Causal Inference in The Age of Big Data: Linearithmic Algorithm for Blocking/Matching/Clustering

Jasjeet S. Sekhon

UC Berkeley

April 4, 2016

What's the Big Deal about Big Data?

- One view: We just have to handle the data
  - Build a bigger computer system
  - It is a database problem
- Another view:
  - we need an integration between inferential and algorithmic thinking
- Measuring human activity has generated massive datasets with granular information that can be used for personalization of treatments, creating markets, modeling behavior
- Many inferential issues: e.g., unknown sampling frames, heterogeneity, targeting optimal treatments, compound loss functions

Massive Experiments

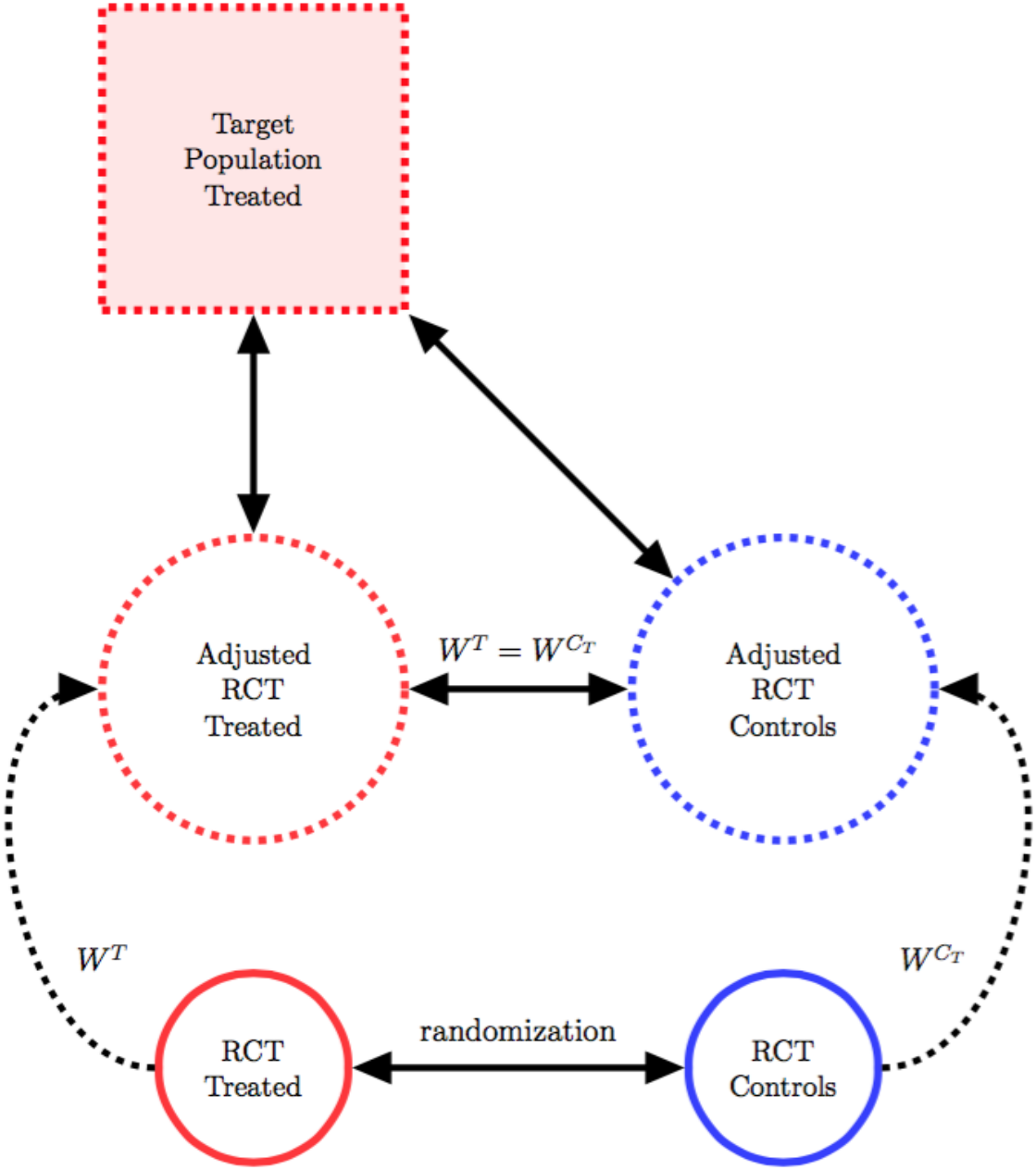
- Rising interest in fine-grained inference: e.g., subgroups
- Some traditional experimental design methods have become computationally infeasible—e.g., blocking
- Blocking: create strata and then randomize within strata
- Polynomial time solution not quick enough. Linearithmic is survivable. Sublinear needed in some cases.
- Algorithm can also be used for matching and clustering

The Problem

- Randomized Controlled Trials (RCTs) are rare and often small, especially a problem with medical experiments
- RCTs usually not conducted on the population of interest
- Combine information from both RCTs and population data to estimate treatment effects in the population
- Precise targeting of treatments, e.g., precision medicine

The Sample Selection Problem

- We want to make inferences for the full population of interest:
  - RCTs raise issues of Randomization Bias (Heckman and Smith 1995): poor external validity
  - NRSs raise issues of Selection Bias, or non random assignment to treatment
- How to combine information from RCTs and NRSs?


Target
Population
Treated
Adjusted
RCT
Treated
$W^T = W^{C_T}$
Adjusted
RCT
Controls
W^T
W^{C_T}
RCT
Treated
randomization
RCT
Controls

A New Blocking Method

The method minimizes the pair-wise Maximum Within-Block Distance: λ

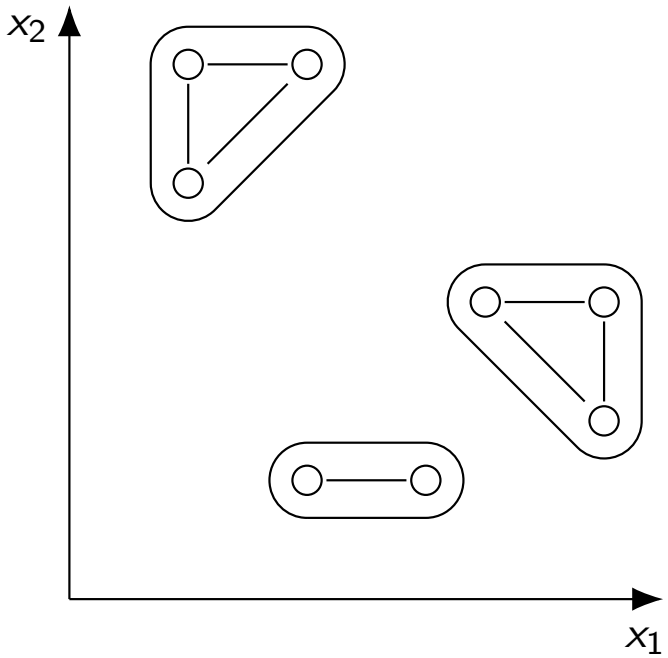
- Any valid distance metric (must satisfy the triangle inequality)
- Ensures good covariate balance by design
- Works for any number of treatments and any minimum number of observations per block
- It is fast: $O(n \log n)$ expected time
- It is memory efficient: $O(n)$ storage
- Approximately optimal: $\leq 4 \times \lambda$
- Special cases
  1. with one covariate: λ
  2. with two covariates: $\leq 2 \times \lambda$

Some Current blocking approaches

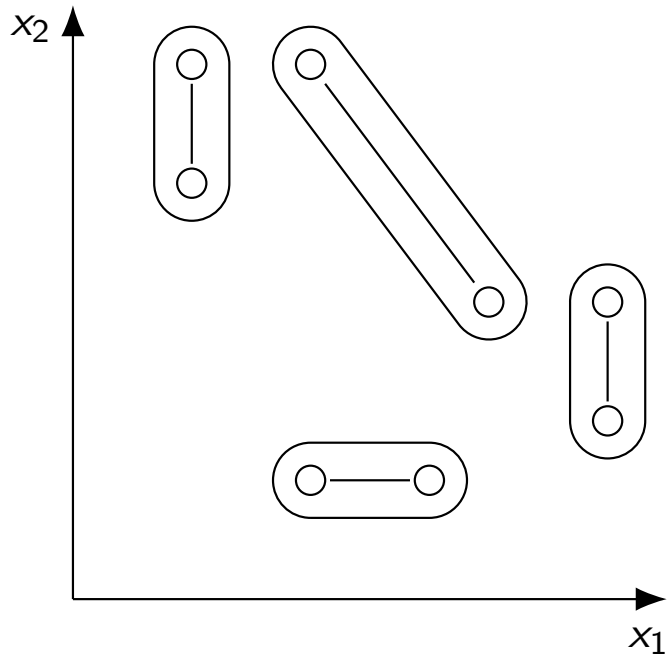
- Optimal Multivariate Matching Before Randomization
  - No efficient way to extend approach to more than two treatment categories
  - Even for two treatment categories, doesn't scale well
- Matched-pairs blocking: Pair "most-similar" units together. For each pair, randomly assign one unit to treatment, one to control
  - Natural clustering in the data ignored
  - Cannot estimate conditional variances
  - Difficulty with treatment effect heterogeneity

Threshold blocking: relaxing the block structure

Threshold blocking



Fixed-sized blocking



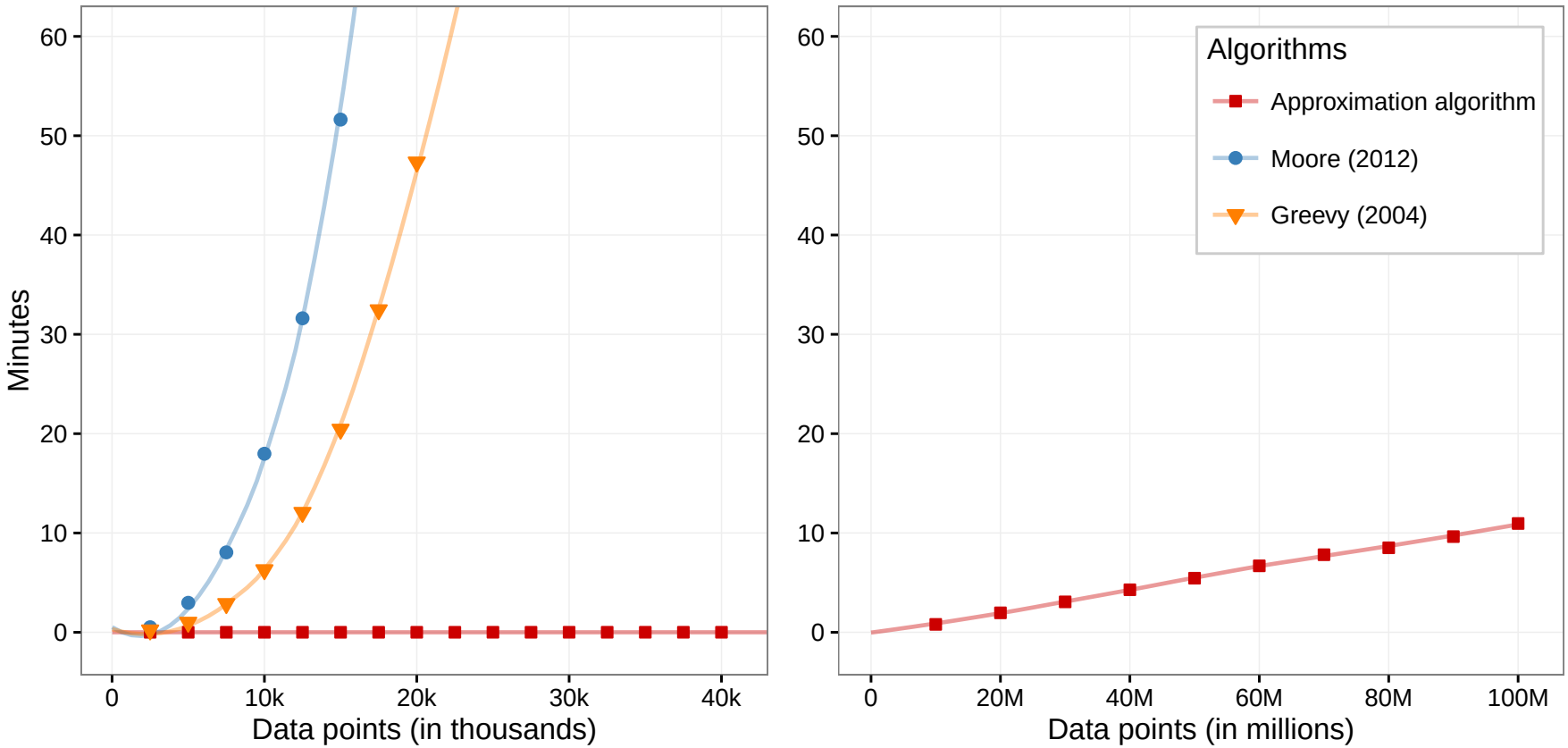
An Advantage

Theorem

For all samples, all objective functions and all desired block sizes, the optimal threshold blocking is always weakly better than the optimal fixed-sized blocking.

- Proof: interpret blocking as an non-linear integer programming problem.
  - The search set of threshold blocking is a superset of fixed-sized blocking

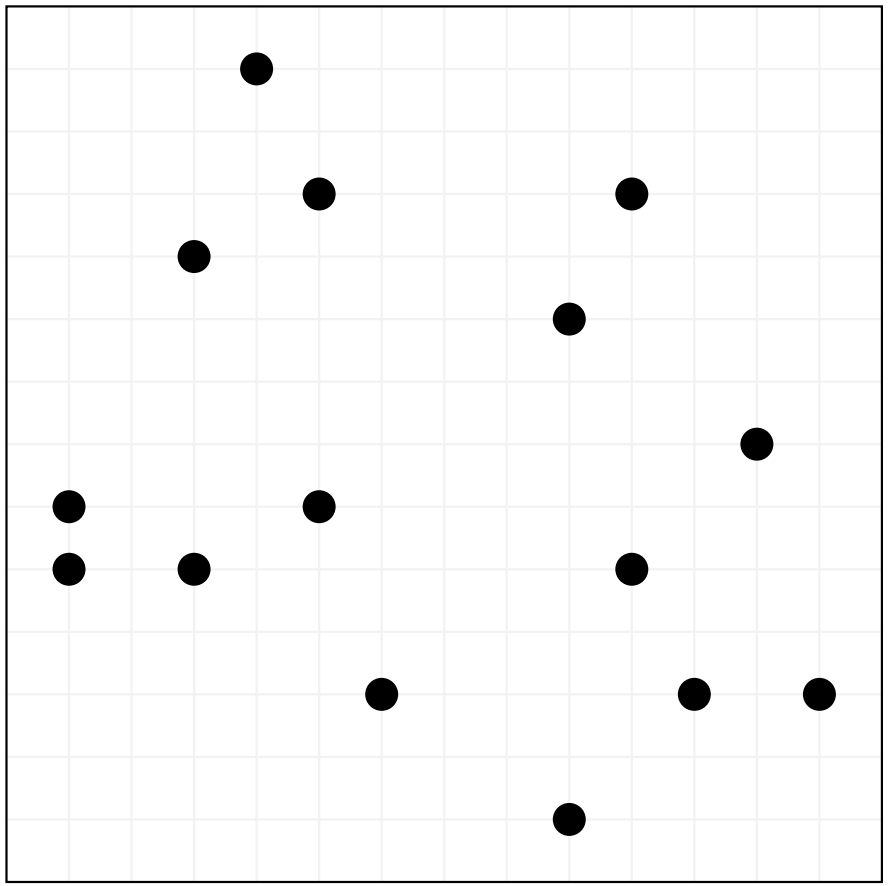
The AppOpt algorithm



The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$



Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor

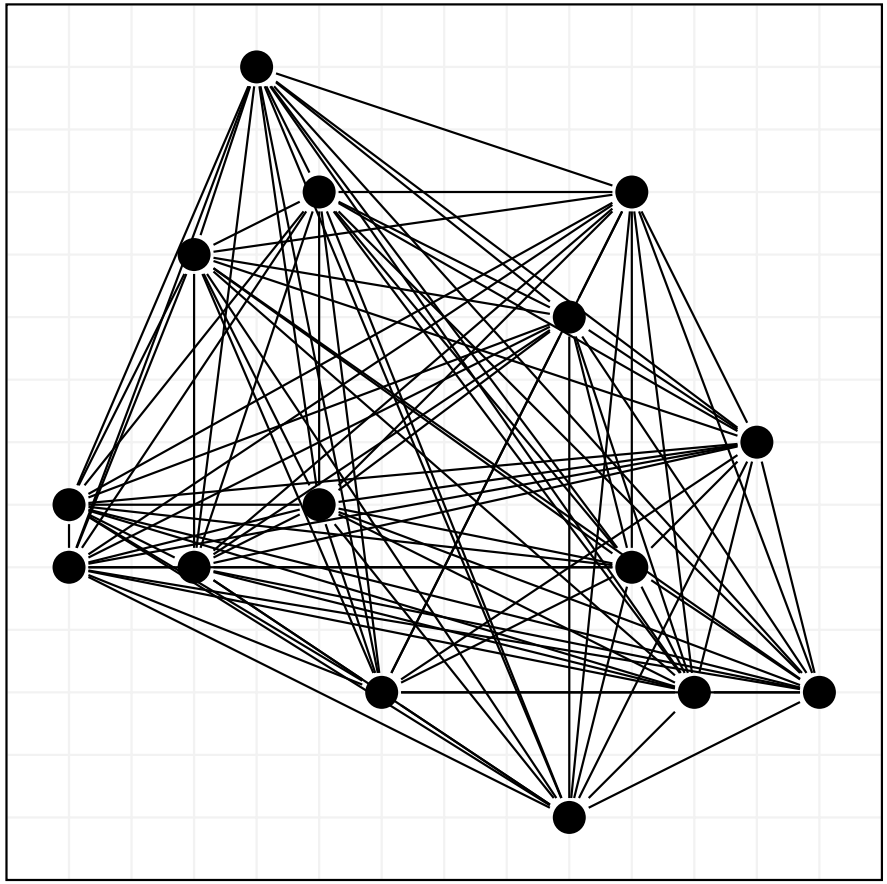
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



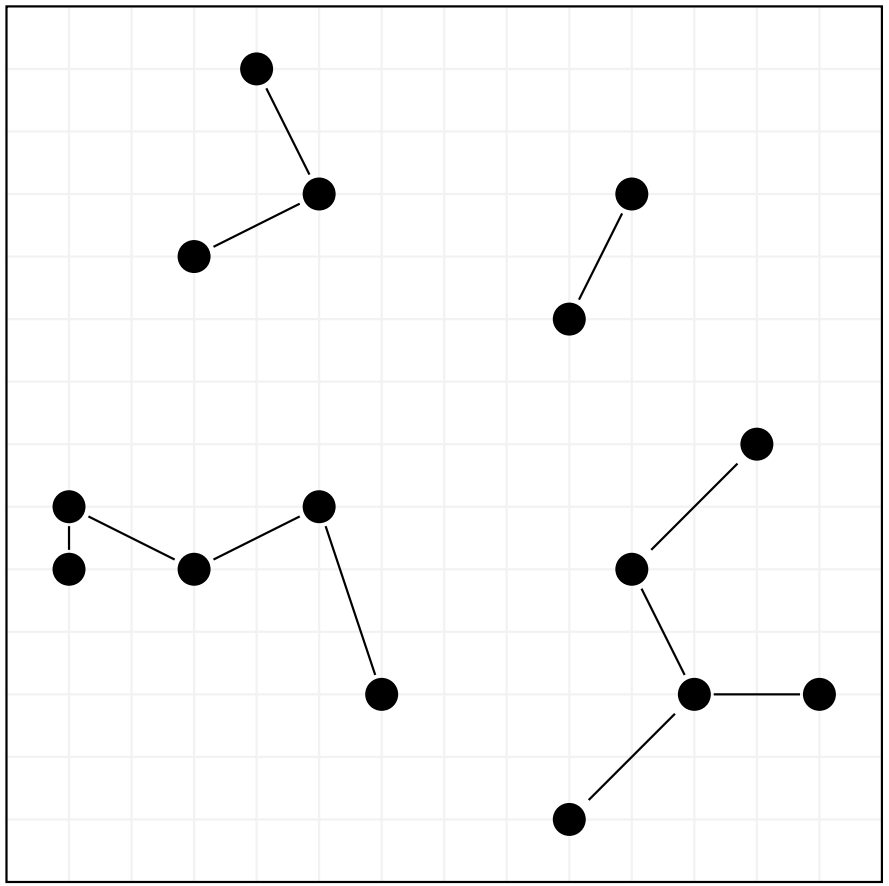
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



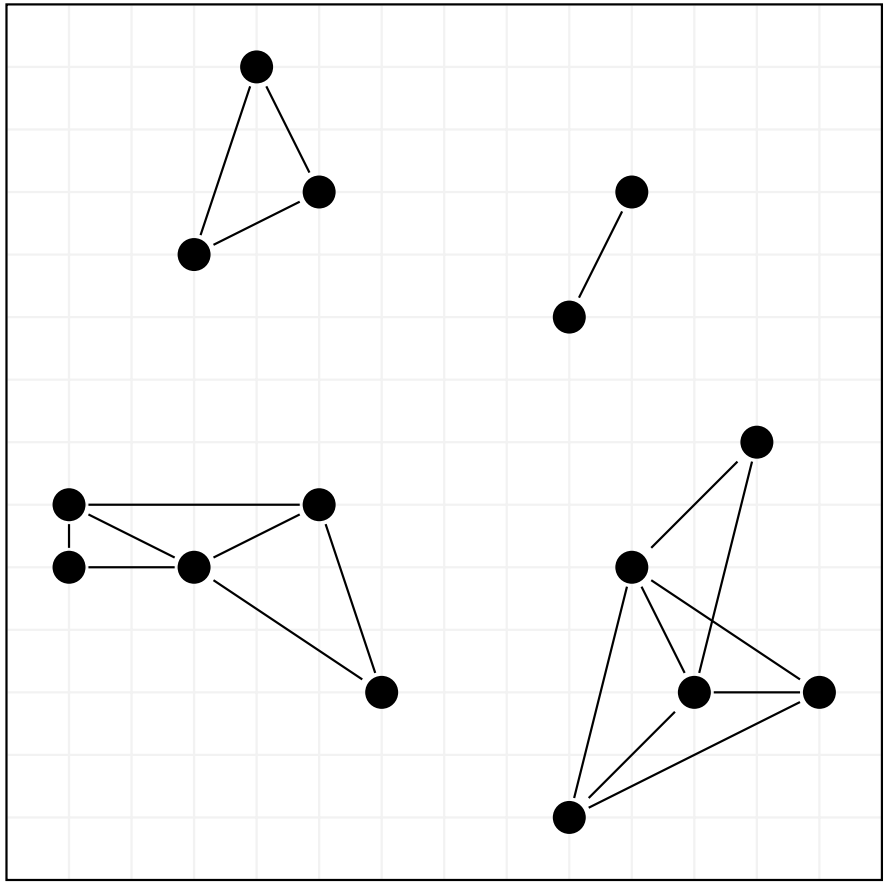
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



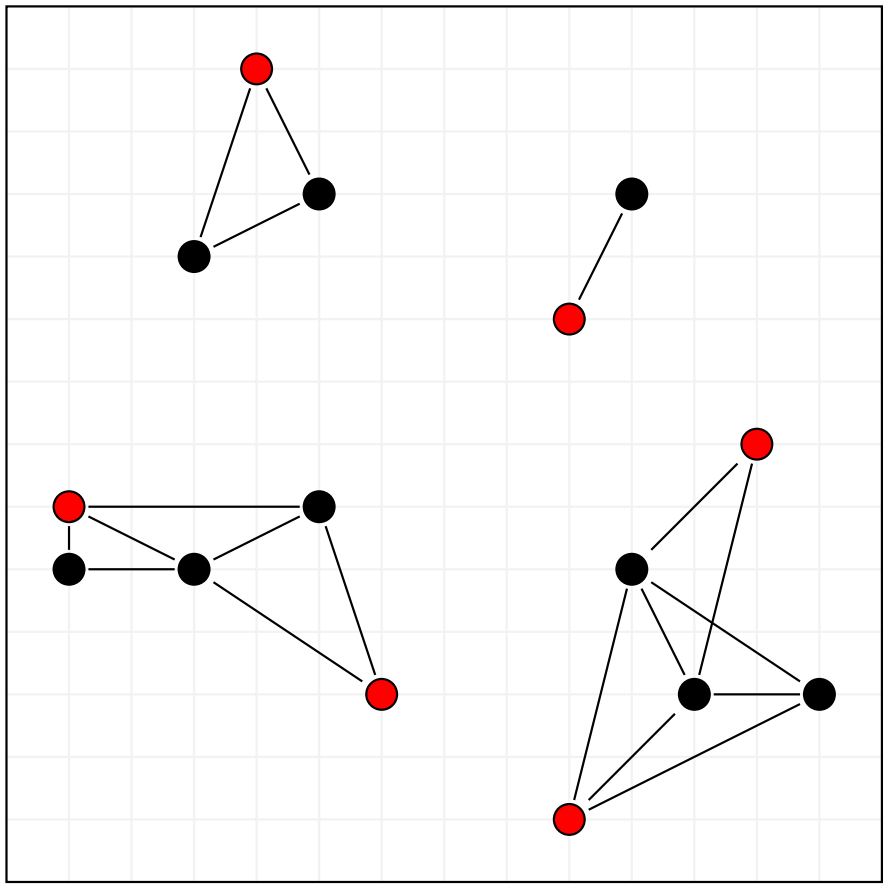
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



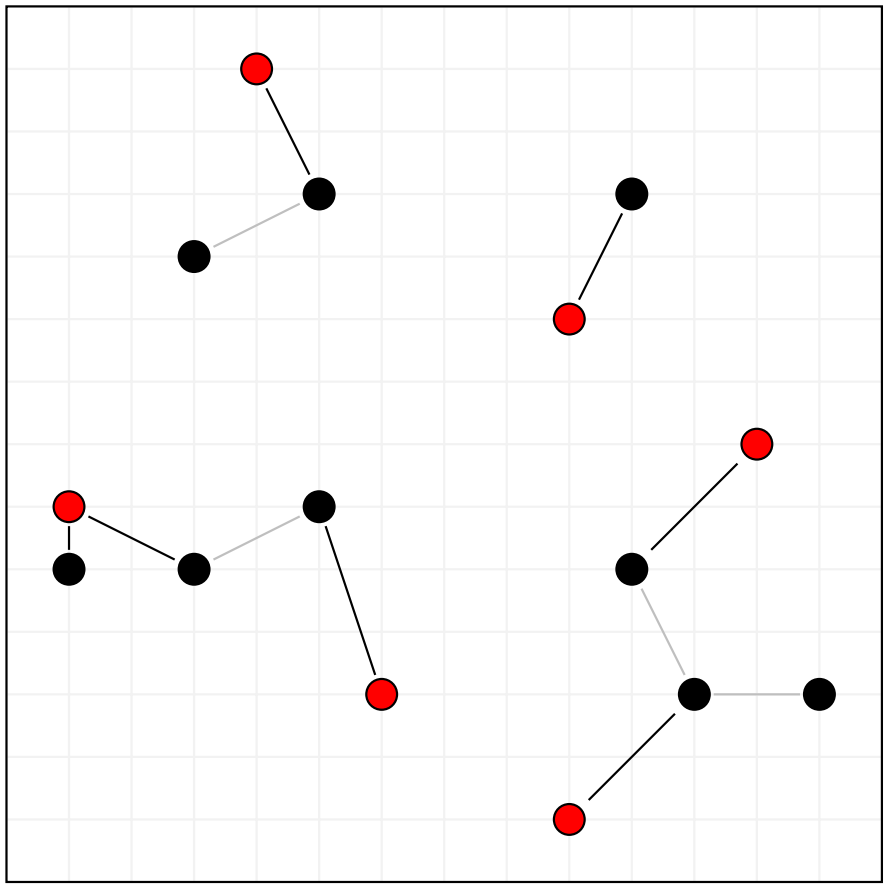
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



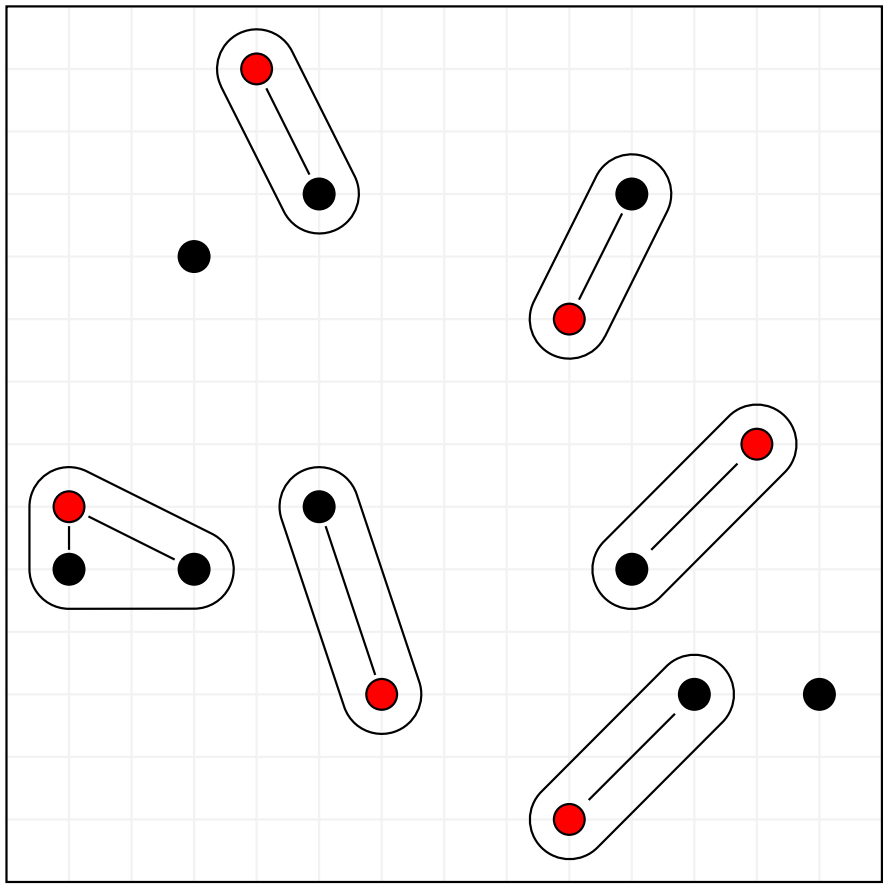
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



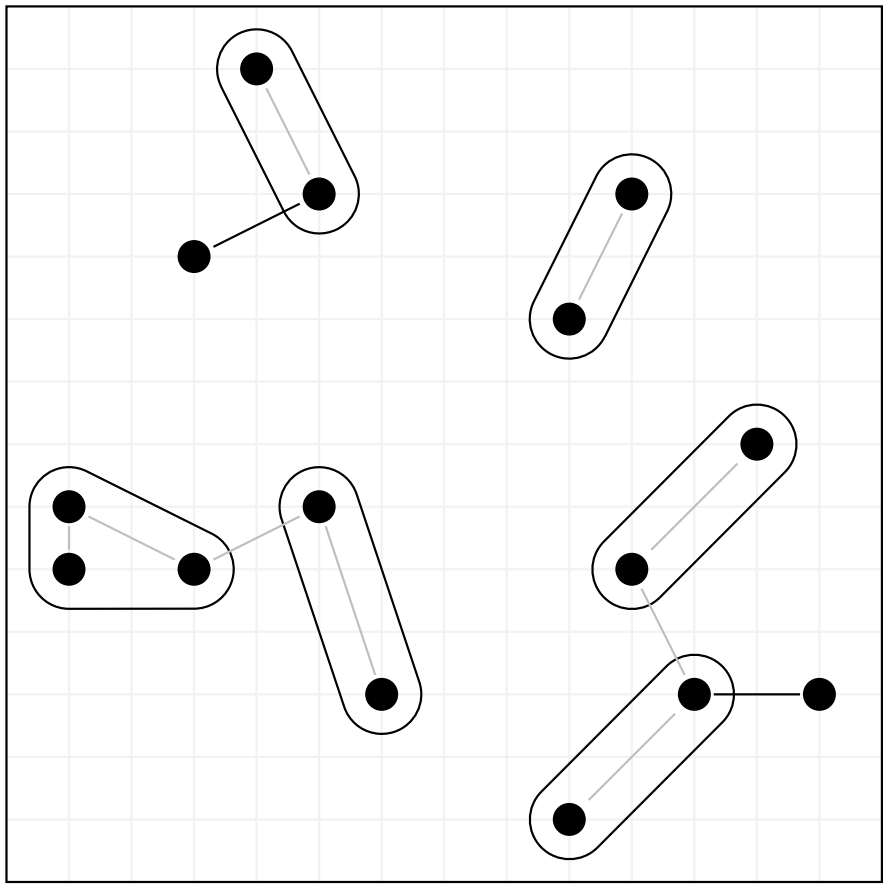
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



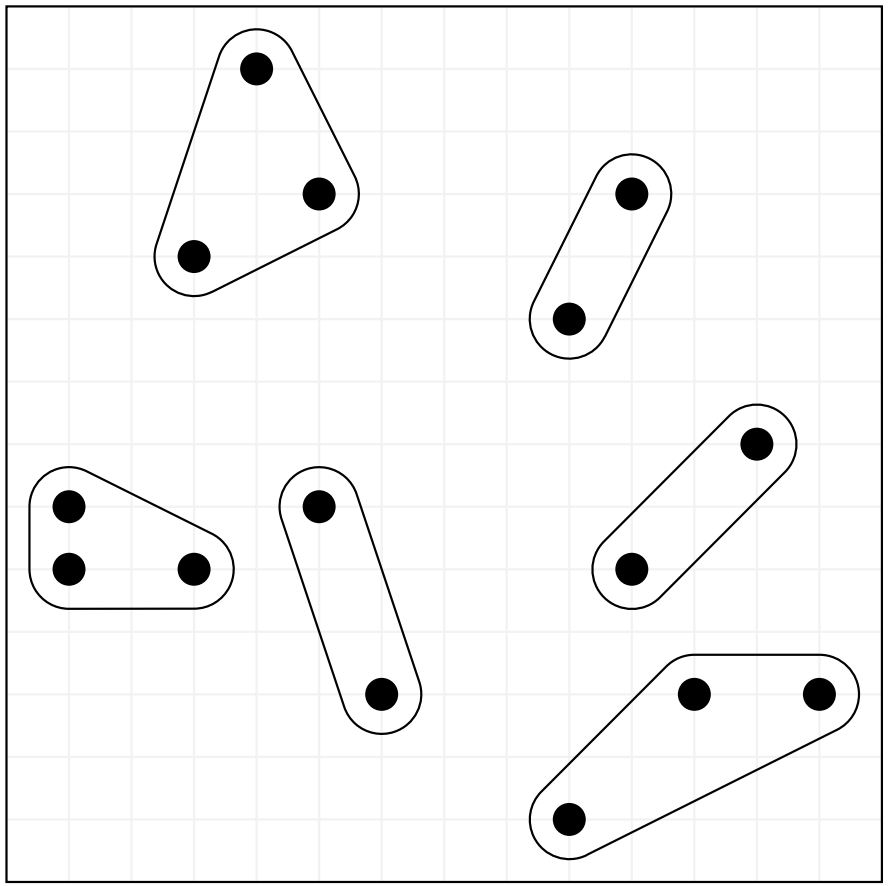
The AppOpt algorithm

Input:

- Units' covariates
- Distance metric
- Minimum block size: $k = 2$

Procedure:

1. A undirected complete graph with distances as edge weights
2. Find $(k-1)$-nearest neighbor graph
3. Construct the second power of NNG
4. Find a maximal independent set (seeds)
5. Form blocks with the seeds and their neighbors in NNG
6. Assign remaining units to a block containing any neighbor



Conclusion

- Closer to clustering than traditional blocking/matching methods
- Fast algorithm:
  - NNG plus $O(d^0 kn)$ time and $O(d^0 kn)$ space
  - K-d trees NN: $O(2^d kn \log n)$ expected time, $O(2^d kn^2)$ worst time, and $O(kn)$ storage
  - Compare with bipartite, network flow methods:
    - e.g., Derigs: $O(n^3 \log n + dn^2)$ worst time and $O(d^0 n^2)$ space

References

- Künzel, Barter, Sekhon, Bickel, and Yu. (on-going). "Estimation of heterogeneous effect via Random Forests."
- Higgins, Sävje, and Sekhon (forthcoming). "Improving Massive Experiments with Threshold Blocking." *Proceedings of the National Academy of Sciences*.
- Bloniarz, Liu, Zhang, Sekhon, Bin Yu. (forthcoming). "Lasso Adjustments of Treatment Effect Estimates in Randomized Experiments.". *Proceedings of the National Academy of Sciences*.
- Hartman, Grieve, Ramsahai, and Sekhon (2015). "From Sample Average Treatment Effect to Population Average Treatment Effect on the Treated: Combining experimental with observational studies to estimate population treatment effects" *Journal of the Royal Statistical Society, Series A*. 10.1111/rssa.12094.
- Miratrix, Sekhon, and Yu. (2013). "Adjusting Treatment Effect Estimates by Post-Stratification in Randomized Experiments." *Journal of the Royal Statistical Society, Series B (Methodology).* 75 (2): 369–396.
- Diamond and Sekhon. (2013). "Genetic Matching for Estimating Causal Effects: A General Multivariate Matching Method for Achieving Balance in Observational Studies" (with Alexis Diamond). *Review of Economics and Statistics.* 95 (3): 932–945.