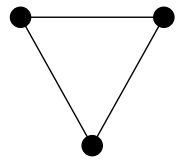
- Construct a $(k-1)$–nearest neighbor subgraph.
- Select block seeds that are "just far enough apart."
- Grow from these block centers to obtain an approximately optimal blocking.
- Approach extends from Hochbaum and Shmoys [1986].

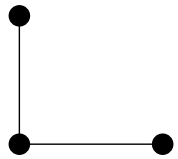
Algorithm step-by-step: Find nearest neighbor graph

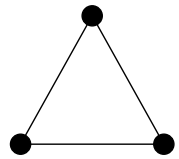
- Construct a $(k-1)$–nearest-neighbors graph.
- For observational study: Use directed nearest-neighbors digraph.
- Can show that edge costs are, at most, λ.

$k = 2$



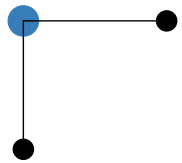


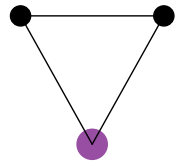


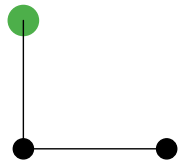


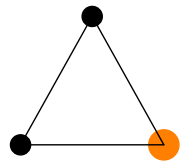
Algorithm step-by-step: Find block centers

- 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.



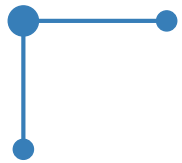


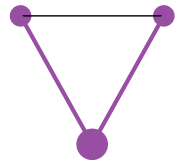


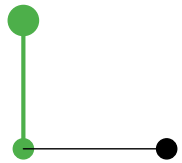


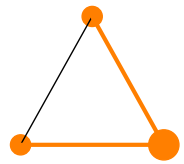
Algorithm step-by-step: Grow from block centers

- 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.



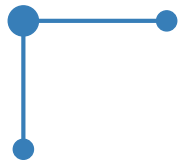


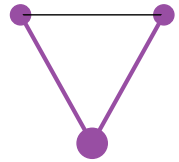


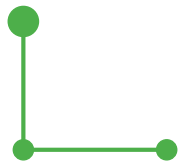


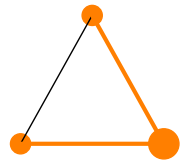
Algorithm step-by-step: Assign all unassigned vertices

- 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.



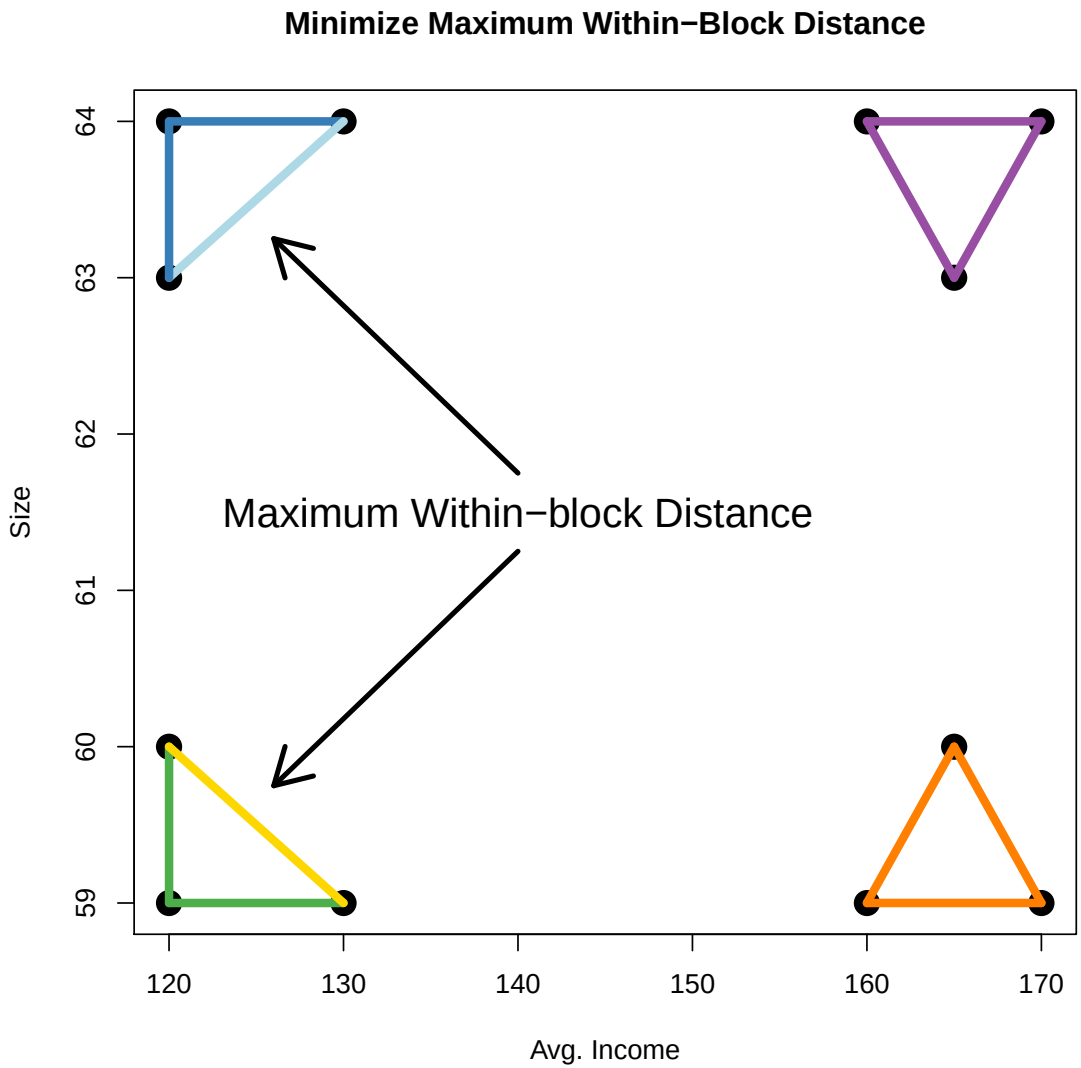






A simple example:

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



Sketch of proof of approximate optimality

- Algorithm is guaranteed to obtain a partition 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 partition 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

  $\Longrightarrow$

  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 partition 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

  $\Longrightarrow$

  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:

$$i\ell_1,\ \ell_1\ell_2,\ \ell_2\ell_3,\ \ell_3 j \tag{1}$$

Sketch of proof

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

$$c_{i\ell_1} + c_{\ell_1\ell_2} + c_{\ell_2\ell_3} + c_{\ell_3 j} \leq 4\lambda.$$

Sketch of proof

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

$$c_{i\ell_1} + c_{\ell_1\ell_2} + c_{\ell_2\ell_3} + c_{\ell_3 j} \leq 4\lambda.$$

- Since edge costs satisfy the triangle inequality:

$$c_{ij} \leq c_{i\ell_1} + c_{\ell_1\ell_2} + c_{\ell_2\ell_3} + c_{\ell_3 j} \leq 4\lambda.$$

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

Heuristic Improvements:

Some quick adjustments can improve performance of algorithm:

- Heuristics for improving selection of block seeds.
- Subdivide blocks with more than $2k$ units
- Local search (e.g. Kernighan–Lin)

Thank you.