Statistical Science
2020, Vol. 35, No. 3, 356–360
https://doi.org/10.1214/19-STS739
Main article: https://doi.org/10.1214/19-STS699

Comment: Matching Methods for Observational Studies Derived from Large Administrative Databases

Fredrik Sävje

I first want to commend Yu, Silber and Rosenbaum (2019) for their paper. The matching procedure they have developed will help countless researchers improve their causal inferences in settings where randomized experiments are infeasible or impractical but where observational data is plentiful.

The aim of my remaining comments is to extend and complement the authors' discussion. I start by comparing their procedure with similar approaches developed in the computer science literature. This broader perspective provides some suggestions for possible improvements and extensions. I continue with a discussion about how optimality may be viewed with respect to statistical performance, match quality and runtime, and I describe an alternative procedure that may better align with some of these objectives. I conclude with some general remarks. To the greatest extent possible, I use the authors' notation.

1. A BROADER PERSPECTIVE

1.1 Previous Work on the Matching Problem

In its most condensed form, the problem the authors consider is to find a μ in $\mathbb{M}$ that minimizes some objective function L where $\mathbb{M}$ collects all injective functions mapping treated units $\mathcal{T}$ to controls $\mathcal{C}$. A common choice for L is the sum of some cost function $\delta : \mathcal{T} \times \mathcal{C} \to \mathbb{R}^+$ over the matches, in which case the problem becomes

$$\mathbb{M}^* = \arg\min_{\mu \in \mathbb{M}} \sum_{t \in \mathcal{T}} \delta\big(t, \mu(t)\big).$$

The task is the same as finding a minimum-cost maximum independent edge set in the complete bipartite graph with $\mathcal{T}$ and $\mathcal{C}$ as parts.[1] Such an edge set can be found as the solution to a minimum-cost network flow problem.

Fredrik Sävje is Assistant Professor, Department of Political Science and Department of Statistics and Data Science, Yale University, Rosenkranz Hall, 115 Prospect Street, New Haven, Connecticut 06520, USA (e-mail: fredrik.savje@yale.edu).

[1] An independent edge set is called a "matching" in the computer science literature, but the term has an extended meaning in the causal inference literature.

Through an impressively productive research program, the authors and their collaborators have described how a large number of variations of the objective function and constraints on $\mathbb{M}$ also can be encoded as minimum-cost network flow problems. Many of these variations, such as fine balancing, are reviewed in detail by the authors.

The network flow approach admits great flexibility, but the associated algorithms do not scale sufficiently well to accommodate large samples. The authors note that runtime tends to grow as $\mathcal{O}(NE + N^2 \log N)$ where N is the number of vertices in the network and E is the number of edges. Generally in matching problems, $E = \Omega(N^2)$ and $N = \Omega(T + C)$, where Ω denotes asymptotic lower bounds, so the time complexity is cubic in the sample size.

One way to reduce runtime is to prune edges in E in a preprocessing step. For example, if one can achieve $E = \mathcal{O}(N \log N)$, runtime grows at only a quasi-quadratic rate. Such pruning must, however, be done with care. One potential problem is that the optimal flow derived from the reduced edge set may not be a maximum independent edge set in the full problem; that is, the matching produced from the pruned edge set may not be in $\mathbb{M}$. A second concern is that the solution might not be an optimal solution in the full problem; that is, the matching may not be in $\mathbb{M}^*$.

The authors set out to develop a procedure to prune edges while ensuring that the network flow solution is in $\mathbb{M}$. To that end, they solve another optimization problem:

$$\mathbb{M}^{\mathrm{B}} = \arg\min_{\mu \in \mathbb{M}} \max_{t \in \mathcal{T}} \delta\big(t, \mu(t)\big).$$

Problems that aim to minimize the maximum edge cost are called *bottleneck problems*. Bottleneck problems rarely have unique solutions. This does not concern the authors, however, because they seek a preprocessing step. With $\mathbb{M}^{\mathrm{B}}$ in hand, they substitute it for $\mathbb{M}$ in the original problem and find a μ in $\mathbb{M}^{\mathrm{B}}$ that minimizes L. Because $\mathbb{M}^{\mathrm{B}}$ is smaller than $\mathbb{M}$, the procedure will reduce runtime as long as the preprocessing can be completed quickly. Furthermore, because $\mathbb{M}^{\mathrm{B}}$ is a subset of $\mathbb{M}$, the matching found in this way will be admissible.