Efficient Balanced Treatment Assignments for Experimentation

David Arbour
Adobe Research

Drew Dimmery
Facebook

Anup Rao
Adobe Research

Abstract

In this work, we reframe the problem of balanced treatment assignment as optimization of a two-sample test between test and control units. Using this lens we provide an assignment algorithm that is optimal with respect to the minimum spanning tree test of Friedman and Rafsky [1979]. This assignment to treatment groups may be performed *exactly* in polynomial time. We provide a probabilistic interpretation of this process in terms of the most probable element of designs drawn from a determinantal point process which admits a probabilistic interpretation of the design. We provide a novel formulation of estimation as transductive inference and show how the tree structures used in design can also be used in an adjustment estimator. We conclude with a simulation study demonstrating the improved efficacy of our method.

1 Introduction

Decision-making often requires engaging with counterfactual questions. For instance, determining whether to give a patient a medication depends on what their health outcomes *would have been* absent the medication. One of the most successful tools for answering these types of counterfactual questions has been experimentation. For a sample of patients, randomly give half of them the medication and half of them a placebo, and measure the *average* health outcomes for each of the two groups. This provides unbiased estimates of the typical response in the sample: the average treatment effect (ATE) [Imbens and Rubin, 2015]. This does not address a doctor's most fundamental concern, however: how would *this* patient respond to treatment, relative to their counterfactual health outcomes under placebo? To answer this question, it is necessary to consider *conditional average treatment effects* (CATE) [Athey and Imbens, 2016]. While the literature has provided many improvements around the design of experiments to measure the former quantity (the ATE), in this paper, we analyze the problem of experimental design for estimation of the CATE.

This work is concerned with extending the capabilities of experimental design along two axes:

Experimental Design for Heterogeneous Treatment Effects. To our knowledge, this is the first work focused on design-based solutions to the estimation of CATEs.

Computationally Efficient Exact Solutions. Both mean (and kernel mean) based measures of imbalance and blocking are NP-hard to optimize [Kallus, 2018, Higgins et al., 2016].

Our primary contributions are:

- Motivate the estimation of CATEs around transductive learning.
- Show that the problem of good experimental design is closely related to a ubiquitous graph-cutting problem through a bias-variance decomposition of the design problem.
- Reorient the problem of balance around a two-sample test between treatment and control covariate profiles.
- Provide an efficient approximation to this problem based on maximum spanning trees, which optimizes a ubiquitous graph-based two-sample test and provides highly accurate estimates of CATEs.

The structure of this paper is as follows. Section 2 describes the problem of experimental design, and estimation of CATEs given a design. Section 3 provides an overview of pre-existing work on experimental design. Section 4 presents the problem of CATE-optimizing experimental design, connects it to graph cutting and discusses existing approaches through this lens. Section 5 presents our proposed design which optimizes test of balance based on the minimum spanning tree.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

Code available at https://github.com/ddimmery/softblock.

Section 6 shows a bevy of simulation evidence demonstrating the strength of our proposed design.

2 Background and Problem Description

We first give some background and notation before introducing the task of this work. Throughout we will consider three sets of variables, $\mathbf{X} \in \mathbb{R}^D$, $A \in \{0,1\}$ and $Y \in \mathbb{R}$. We assume that $\mathbf{X}$ is pretreatment, i.e. the values are not caused by A or Y. We will also assume that Y is given as some function of X, A, and mean-zero noise. Given a set $i = 1, \ldots, N$ of realizations of Y and A the potential outcomes [Rubin, 2011], $Y^{A=0}$, $Y^{A=1}$ are the values of Y that would have been observed had treatment been observed at $A = 1$ or $A = 0$, respectively. For some mathematical statements it is more convenient to annotate treatment as being in $\{-1, 1\}$, and we will indicate this by the use of a vector $\mathbf{u}$, wherein -1 notates control and 1 indicates treatment. Causal effects are then, in turn, derived as contrasts between potential outcomes. In this paper, we will consider two causal estimands:

- The *Conditional Average Treatment Effect (CATE)* is the conditional effect of treatment, CATE $= \mathbb{E}\left[Y^{A=1} - Y^{A=0} | \mathbf{X} = \mathbf{x}\right]$.
- The *Average Treatment Effect (ATE)* is an estimate of the marginal effect of treatment from a finite sample. The ATE is easily expressed as an expectation of the CATE, $\int_{\mathbf{x} \in \mathbf{X}} \mathbb{E}\left[Y^{A=1} - Y^{A=0} | \mathbf{X} = \mathbf{x}\right] p(\mathbf{x}) d\mathbf{x}$.

Optimal experimental design—the central task of this paper—considers the following problem. Given the set of pre-treatment covariates $\mathbf{X}$, how should treatment be assigned to each individual in order to to obtain an unbiased estimate of a causal estimand with minimal variance? Optimal experimental design for estimating the ATE has been studied for decades (c.f., Fisher [1935], Morgan et al. [2012], Hall et al. [1995], Kallus [2018], Higgins et al. [2016]). In the general setting, Kallus [2018] showed that complete randomization is minimax optimal. However, with additional assumptions placed over the potential outcomes, improvement can be made through careful allocation. One such assumption, which we will employ throughout the remainder of the paper, is that the potential outcomes are smooth functions with additive noise. More precisely, we introduce the following assumptions

Assumption 1. *The pre-treatment covariates, $\mathbf{x}$, belong to a metric space, with the corresponding metric denoted $d(x, x')$, and are drawn from some distribution $p(x)$ with finite variance. In this paper, we assume that x_i is drawn from some (possibly unknown) distribution $p(x)$ and that the domain is a metric space, $\mathcal{X}$ with the metric d_X.*

Assumption 2. *Each of the potential outcomes, are drawn from the following generative process*

$$f^{A=a}(\mathbf{X}_i) = \mathbb{E}\left[Y_i^{A=a} | \mathbf{X}_i = x\right]$$
$$\epsilon_i^{A=a} = Y_i^{A=a} - f^{A=a}(\mathbf{X}_i)$$

Where $\epsilon^{A=a}$ is mean zero.

Assumption 3. *Each potential outcome function $f^{A=a}(x)$, $a \in \{0,1\}$, is Lipschitz continuous with Lipschitz constant, L.*

3 Balance in existing designs

There have been a plethora of design procedures that attempt to explicitly improve balance. These approaches fall into three primary camps:

- **Blocking** [Greevy et al., 2004, Higgins et al., 2016]. Units are divided into a partition and then a fixed number of units are randomly given treatment within each stratum. This ensures that treatment is balanced on stratum indicators.
- **Rerandomization** [Morgan et al., 2012, Li et al., 2018]. Units are assigned completely randomly to treatment, then balance is checked. If imbalance is too high, then randomization is performed again. This process is repeated until imbalance is below some a priori specified level.
- **Optimization** [Kallus, 2018, Harshaw et al., 2020]. An optimization procedure is used to find the best vector of assignments to treatment in order to minimize some measure of imbalance. This assignment may be deterministic.

These approaches can be difficult to scale to the necessary sample sizes for the online environment, as finding optimally balanced treatment assignments is an NP-hard problem.

The optimization objective most commonly employed for optimal experimental design is mean balance (c.f. Morgan et al. [2012], Kallus [2018]), i.e., minimizing the distance in means between the instances of $\mathbf{X}$ that are allocated to treatment and control, respectively. This measure can be extended to incorporate higher order and non-linear dependencies by applying a feature transformation, ϕ, to the covariates. The resulting optimization problem is then given by

$$\min_{\mathbf{a} \in \{0,1\}} g\left(\mathbf{a}\phi(\mathbf{X}_i), (1-\mathbf{a})\phi(\mathbf{X}_i)\right)$$

where $g(\cdot)$ is a distance function. Popular choices for $g(\cdot)$ are Euclidean [Hansen and Bowers, 2008], and Mahalanobis [Morgan et al., 2012] distance. Of particular interest to this work is the balance measure used by Kallus [2018] which considers the mean difference

between $\mathbf{X}$ after projecting the covariates in to a reproducing kernel Hilbert space (RKHS). The optimal experimental design in this setting corresponds to solving the following binary quadratic program, termed the pure strategy optimal design (PSOD) by its author,

$$\min_{\mathbf{u}\in\{-1,1\}} \frac{4}{N^2} u^T \mathbf{K} u \tag{1}$$

where $\mathbf{K}$ is the Gram matrix of $\mathbf{X}$ with respect to an RKHS. Under smoothness assumptions on the potential outcome function, the solution to PSOD was shown to be Bayes optimal, with variance guarantess comparable to those provided by post-hoc regression adjustment.

While mean balance has intuitive and theoretical appeal, it also comes with significant computational disadvantages. Kallus [2018] shows that PSOD, which accomodates a large number of mean balance measures, is equivalent to solving the balanced number partition problem which is known to be NP-hard. The implemented solution requires solving a semi-definite program which prevents the applicability of the method to moderately large domains (in the hundreds to thousands).

4 Experimental Design for CATE

Design for the average treatment effect has received considerable attention in the literature. Less studied, however, is design specifically targeting conditional average treatment effects. Recently, this quantity has gained substantial attention due to Athey and Imbens [2016], Wager and Athey [2018] and the broader literature around conditional average treatment effect estimation [Shalit et al., 2017, Shi et al., 2019][1]. The CATE is given by the difference in potential outcomes conditioned on x, $Y^{A=0}(x) - Y^{A=1}(x)$, $x \in X$. The central task considered within this paper is allocating treatment to estimate the CATE well (we will make this statement more formal shortly).

To motivate our design task, we begin with an estimator for the conditional average treatment effects. We will restrict ourselves to distance based regression functions,

$$\hat{f}(x_i) = \sum_j^N w_{ij} y_j \qquad s.t. \sum_j^N w_{ij} = 1 \tag{2}$$

Special cases of this general formulation are k-nearest neighbors regression as well as Nadaraya-Watson kernel regression. These estimators are non-parametric and fairly flexible. We focus on this estimator due to its analytical tractability in combination with its generally reasonable performance as a non-parametric estimator. As we will show in section 6, a design which is effective for this estimator will typically also be effective for other CATE estimators. Under the assumed model, the empirical estimate of the CATE can be written as

$$\hat{\tau}_i = (2a_i - 1)\left(y_i - \sum_{j=1}^n w_{ij} y_j\right). \tag{3}$$

Equation 3 can be interpreted as two independent regressions inferring the potential outcomes of $Y^{A=1}$ and $Y^{A=0}$, where the predictions for observed potential outcomes are constrained to be equal to the observed outcome. In the conditional average treatment effect estimation literature, training an outcome for each potential outcome surface is often referred to as a "T-learner" [Künzel et al., 2019], but due to our restriction that observed potential outcomes take their observed values, our approach is more similar to the "X-learner" of Künzel et al. [2019]. This restriction is also often employed in the tranductive learning setting, for example, by Zhu et al. [2003]. Framed in terms of transductive inference, our task of CATE estimation is to impute the counterfactual for each unit, and this imputation of counterfactuals is the only way that error is introduced into our estimation problem.

To our knowledge, this paper is the first to examine designing an experiment explicitly for the estimation of CATEs. We do so by viewing experimental design as an optimal graph cut problem. We discuss the details of the connection between graph cutting and experimental design for CATE estimation next.

4.1 Graph cutting in experiments

A natural interpretation of the assignment problem is to view the observations of covariates as nodes in a graph with treatment being an indicator of missingness. Through this lens we see that the task of treatment assignment can be interpreted as minimizing the risk of two interrelated regression problems: predicting the control counterfactual for treatment using only control units, and predicting the treated counterfactual for control using only the treated units. The resulting optimization problem is then given as

$$\min_A \sum_i^N \left| \sum_j \frac{\mathbb{1}(a_i \neq a_j) e_{i,j}}{\sum_k \mathbb{1}(a_i \neq a_k) e_{i,k}} y_j - y_i \right| \tag{4}$$

where we refer to similarity between points as $e_{i,j}$ and replace $w_{i,j}$ with a more explicit expression. Note that the choice of similarity function, as before, is a design

[1] In the machine learning literature this is sometimes referred to as "individual treatment effect" estimation.

choice made by the practitioner. As with most causal inference applications, the outcomes are unobserved which can make reasoning over design choices difficult a priori. However, after leveraging the Lipschitz assumption (assumption 3) the following proposition allows for a bound on the bias and variance of the regression function.

Proposition 1. *[Anava and Levy, 2016] The bias and variance of an estimate for any one point, $x^\star$ is bounded by*

$$\left|\sum_{i=1}^{n} w_{\star,i} y_i - f(x_\star)\right| \leq \underbrace{C\|\mathbf{w}_\star\|_2}_{variance} + \underbrace{L\sum_{i=1}^{n} w_{\star,i} d(x_i, x_\star)}_{bias}$$

with probability $1-\delta$, where L is the Lipschitz constant as in assumption 3, d is a distance measure, $C = b\sqrt{2\log(\frac{2}{\delta})}$, and b upper bounds noise, i.e., $|\epsilon| \leq b$.

A proof is provided in the supplement for completeness. Proposition 1 provides an expression for the error which relies only on observable quantities, namely the distance between treatment and controls and the regression weights, and an assumption on the magnitude of noise. The optimization problem in equation 4 can then be recast as

$$\min_{A} \sum_{j}^{n} C\|\mathbf{w}_j\|_2 + L\sum_{i=1}^{n} w_{j,i} d(x_i, x_j)$$

with $w_{j,i} = \frac{\mathbb{1}(a_i \neq a_j) e_{i,j}}{\sum_k \mathbb{1}(a_i \neq a_j) e_{i,k}}$ as in equation 4. This lens makes explicit the tradeoffs between bias and variance in the design. It should come as no surprise that the optimal design will be heavily reliant on the distribution of $\mathbf{X}$ and the magnitude of the noise term, i.e. the size of b. For example, on one extreme when b is close to zero, then the best choice will be to concentrate all of the weight on the first nearest neighbor. As we discuss in section 4.3, this corresponds to a greedy design which two-colors a one-nearest neighbor graph. More generally, it is necessary to reason over trade-offs that are occurring with respect to the experimental design.

In this work, we propose to view these choices by recasting the problem of experimental design in terms of graph cutting. Specifically, we consider a graph, G where the edge weights, $e_{i,j}$ are the similarity between $\mathbf{x}_i$ and $\mathbf{x_j}$. After remapping treatment to $\{-1, 1\}$ via $\mathbf{u} = 2\mathbf{a} - 1$, the problem of treatment assignment can be recast as choosing an assignment. This view is quite natural, since the set of cut edges, i.e., edges where $a_i \neq a_j$ are those which are used to infer the counterfactuals in the nearest neighbor regression. The following proposition makes this more formal by relating the risk of the regression estimator to the Maxcut problem

Proposition 2.

$$\sum_{i}^{N}\left|\sum_{j} w_{i,j} y_j - y_i\right| \leq \sum_{i} \epsilon_i + \frac{e_{sum} - \sum_{i,j} \mathbb{1}(a_i \neq a_j) e_{ij}}{d_{\min}}$$

Where $e_{sum} = \sum_{i,j} e_{i,j}$, and $d_{\min} = \min_i \sum_j e_{i,j}$.

The proof is provided in the supplement. The first term is an irreducible component which corresponds to the estimation error due to non-smoothness of the potential outcome function. It shows the error under an oracle scenario in which the unobserved potential outcome for a unit is estimated based on the potential outcomes for *all* other units. It further presumes this estimation is performed for every unit, which is not possible due to the fundamental problem of causal inference. This represents the Bayes risk of the estimation problem: the lower bound of the error incurred for this estimation. The second term is more interesting, as it describes the error due to the assignment process we choose. While e_{sum} and $d_{\min}$ do not depend on the assignment (and therefore optimal design need not incorporate them), the remaining piece does. This term is the negative of the objective of the Maxcut graph-cutting problem, which we now describe in greater detail.

4.2 Maxcut

First, informally: maxcut divides the nodes of a graph into two disjoint and exhaustive subsets by removing ("cutting") edges with the maximum edge-weights.

A common way to write this is through the use of the graph Laplacian:

$$\max_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^T L \mathbf{u} \tag{5}$$

where $\mathbf{u}$ corresponds to which set each node belongs to, denoted -1 and 1. The graph Laplacian is a matrix which represents the structure of the network, formed as the diagonal matrix of node-degree minus the incidence matrix of edge weights, $D - G$. Maxcut is a canonical NP-hard problem and, is not amenable to a polynomial-time approximation scheme unless the unique games conjecture is true [Khot et al., 2007, Goemans and Williamson, 1995]. Common approximation algorithms include semidefinite programming [Goemans and Williamson, 1995, Trevisan, 2012]. The best known approximation ratio for this problem, in general, is through semidefinite programming, with a ratio of $\frac{16}{17} \approx 0.941$. Given the difficulty of this problem, it is not possible to uniquely minimize Proposition 2 in polynomial time, so we will focus only on efficient algorithms for the computation of a design. Note that the kernel allocation procedure proposed

by Kallus [Kallus, 2018] is isomorphic to the Maxcut problem. We provide a proof of this correspondence in the supplement.

Certain special cases, however, allow for efficient solutions to Maxcut. Among these are particular bipartite graphs such as forests and trees. These graphs, for instance, admit solutions to Maxcut in *linear* time.

4.3 Optimal Matched Pair Designs

Propositions one and two can help shed light on common experimental designs, such as the matched pair design [Imai, 2008].

Kallus [2018] demonstrates that, when outcomes are Lipschitz, implementing this design by finding the max weight matching is optimal. This can be efficiently implemented using, e.g. the Edmonds' algorithm [Edmonds, 1967]. This optimality result, however, restricts the set of designs to those which may be defined as a matching on the graph. A graph matching, of course, may have no two edges which share an end-point. Our result demonstrates that a wider class of designs may be considered, opening the door to stronger assignment mechanisms.

In the observational literature on matching methods, there is a distinction drawn between greedy and so-called "optimal" matching [Stuart, 2010, Hansen and Klopfer, 2006, Parikh et al., 2018]. The distinction being that a greedy matching algorithm can "double dip," using the same unit as the matched control for multiple treated units. The experimental design based on optimal matching is the Kallus [2018] matched pair design, but we can similarly form a greedy design by two-coloring the one-nearest neighbor graph. The decomposition of Proposition 1 gives us a ground on which to compare these designs. The greedy design ensures minimal pointwise bias for the CATE by minimizing the distance to a match. While providing the minimum pointwise bias of the design, the variance properties are not so clearcut. Depending on specific properties of the data, either the greedy design or the matched-pair design could be lower variance.

4.4 On Optimal Designs

We now turn to the question: can we construct a feasible "optimal" design? To provide a specific example from the previous section, how should practitioners decide between greedy designs (which may imply higher leverage for certain observations) and non-greedy designs (which may imply higher bias for the imputation of CATEs for some units)? This question does not have easy answers. Indeed, a simple example can illustrate this conundrum. Suppose a graph with one point in the center in two dimensions with $n-1$ points surrounding it in a circle. Further suppose that each of these points is r units away from the centerpoint, but $s > r$ units away from the next closest point on the exterior. Assume that each unit has a residual, ϵ (as in Assumption 2), which is drawn from a mean zero normal distribution with standard deviation σ_c for the center point and σ_e for exterior points. The greedy design would ensure that the center and the exterior points received different treatments. The matched pair design would pair the center with one random exterior point, and then match all other exterior points with a neighboring exterior point. Then we can write the expectation of the bound in equation 2 for the center point in the matched-pair design as $\sigma_e + Lr$. For one point in the exterior, that quantity is $\sigma_c + Lr$, while for all others it is $\sigma_e + Ls$. For the greedy design, this quantity would instead be $\frac{\sigma_e}{n-1} + Lr$ for the center point. For all exterior points, the bound would be $\sigma_c + Lr$.

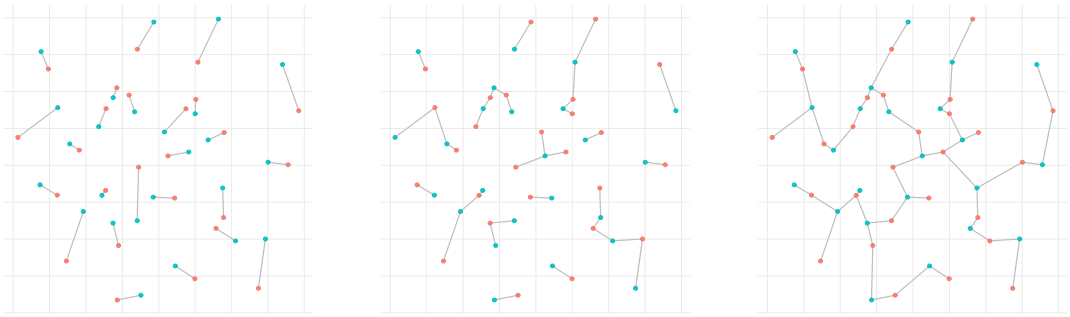
Depending on the relative values of σ_c versus σ_e and the distance s versus r, either the greedy design or the matched-pair design could minimize the bound. That is, if σ_c is very large, then the greedy design will tend to exhibit variance properties that overwhelm its low bias. Similarly, if σ_e tends to be larger (or $s \gg r$), then the matched-pair design will tend to have unacceptably large biases that will overwhelm its variance properties. Of course, in the asymptotic regime, only bias matters and thus the greedy design will minimize this bound. In finite samples, this thoroughly unsatisfying bias-variance tradeoff demonstrates that the optimal design depends crucially on properties of the data which are unknowable a priori. In short, we do not seek an optimal design, but instead simply designs that make a reasonable tradeoff between bias and variance for many applied situations.

Practitioners who understand more about their data, such as the extent of heteroskedasticity and the smoothness of the conditional expectation function can therefore make better decisions about design than any overarching theoretical statement that we can provide here.

5 Novel Designs through Cutting Spanning Trees

We begin by limiting our space of algorithms to scenarios in which Maxcut can be efficiently solved. Since trees and forests admit linear-time solutions to Maxcut, we focus on them.

In proposition 2, it is clear that integrated absolute bias is minimized when, for each unit, the similarity to the units with positive weights (i.e. the impute coun-



(a) Matched Pairs (b) Greedy Neighbors (c) SoftBlock

Figure 1: Three different designs on the same data.

terfactual) are maximized, as discussed in Section 4.3. The easiest way to ensure this is to match each unit with its closest neighbor in the graph and ensure that each neighbor in this newly sparsified graph receives different treatments than its neighbors. This solution is where we begin; we call this design "GreedyNeighbors", because the design is realized by solving Maxcut *exactly* on the one-nearest-neighbor graph. The nearest neighbor graph can be computed efficiently in $\mathcal{O}(n \log n)$ time by using a *kd*-tree. The one-nearest-neighbor graph is a forest, so solving Maxcut is trivially accomplished in $\mathcal{O}(n)$ by greedily walking the forest alternating treatment assignment. Thus, this design is realizable in aggregate $\mathcal{O}(n \log n)$ time. An important thing to note about this design in contrast to typical matched-pair designs is that a unit may be "matched" to more than one unit. Note that there are many realizable assignments with the GreedyNeighbors design, as each disconnected subgraph of the nearest-neighbor graph is assigned independently. This implies that there are 2^M possible assignments, where M is the number of disconnected subgraphs of the nearest-neighbor graph.

Algorithm 1: Deterministic Friedman-Rafsky Minimizing Design

input : $X \in \mathbb{R}^D$
output: Assignments A, Spanning Tree T
$G \leftarrow$ Similarity matrix constructed from X
$T \leftarrow$ Maximum Spanning Tree(G)
$A \leftarrow$ MAXCUT(T)

This design, however, despite minimizing bias on the CATE estimates, is needlessly high variance. Each added edge will stabilize the variance component in the decomposition in proposition 1. Thus, adding edges will reduce the variance of the ultimate solution (at the expense of some additional possibility for bias). We propose a design which manages this tradeoff in a computationally tractable way based on the maximal spanning tree of the original similarity graph. Algorithm 1 summarizes this design. In short, the maximal spanning tree (MST) is the largest tree over the graph which contains no loops or cycles. The maximal spanning tree always contains the nearest neighbor graph (as in, all edges of the nearest neighbor graph also are within the maximal spanning tree). Since the MST is a tree, it can also be solved trivially by Maxcut in $\mathcal{O}(n)$. The MST itself can be computed in $\mathcal{O}(n \log n)$. Thus, the full procedure requires, again, only $\mathcal{O}(n \log n)$ time complexity. Adding any additional edge to the MST which fails to preserve the bi-partiteness of the graph will make it no longer amenable to a greedy solution to Maxcut. This makes it the largest graph (in terms of total edge-weight), for which Maxcut is *necessarily* able to be efficiently solved. We refer to this algorithm as "SoftBlock", since it softens the idea of a blocked design by allowing for substantial correlations between any two units (rather than simply units which lie within the same block).

In Figure 1, a fixed set of covariate observations in $\mathbb{R}^2$ are assigned treatment according to three different methods. As can be seen, the graph undergirding the Greedy Neighbors design is nested within the MST of SoftBlock. The distinction between Matched Pairs and the Greedy Neighbors design is that all nodes in the former have degree one, while the latter design has no such restriction.

5.1 Probabilistic Interpretation

We now provide a probabilistic interpretation of the proposed design. Starting with the observation that the set of all random spanning trees defines a determinantal point process (DPP) where the probability of a spanning tree is proportional to the product of its edge weights [Lyons and Peres, 2017], i.e. $p(T) \propto \prod_{i,j \in E(T)} e_{i,j}$. This can be trivially modified to represent a distribution where each tree's probability is given by its respective balance by first consid-

ering an exponentiation of the weights, i.e., $p(T) \propto \prod_{i,j \in E(T)} \exp(e_{i,j}) = \exp\left(\sum_{i,j \in E(T)} e_{i,j}\right)$.

It's easily observed that when the sum of the weights is maximized (that is, the MST), the probability of the tree is also maximized. Thus, SoftBlock, the design based on the MST, is the MAP estimate from this DPP.

5.2 Balance and Graph Two Sample Tests

All designs we have considered correspond to a particular test of balance between treated and control units. For example, rerandomization using the Mahalonobis distance minimizes a t-test, and as we detail in the appendix, problem 1 corresponds to minimizing an uncentered version of maximum mean discrepancy [Gretton et al., 2012].

As it turns out, SoftBlock shares an interesting connection to the minimum spanning tree test of Friedman and Rafsky [1979]. Specifically, the graph based test addresses the problem of detecting differences between two distributions by viewing the problem in terms of a cut on a minimum spanning tree. The procedure is as follows. The two samples $\mathbf{X}_0$, and $\mathbf{X}_1$ are pooled and a similarity graph, $\mathcal{G}$ is constructed according to an analyst specified similarity metric. The minimum spanning tree, $\mathcal{T}$ for $\mathcal{G}$ is then found. The test statistic is defined as the number of edges in $\mathcal{T}$ that connect samples from $\mathbf{X}_0$ and $\mathbf{X}_1$, i.e., $\text{FR}(\mathbf{X}_0, \mathbf{X}_1) = \sum_{\mathbf{x}_i \in \mathbf{X}_0, \mathbf{x}_j \in \mathbf{X}_1} \frac{\mathbb{1}(\mathbf{x}_i, \mathbf{x}_j) \in E(\mathcal{T})}{N-1}$, where $E(\mathcal{T})$ are the set of edges in the minimum spanning tree, $\mathcal{T}$, and N are the total number of samples in the pooled dataset. The test is minimized if the two samples share only one edge in the spanning tree, and maximized when edges connect units from different samples as much as possible. This procedure was shown to be asymptotically normal and consistent by Henze et al. [1999]. The SoftBlock assignment mechanism directly minimizes the Friedman-Rafsky test statistic. By optimizing a consistent test of balance, our procedure asymptotically guarantees balance on covariates between groups. Given that this is a consistent test, we can be sure that even though we aren't directly optimizing linear balance, we will converge to linear balance in the limit. In finite samples, this procedure may sacrifice some degree of linear balance relative to traditional blocking procedures. Essentially, linear balance implies a computationally intractable (i.e. NP-hard) exact solution, while the use of a different metric of balance provides a simple polynomial time algorithm (with equivalence to the linear problem in the limit).

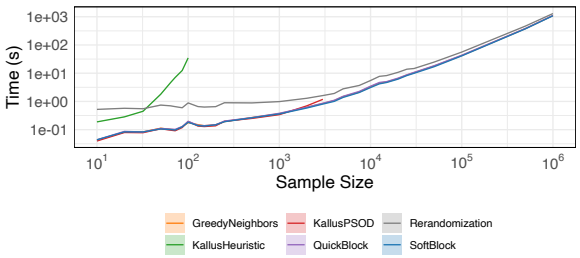


Figure 2: Runtime of various methods for experimental design. Both axes are logarithmic.

6 Experiments

In this section, we present experiments demonstrating the effectiveness of SoftBlock. We begin by describing the methods we benchmark against:

- Bernoulli randomization. This method flips a fair coin for each unit. This method is minimax optimal for the ATE as per Kallus [2018].
- Rerandomization. The method of Morgan et al. [2012] randomizes, checks balance (by Mahalanobis distance) and, if it's too high, repeats. In our implementation, we use the heuristic of Kallus [2018], which accepts a randomization with only 1% probability. Thus, it ensures that the chosen design has one of the 1% most balanced designs (in terms of Mahalanobis distance).
- QuickBlock. The method of Higgins et al. [2016] finds an approximate blocking solution based on a k nearest neighbor graph. They find that it performs comparably or better to Greevy et al. [2004] "optimal" blocking.
- Kallus' PSOD and Heuristic Designs. These designs of Kallus [2018] optimize assignments to minimize mean imbalance in an RKHS.
- Optimal matched pair designs. These designs solve the maximum weight matching problem and then randomize which unit in each pair receives treatment [Kallus, 2018, Imai, 2008].

We consider a variety of linear and non-linear data generating processes, defined in detail in Table 1 in the Appendix. The QuickBlockDGP was the primary simulation used in Higgins et al. [2016], consisting of four uniform random variables multipled together. We additionally provide a simulation with a linear outcome, one based on a sinusoid, and one with covariates distributed along two circumscribed circles.

Figure 2 shows the runtimes of these various methods on the TwoCircles data generating process. At very low sample sizes, Kallus's [2018] PSOD method is the fastest way to design an experiment and estimate effects, but by moderate sample sizes is out-

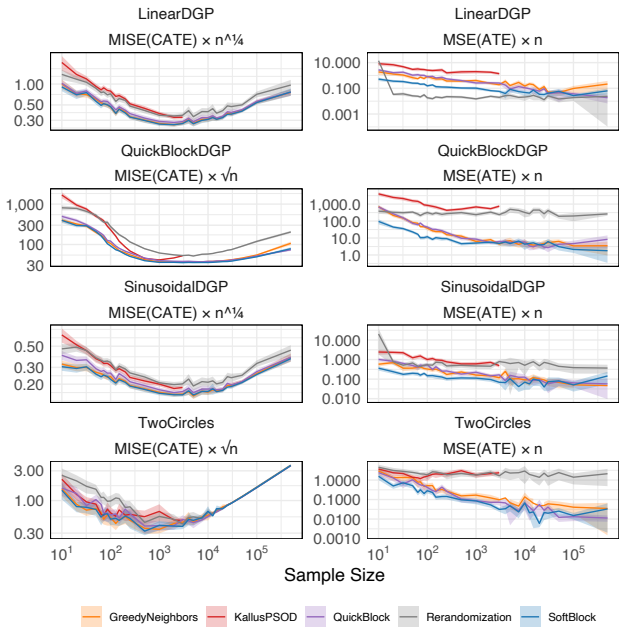


Figure 3: SoftBlock performs well across a wide-array of data-generating-processes and sample sizes. These plots display mean squared-error and mean integrated-squared-error for the ATE and CATE, respectively. These values are multiplied by a function of sample size for ease of comparison.

paced by QuickBlock, SoftBlock and the GreedyNeighbors methods. SoftBlock is faster than QuickBlock at nearly all sample sizes, but the two approaches increase computational time at a similar rate.

Figure 3 shows the performance of the methods on the simulation setups in Table 1. Note that values in this chart are normalized for sample size (errors are multiplied by $n^{1/\kappa}$ to allow for easier comparison across a wide array of sample sizes). In the LinearDGP, the methods which estimate the ATE with Lin's [2013] regression-adjustment method are, in fact, *correctly specified* parametric models. As such, they (Rerandomization, Greedy Nearest Neighbors and Bernoulli randomization) have much lower error than competitor methods. SoftBlock, however, converges to nearly the same error by around $n = 10000$. In the LinearDGP, SoftBlock and Greedy Nearest Neighbors are substantially more effective at estimating the CATEs than competitor methods, with SoftBlock outperforming Greedy Nearest Neighbors. Similar patterns hold in terms of the CATE on all DGPs, with QuickBlock performing the closest to SoftBlock, particularly at higher sample sizes. For estimating the ATE on the non-linear DGPs, SoftBlock is nearly always the most effective method, often substantially so, for example in moderate sample sizes on the QuickBlockDGP. The comparison between the GreedyNeighbors design and SoftBlock is informative, since the MST always contains the nearest neighbor graph. SoftBlock has two main advantages over this design. First, it reduces variance by using more than just the closest neighbor (for instance, sometimes the two nearest neighbors are both very close, so it would be wise to use both of them). Second, by being a single connected graph, it ensures that the assignments across different pairs of nearest neighbors are "lined up". That is, it avoids certain bad randomizations, in which, for example, two nearby edges are oriented in the same direction wherein the unit with larger covariate value is assigned treatment in both pairs. The cut on the MST, on the other hand, is more likely to insulate against this eventuality by connecting these subgraphs and ensuring the orientation of treatments do not match.

Figure 4 shows the performance of the design-based estimators for CATEs. In contrast to the previous figure, which estimates CATEs with a random forest T-learner, this figure shows the CATEs estimated by only the specific estimator implied by the design. This means that, for a blocking estimator, a difference-in-means estimator is used within each block to impute conditional effects (which are assumed constant within blocks). For SoftBlock, the CATE estimator is the difference of the observed ego unit and its synthetic counterfactual constructed by the weighted average of its neighbors in the minimum-spanning-tree as analyzed in section 4.1. In this comparison, SoftBlock performs substantially and consistently better than other designs. The comparison to blocking methods in this experiment demonstrate why SoftBlock is able to do better at estimating the CATE than other methods: it is optimized to ensure good interpolation across the entire space. In particular, we can once again see as informative its comparative stability relative to the matched-pair designs (note that the Kallus [2018] matched-pair design is infeasibly slow to display above sample sizes of 100 in this simulation).

Figure 5 shows the performance of various methods on the IHDP simulation study, as introduced by Hill [2011]. We compare using setting "B", in which the outcome model is nonlinear and the treatment effect is not constant. In this data, SoftBlock provides the lowest error estimates of the ATE, and all methods tend to perform well for estimating the CATE with a random forest T-learner.

In the appendix, figure 6 shows the sensitivity of the Kallus [2018] methods and SoftBlock to hyperparameters (SoftBlock is very robust, while the Kallus [2018] PSOD method only performs comparably when hyperparameters are chosen optimally).

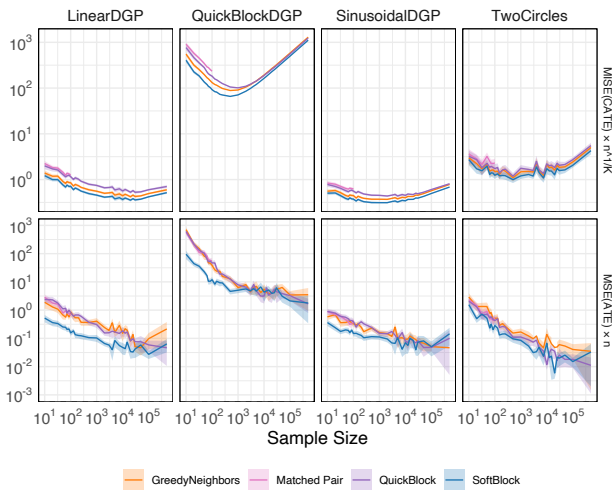


Figure 4: The design-based estimators from SoftBlock are appreciably better than existing methods. Mean squared-error and mean integrated-squared-error for the design-based estimators of the ATE and CATE, respectively. These values are multiplied by a function of sample size for ease of comparison.

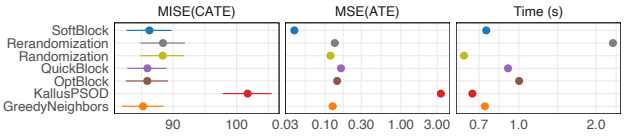


Figure 5: Mean squared-error of the ATE, mean integrated-squared-error of the CATE and time to calculate a design and perform estimation.

7 Conclusion

In this paper, we've provided a framework through which to think about designs for conditional average treatment effect estimation and provided a formulation of the problem as graph cutting. Through this framework we presented two novel experimental designs which are well-suited to estimating CATEs and compare them to prior work. Simulations demonstrate that this method provides an improvement in terms of both computational tractability as well as efficiency.

References

Jerome H Friedman and Lawrence C Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, 7(4):697–717, 1979.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

Nathan Kallus. Optimal a priori balance in the design of controlled experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):85–112, 2018.

Michael J Higgins, Fredrik Sävje, and Jasjeet S Sekhon. Improving massive experiments with threshold blocking. *Proceedings of the National Academy of Sciences*, 113(27):7369–7376, 2016.

Donald B Rubin. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 2011.

Ronald A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.

Kari Lock Morgan, Donald B Rubin, et al. Rerandomization to improve covariate balance in experiments. *The Annals of Statistics*, 40(2):1263–1282, 2012.

Peter Hall, Joel L Horowitz, and Bing-Yi Jing. On blocking rules for the bootstrap with dependent data. *Biometrika*, 82(3):561–574, 1995.

Robert Greevy, Bo Lu, Jeffrey H Silber, and Paul Rosenbaum. Optimal multivariate matching before randomization. *Biostatistics*, 5(2):263–275, 2004.

Xinran Li, Peng Ding, and Donald B Rubin. Asymptotic theory of rerandomization in treatment–control experiments. *Proceedings of the National Academy of Sciences*, 115(37):9157–9162, 2018.

Christopher Harshaw, Fredrik Sävje, Daniel Spielman, and Peng Zhang. Balancing covariates in randomized experiments using the gram-schmidt walk, 2020.

Ben B Hansen and Jake Bowers. Covariate balance in simple, stratified and clustered comparative studies. *Statistical Science*, pages 219–236, 2008.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.

Claudia Shi, David Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. In *Advances in Neural Information Processing Systems*, pages 2507–2517, 2019.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10): 4156–4165, 2019.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.

Oren Anava and Kfir Levy. k*-nearest neighbors: From global to local. In *Advances in neural information processing systems*, pages 4916–4924, 2016.

Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM Journal on Computing*, 37(1):319–357, 2007.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6): 1115–1145, 1995.

Luca Trevisan. Max cut and the smallest eigenvalue. *SIAM Journal on Computing*, 41(6):1769–1786, 2012.

Kosuke Imai. Variance identification and efficiency analysis in randomized experiments under the matched-pair design. *Statistics in medicine*, 27(24): 4857–4873, 2008.

Jack Edmonds. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4): 233–240, 1967.

Elizabeth A. Stuart. Matching methods for causal inference: A review and a look forward. *Statist. Sci.*, 25(1):1–21, 02 2010. doi: 10.1214/09-STS313. URL https://doi.org/10.1214/09-STS313.

Ben B Hansen and Stephanie Olsen Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627, 2006.

Harsh Parikh, Cynthia Rudin, and Alexander Volfovsky. Malts: Matching after learning to stretch. *arXiv preprint arXiv:1811.07415*, 2018.

Russell Lyons and Yuval Peres. *Probability on trees and networks*, volume 42. Cambridge University Press, 2017.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Norbert Henze, Mathew D Penrose, et al. On the multivariate runs test. *The Annals of Statistics*, 27(1): 290–298, 1999.

Winston Lin. Agnostic notes on regression adjustments to experimental data: Reexamining freedman's critique. *The Annals of Applied Statistics*, 7(1):295–318, 2013.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Nathan Kallus. Balanced policy evaluation and learning. *arXiv preprint arXiv:1705.07384*, 2017.

A. Gretton, K. Fukumizu, C.H. Teo, L. Song, B. Schölkopf, and A.J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, pages 585–592, 2008.

Le Song. *Learning via Hilbert space embedding of distributions*. 2008.