Variable Importance Matching for Causal Inference

Quinn Lanners[1] **Harsh Parikh**[2] **Alexander Volfovsky**[3] **Cynthia Rudin**[2] **David Page**[1]

[1]Dept. of Biostatistics, Duke University, Durham, NC, USA.
[2]Dept. of Computer Science, Duke University, Durham, NC, USA.
[3]Dept. of Statistical Science, Duke University, Durham, NC, USA.

Abstract

Our goal is to produce methods for observational causal inference that are auditable, easy to troubleshoot, accurate for treatment effect estimation, and scalable to high-dimensional data. We describe a general framework called Model-to-Match that achieves these goals by (i) learning a distance metric via outcome modeling, (ii) creating matched groups using the distance metric, and (iii) using the matched groups to estimate treatment effects. Model-to-Match uses variable importance measurements to construct a distance metric, making it a flexible framework that can be adapted to various applications. Concentrating on the scalability of the problem in the number of potential confounders, we operationalize the Model-to-Match framework with LASSO. We derive performance guarantees for settings where LASSO outcome modeling consistently identifies all confounders (importantly *without* requiring the linear model to be correctly specified). We also provide experimental results demonstrating the method's auditability, accuracy, and scalability as well as extensions to more general nonparametric outcome modeling.

1 INTRODUCTION

Matching methods are a popular approach to causal inference on observational data due to their conceptual simplicity. These methods aim to emulate randomized controlled trials by pairing similar treated and control units, thus allowing for treatment effect estimation [Stuart, 2010]. One significant benefit of using matching methods is their *auditability* along with their accuracy. An auditable method allows domain experts to validate the estimation procedure, argue about the violation of key assumptions, and determine whether the analysis is trustworthy. Since causal analyses often depend on untestable assumptions, it is critical to determine whether all important confounders are accounted for, if data are processed correctly, and whether the treatment and control units in the matched groups are cohesive enough to be comparable [Parikh et al., 2022b]. Parikh et al. [2022a] and Yu et al. [2021] showed that the audit of matched groups using external unstructured data is crucial in healthcare and social science scenarios. In high-stakes scenarios, audibility enables domain experts to make data-driven and *trustworthy* decisions and policies.

We would ideally be able to match units that are exactly identical to one another except for treatment assignments [Rosenbaum and Rubin, 1983]. However, exact matches are almost impossible in high-dimensional settings with continuous covariates [Parikh et al., 2022c]. In such scenarios, we aim to create *almost-exact matches* on important covariates. The question then becomes how to construct a distance metric between units that determines who should be in a unit's matched group. We want to learn a distance metric that provides *accurate* causal estimates, ensures *auditability* so we can evaluate and troubleshoot, and is *scalable* to large observational datasets that might be used for high-stakes policy decisions.

We introduce the *Model-to-Match* framework which uses variable importance from prognostic score models to learn a distance metric. The framework has three steps. First, we use machine learning to estimate outcomes and use the measured variable importance to construct a distance metric. Second, we use the learned distance metric to match treatment and control units into matched groups. Third, we use the matched groups to estimate conditional average treatment effects (CATEs). Our research focuses on the first step in this framework, as learning a good distance metric is an essential but difficult step to ensure that matching yields accurate treatment estimates.

A special case of our framework is called *LCM – LASSO Coefficient Matching*. LCM has the characteristics we desire. It is able to accurately estimate treatment effects, creates

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

almost-exact matched groups, and scales better than other comparable methods by orders of magnitude. LCM uses LASSO (Least Absolute Shrinkage and Selection Operator) coefficients to identify important variables and then uses K-nearest neighbors to construct matched groups. LCM benefits from both the efficiency of parametric models and the power of nonlinear modeling by leveraging a parametric method to learn which features to match on and then a nonparametric approach for treatment effect estimation. It is simple to implement yet works extremely well. We perform extensive empirical studies to compare LCM's performance with existing methods. Our results demonstrate that LCM can accurately and efficiently recover true treatment effects even in high-dimensional and non-linear setups without compromising auditability (Section 6). We further propose adaptations of our framework such as (a) metalearner LCM, (b) feature importance matching using decision trees, and (c) LCM-augmented-prognostic scores that work well in complex scenarios (Section 7).

2 BACKGROUND AND ASSUMPTIONS

We study the setting where every individual i in the population $\mathcal{S}$ is assigned to one of the two treatments $T_i \in \{0, 1\}$. Under the stable unit treatment value assumption (SUTVA), we define the potential outcomes of individual i as $Y_i(0)$ and $Y_i(1)$. We consider an i.i.d sample of n individuals, $\mathcal{S}_n$, where for each individual i we observe a p-dimensional pre-treatment covariate vector $\mathbf{X}_i$, an assigned treatment T_i, and an observed outcome $Y_i = Y_i(1)T_i + Y_i(0)(1 - T_i)$.

The individualized treatment effect is defined as $\tau_i := Y_i(1) - Y_i(0)$. Since we observe only one of the potential outcomes for each unit, τ_i is not observed for any of the units. We need to impute the missing potential outcomes to estimate the treatment effects of interest [Rubin, 2011]. In our setup, we are interested in identifying (a) conditional average treatment effects (CATEs) $\tau(\mathbf{x}) := \mathbb{E}[\tau_i \mid \mathbf{X}_i = \mathbf{x}]$ for all $\mathbf{x} \in Dom(\mathbf{X})$, and (b) the average treatment effect (ATE) $\tau := \mathbb{E}[\tau_i]$.

In observational data (where the treatments are not randomized), the treatment choice and potential outcomes can depend on common variables, which are referred to as confounders. In our setup, we assume that the set of confounders is a subset of the set of pre-treatment covariates, and potential outcomes and treatment assignment are conditionally independent given $\mathbf{X}$: $(Y_i(1), Y_i(0)) \perp T_i \mid \mathbf{X}_i$. This is referred to as conditional ignorability [Rubin, 1974]. Lastly, we assume that the probability of a unit receiving treatment t is bounded away from 1 and 0: $0 < P(T_i = t \mid \mathbf{X}_i = \mathbf{x}) < 1$. This is referred to as the positivity assumption. Combining the positivity and conditional ignorability assumptions, adjusting for pre-treatment covariates ($\mathbf{X}$) is sufficient to identify CATEs and ATE.

Matching Methods. Matching methods use a distance-metric, $d_{\mathcal{M}}$, on $\mathbf{X}$ to group similar units with different treatment assignments in order to estimate the causal effects of treatment T on outcome Y. The most popular matching techniques are *propensity score matching* (PSM) [Rosenbaum and Rubin, 1983] and *prognostic score matching* (PGM) [Hansen, 2008b]. These techniques project the data to a lower dimensional propensity or prognostic score, which are then used for matching. These projections can be sensitive to modeling choices that affect the accuracy of the treatment effect estimates [Kreif et al., 2016]. Further, the units within a matched group can be far from each other in covariate space – i.e., the matched groups are generally not auditable [Parikh et al., 2022c]. To date, the only observational causal inference techniques that attempt to optimize accuracy while maintaining auditability are those stemming from the almost-matching-exactly (AME) framework, namely the optimal matching (optMatch) [Yu et al., 2021, Kallus, 2017], genetic matching (GenMatch) [Diamond and Sekhon, 2013], FLAME/DAME [Wang et al., 2017, Dieng et al., 2019], MALTS [Parikh et al., 2022c, 2019, 2022a] and AHB [Morucci et al., 2020] algorithms. FLAME/DAME can scale to extremely large datasets but handles only categorical variables. GenMatch, MALTS, and AHB can also handle both continuous and categorical variables but do not scale as well, thereby limiting their usefulness (see Figure 3 in Section 6). What we develop is a method that yields accurate treatment effect estimates and is auditable like MALTS but can scale to much larger datasets and run at a fraction of the time.

Formally, for a unit i, the K-nearest neighbors of units with treatment t' and the corresponding matched group $\text{MG}_{d_{\mathcal{M}}}(\mathbf{X}_i)$ are defined as

$$\text{KNN}_{d_{\mathcal{M}}}(\mathbf{X}_i, t') := \left\{ k : \sum_{j \in \mathcal{S}_n^{(t')}} \mathbb{1}\left[\begin{array}{l} d_{\mathcal{M}}(\mathbf{X}_i, \mathbf{X}_j) \\ < d_{\mathcal{M}}(\mathbf{X}_i, \mathbf{X}_k) \end{array} \right] < K \right\},$$

$$\text{MG}_{d_{\mathcal{M}}}(\mathbf{X}_i) := \bigcup_{t' \in \{0,1\}} \text{KNN}_{d_{\mathcal{M}}}(\mathbf{X}_i, t'),$$

where $\mathcal{S}_n^{(t')} := \{j : T_j = t'\}$ represents the set of units whose treatment assignment is t'. Match groups can then be used to estimate potential outcomes, $\widehat{Y}_i(t') = \psi\left(\text{KNN}_{d_{\mathcal{M}}}(\mathbf{X}_i, t')\right)$, where ψ is a function of the outcomes of the K-nearest neighbors (e.g. arithmetic mean). As we will see, a high quality distance metric is key to creating accurate estimates. A good distance metric can lead to interpretable matched groups and accurate treatment effect estimates; a poor distance metric leads to neither.

Non-matching Methods. There are a number of non-matching frameworks that can estimate conditional average treatment effects. Regression methods, particularly doubly

robust regression methods, are often used to estimate CATEs [Farrell, 2015]. However, their performance is highly sensitive to model misspecification, requiring either the propensity or outcome model to be correctly specified. Machine learning methods are also popular for estimating CATEs. The most commonly used machine learning methods include Bayesian additive regression trees (BART) [Hahn et al., 2019], double machine learning [Chernozhukov et al., 2017], and generalized random forests [Athey et al., 2019]. While these methods can accurately estimate CATEs, they are often significantly less interpretable than matching methods and are not auditable. Additionally, previous almost-matching-exactly literature has shown that AME methods achieve similar CATE estimation accuracy to machine learning approaches while maintaining auditability Parikh et al. [2022c], Morucci et al. [2020], Wang et al. [2017]. For these reasons, in this paper we focus on comparing LCM to other matching methods and AME methods in particular. We include an experiment comparing LCM to machine learning methods on a high-dimensional quadratic dataset in the Supplementary Material.

3 MODEL-TO-MATCH FRAMEWORK

We propose a framework, called *Model-to-Match*, that focuses on combining prognostic score modeling with distance metric learning for almost-exact matching. Our framework is divided into three steps: (i) learning the weight matrix $\mathcal{M}$ of a distance metric $d_{\mathcal{M}}$ using a machine learning model, (ii) creating matched groups using the learned $d_{\mathcal{M}}$, and (iii) estimating treatment effects using the matched groups.

In our framework, we restrict ourselves to binary and continuous pre-treatment covariates. As such, all categorical covariates are dummified in the data preprocessing steps. We let p indicate the dimensionality of the final covariate space after preprocessing. This facilitates the use of more feature-importance methods (such as LASSO) and allows the feature space to be more finely weighted.

We choose our distance metric, $d_{\mathcal{M}}$, such that for any $\mathbf{X}_1$ and $\mathbf{X}_2$, $d_{\mathcal{M}}(\mathbf{X}_1, \mathbf{X}_2) = \|\mathcal{M}\mathbf{X}_1 - \mathcal{M}\mathbf{X}_2\|_m$. $\mathcal{M}$ is a $p \times p$ matrix and the m in $\|\cdot\|_m$ is flexible and can be any positive integer.

To learn a distance metric in our Model-to-Match framework we first train two machine learning models $f^{(0)}$ and $f^{(1)}$, such that for any $i \in \mathcal{S}_n$, $\widehat{Y}_i(t') = f^{(t')}(\mathbf{X}_i)$. For each $j \in \{1, 2, \ldots, p\}$ we then calculate θ_j, the importance of covariate $X_{\cdot,j}$ to $f^{(0)}$ and $f^{(1)}$.

Variable Importance Example 1: If the f's are linear estimators, such as LASSO or Ridge where $f^{(t)}_{\boldsymbol{\beta}^{(t)}} = \mathbf{X}\boldsymbol{\beta}^{(t)}$, then θ_j can be $\sum_{t' \in \{0,1\}} \frac{|\beta_j^{(t')}|}{\|\boldsymbol{\beta}^{(t')}\|_1}$.

Variable Importance Example 2: If the f's are decision trees then θ_j can be measured via Gini importance, feature permutation importance, or a similar feature importance metric.

Variable Importance Example 3: For f's from backward elimination with ordinary least squares, θ_j can be equal to the drop in R^2 when the j-th feature $X_{\cdot,j}$ is removed.

Variable Importance Example 4: For any generic model class, θ_j can be measured via subtractive model reliance, which measures the change in the loss of a model when a covariate is perturbed [Fisher et al., 2019].

We then set all the diagonal entries, $\mathcal{M}_{j,j}$, in the distance metric $\mathcal{M}$ to be equal to $|\theta_j|$ and all the non-diagonal entries in $\mathcal{M}$ to zero. By constructing $\mathcal{M}$ in this way we can interpret each weight, $\mathcal{M}_{j,j}$, as the relative feature importance of covariate $X_{\cdot,j}$.

We are interested in having an $\mathcal{M}$ that is sparse so that we only match on the important covariates. Further, we want the estimation of f's to be scalable in both the number of samples and the number of covariates. Keeping these requirements in mind, we use ℓ_1-regularized regression, i.e., LASSO, as the modeling method of choice for the majority of this paper. However, our framework is general and can be applied to any supervised model class. For example, we discuss using shallow regression trees to model the f's in Section 7. In practice, LASSO performs well for this step of the framework.

4 LINEAR COEFFICIENT MATCHING

In this section, we operationalize the *Model-to-Match* framework using LASSO [Tibshirani, 1996] as the machine learning algorithm for learning the distance metric and refer to this as LASSO Coefficient Matching (LCM). As in the example in Section 3, we use scaled absolute values of LASSO's coefficients as the diagonal entries for an $\mathcal{M}^*$. Since LASSO's coefficients are sparse, the entries of $\mathcal{M}^*$ will be sparse. This creates a distance metric $d_{\mathcal{M}^*}$ that prioritizes tighter matches on a small number of important covariates, leading to faster runtimes and facilitating matched groups that are close in important covariates.

We perform *honest* causal estimation for a given observed dataset $\mathcal{S}_n$. Broadly, honest causal estimation means that we do not use the same data to learn about the control variables as we do for inference [Ratkovic, 2019]. We achieve honesty by dividing the data into two disjoint subsets: $\mathcal{S}_{n,tr}$ and $\mathcal{S}_{n,est}$. In Step (i), we use $\mathcal{S}_{n,tr}$ to estimate $\boldsymbol{\beta}$'s and, by consequence, learn $d_{\mathcal{M}^*}$. Algorithm 1 describes our training step to learn $\mathcal{M}^*$ using LASSO. In Step (ii), we then perform matching with replacement using $d_{\mathcal{M}^*}$ to get matched groups, $\text{MG}_{d_{\mathcal{M}^*}}(\mathbf{X}_i)$, for each unit $i \in \mathcal{S}_{n,est}$. In Step (iii), we use $\text{MG}_{d_{\mathcal{M}^*}}(\mathbf{X}_i)$ to estimate the CATE for $\mathbf{X} = \mathbf{X}_i$ as

$\widehat{\tau}(\mathbf{X}_i) = \widehat{Y}_i(1) - \widehat{Y}_i(0)$ where

$$\widehat{Y}_i(t') = \frac{\sum_{k \in \text{MG}_{d_{\mathcal{M}^*}}(\mathbf{X}_i)} \mathbb{1}[T_k = t'] Y_k}{\sum_{k \in \text{MG}_{d_{\mathcal{M}^*}}(\mathbf{X}_i)} \mathbb{1}[T_k = t']}.$$

Data: Dataset $\mathcal{S}_{n,tr}$
Result: Distance metric $\mathcal{M}^*$
begin
  $W \leftarrow [0, ..., 0] \in \mathbb{R}^p$
  (Loop over treatment possibilities.)
  for t' *in* $\{0, 1\}$ **do**
    (Find units that have treatment t'.)
    $\mathcal{S}_{n,tr}^{(t')} \leftarrow \{i \in \mathcal{S}_{n,tr} : T_i = t'\}$
    (Run LASSO to get coefficients.)
    $\hat{\boldsymbol{\beta}}^{(t')} \leftarrow$
     $\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \lambda \|\boldsymbol{\beta}\|_1 + \sum_{i \in \mathcal{S}_{n,tr}^{(t')}} (Y_i - \mathbf{X}_i \boldsymbol{\beta})^2$
    (Average the element wise absolute values of the coefficients across treatment and control.)
    for l *in* $\{1, ..., p\}$ **do**
      $W_l \leftarrow W_l + \frac{|\hat{\beta}_l^{(t')}|}{\|\hat{\boldsymbol{\beta}}^{(t')}\|_1}$
    end
  end
  (Coefficients used as stretches in distance metric.)
  $\mathcal{M}^* \leftarrow \mathbf{0}_{p \times p}$
  for l *in* $\{1, ..., p\}$ **do**
    $\mathcal{M}^*_{l,l} \leftarrow \frac{1}{2} W_l$
  end
end

Algorithm 1: Algorithm to estimate $\mathcal{M}^*$ using LASSO

Since we perform honest causal inference where we do not use the same data to learn $d_{\mathcal{M}^*}$ as we do for estimating CATEs, our method performs η-fold cross-fitting by swapping the training set each time. This is similar to the strategy used in Chernozhukov et al. [2018] and enables the estimation of CATEs for all $i \in \mathcal{S}_n$. Because LASSO does not need many observations to fit the data well, we use only one of the η splits as the training set and the data in the remaining $(1 - \eta)$ splits as the estimation set. Using a smaller amount of data in the learning step allows us to create match groups with a larger portion of the data. Because the nearest neighbor-based estimation in Step (iii) is local and non-parametric, more data will improve the quality of matched groups and the accuracy of the CATEs.

For matching we employ the Manhattan distance to align with the additive linear form and $\|\cdot\|_1$ regularization of LASSO. In particular, for all $i, j \in \mathcal{S}_{n,est}$, $d_{\mathcal{M}^*}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{l=1}^{p} \mathcal{M}^*_{l,l} |\mathbf{X}_{i,l} - \mathbf{X}_{j,l}|$. Our method has three hyperparameters: η, λ, and K. We learn λ using cross-validation in our training in Step (i). The number of nearest neighbors, K, and the number of splits for cross-fitting, η, can be chosen through cross-validation or set manually.

5 THEORETICAL RESULTS

Here, we prove optimality properties of using LASSO to learn our distance metric. We then show under what conditions LCM guarantees consistency in CATE estimation. Proofs are included in the Supplementary Materials.

Theorem 5.1 motivates LCM. It shows that if the potential outcomes are linear in the predictors then using the absolute values of the coefficients in these models as the stretches in a distance metric guarantees that as the distance between two units decreases, their expected outcomes become closer.

Theorem 5.1. *[Closeness in* $\mathbf{X}$ *implies closeness in* Y*]. Consider a* p*-dimensional covariate space where for* $t' \in \{0, 1\}$, $f^{(t')}(\mathbf{X}_i) = \mathbb{E}[Y_i | \mathbf{X} = \mathbf{X}_i, T = t'] = \mathbf{X}_i \boldsymbol{\beta}^{(t')}$. *Construct* $\mathcal{M} \in \mathbb{R}^{p \times p}$ *where for all* $l, r \in \{1, ..., p\}$ $\mathcal{M}_{l,l} = |\beta_l^{(t')}|$ *and for* $l \neq r$ $\mathcal{M}_{l,r} = 0$*. Then,* $\forall i, j$*, we have that* $d_{\mathcal{M}}(\mathbf{X}_i, \mathbf{X}_j) \geq \left| f^{(t')}(\mathbf{X}_i) - f^{(t')}(\mathbf{X}_j) \right|$.

From here, we define a diagonal Mahalanobis distance matrix as any $\widetilde{\mathcal{M}} \in \mathbb{R}^{p \times p}$ that is diagonal (for all $l, r \in \{1, ..., p\}$, $l \neq r$, $\widetilde{\mathcal{M}}_{l,r} = 0$) and has non-negative entries ($\widetilde{\mathcal{M}}_{l,l} \geq 0$). We show in Theorem 5.2 that the $\mathcal{M}$ from Theorem 5.1 is the optimal stretch matrix, compared to any other equally scaled diagonal Mahlanobis distance matrix, in regards to the maximum absolute difference in expected outcomes.

Theorem 5.2. *[Optimality of* $\mathcal{M}$*] Using the setup of Theorem 5.1, let* $\text{supp}(\mathbf{X}) = \mathbb{R}^p$*. Consider an arbitrary diagonal Mahalanobis distance matrix* $\widetilde{\mathcal{M}} \in \mathbb{R}^{p \times p}$ *where* $\sum_{l=1}^{p} |\widetilde{\mathcal{M}}_{l,l}| = \sum_{l=1}^{p} |\boldsymbol{\beta}_l^{(t')}|$ *and* $\widetilde{\mathcal{M}}_{l,l} > 0$ *when* $|\beta_l^{(t')}| > 0$*. For some* $\epsilon \geq 0$ *and* $\mathbf{X}_1 \in \mathbb{R}^p$*, define* $S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1) := \{\mathbf{X}_2 : \mathbf{X}_2 \in \mathbb{R}^p, d_{\widetilde{\mathcal{M}}}(\mathbf{X}_1, \mathbf{X}_2) = \epsilon\}$*. Then,*

$$\sup_{\mathbf{X}_2 \in S_{\mathcal{M},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_2)| \leq \sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_3)|.$$

These results show how a linear outcome model induces a meaningful distance metric for causal inference. The following theorem states that when we do not know the true value of the coefficients (and more generally when the model is non-linear but LASSO still recovers its support), we can employ the LCM procedure of Section 4 to generate a distance metric that yields consistent estimates of CATEs. This theorem uses the notion of variable importance, as discussed in Section 3.

Theorem 5.3. *[Consistency of LCM] For $t' \in \{0, 1\}$, let $f^{(t')}(\mathbf{X}_i) = \mathbb{E}[Y_i|\mathbf{X} = \mathbf{X}_i, T = t']$. Let $f^{(t')}$ be Lipschitz continuous and,*

$$\text{supp}\left(f^{(t')}\right) := \left\{j : \textit{importance of } \mathbf{X}_{\cdot,j} \textit{ in } f^{(t')} \textit{ is } > 0\right\}.$$

Denote $d_{\mathcal{M}^*}$ as the distance metric learned by LCM in Section 4 and let $\Gamma(\mathcal{M}^*) = \{j : \mathcal{M}^*_{j,j} > 0\}$. LCM is consistent for CATE estimation if $\text{supp}\left(f^{(0)}\right) \bigcup \text{supp}\left(f^{(1)}\right) \subseteq \Gamma(\mathcal{M}^*)$.

This result follows from LASSO and its adaptations' ability to estimate sparse coefficient vectors in high dimensions, even when $n < p$ [Meinshausen and Yu, 2009, Zhou, 2010, Wasserman and Roeder, 2009, Meinshausen and Bühlmann, 2006]. LASSO also exhibits consistency for feature selection in some nonlinear settings [Zhang et al., 2016].

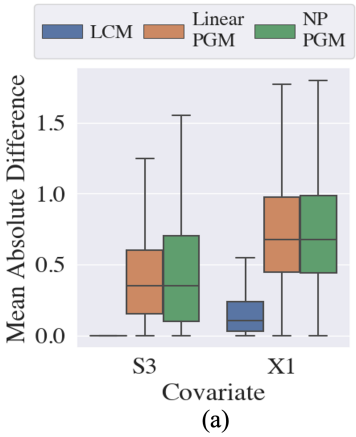
6 EXPERIMENTAL RESULTS

Our experiments focus on factors crucial in high-stakes causal inference. **(i) Accuracy and Auditability**: We compare LCM's matched groups to PGM's and highlight the importance of auditability. **(ii) Nonlinear Outcomes**: We study if LCM is sensitive to model misspecification and compare our results to linear PGM (which uses the same underlying prognostic model as LCM). **(iii) Scalability**: We compare LCM to existing AME algorithms in both runtime and estimation performance as both the number of observations and the number of features increase.

6.1 ACCURACY AND AUDITABILITY

Matching enables us to investigate whether a CATE is estimated in a trustworthy manner by *auditing* the quality of the matched groups. We now highlight how LCM produces accurate estimates while matching tightly on important covariates. We work with the ACIC 2018 Atlantic Causal Inference Conference semi-synthetic dataset [Carvalho et al., 2019], which is based on data from the National Study of Learning Mindsets randomized trial [Yeager, 2015-2016]. The dataset contains 10,000 students across 76 schools. There are four categorical student-level covariates and one categorical and five continuous school-level covariates. Carvalho et al. [2019] constructed this semi-synthetic dataset by drawing covariates from the real experiment and then synthetically generating treatment assignments and outcomes. Details can be found in Carvalho et al. [2019].

We ran our method alongside linear PGM, computed using LASSO, and nonparametric PGM, computed using gradient boosted trees. All three methods recover ATE estimates that are close to the true value of 0.24 – **LCM**: 0.249, **Linear PGM**: 0.251, and **Nonparametric PGM**: 0.260, which are also in line with the estimates of other interpretable and uninterpretable methods described in Carvalho et al. [2019].



(a)

	T	S3	X1
Query	1	5	-.05
LCM	0	5	-.05
	0	5	-.05
	0	5	-.05
	0	5	-.05
Linear PGM	0	5	-.05
	0	5	-.05
	0	**6**	**1.26**
	0	**6**	**1.26**
NP PGM	0	5	-.05
	0	5	-.05
	0	**6**	**.98**
	0	**6**	**.98**

(b)

Figure 1: Closeness in important covariates for matched groups produced by LCM, linear PGM, and nonparametric (NP) PGM. *(a)* shows the mean absolute difference between a query unit and its matched group's covariate values. Smaller values imply better and tighter matches. *(b)* shows, for a random sample, the four nearest neighbors of opposite treatment under LCM, linear PGM, and NP PGM. In *(b)*, the text in red indicates values that are far from the query unit's value. S3 indicates the self-reported prior achievements of students and is important for selection into treatment, and X1 indicates school-level average mindset score of the students and is an effect modifier.

While all three methods accurately estimate the ATE, *only LCM matches almost exactly on important covariates*. We compare how tightly LCM, linear PGM, and nonparametric PGM fit on a covariate that is identified as important for selection into treatment (S3) and one that is an effect modifier (X1). Figure 1 shows that LCM matches tighter on important covariates than PGM. In this way, LCM more closely emulates exact-matching and results in more intuitive and auditable match groups. The fact that LCM accurately estimates the treatment effect and matches so tightly on these important covariates increases the trust we have in our conclusions. We expand on these findings and show that LCM matches tighter across all the effect modifiers in the Supplementary Material.

6.2 NONLINEAR OUTCOMES

We have shown that linear prognostic score matches are not tight on important covariates, leading to unintuitive matched groups. However, LASSO estimated prognostic scores are more interpretable than scores estimated with gradient boosted trees. This interpretability comes at a cost: the performance of linear PGM heavily depends on the linearity of the underlying data generation process. LCM is more

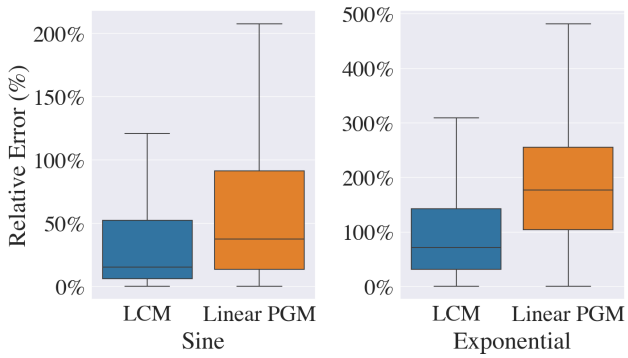


Figure 2: CATE estimation accuracy of LCM and Linear PGM on nonlinear synthetically generated datasets **Sine** and **Exponential**. The y-axis is the absolute CATE estimation error relative to the true ATE.

robust to nonlinear data because its LASSO component is used *only to determine the relative weight of features* in the distance metric (not to model the outcome with a linear combination of the covariates).

We compare CATE estimation accuracy of LCM and linear PGM on two synthetically generated datasets where the outcome is a non-linear function of the covariates. We call these datasets **Sine** and **Exponential** to align with their underlying potential outcome functions. We generate 5000 samples and 100 covariates for each dataset. For **Sine**, the outcome function is

$$Y_i = \sin(X_{i,1}) - T_i \sin(X_{i,2}).$$

Whereas, for **Exponential**, the outcome function is

$$Y_i = 2e^{X_{i,1}} - \sum_{j=2}^{3} e^{X_{i,j}} + T_i e^{X_{i,4}}.$$

We outline the specific details of the data generation processes in the Supplementary Material. Figure 2 shows that LCM is more robust to nonlinear outcome functions than linear PGM.

Again, the superior performance of LCM is unsurprising because it performs nonlinear estimation in Step (iii), using the linear LASSO method in Step (i) only to pinpoint important covariates upon which nonlinear estimation can be successfully performed.

6.3 SCALABILITY

Existing almost-matching-exactly methods learn covariate weights and/or create match groups through computationally expensive and data hungry optimization algorithms. In this section we compare LCM to MALTS [Parikh et al., 2022c], GenMatch [Diamond and Sekhon, 2013], and AHB [Morucci et al., 2020] in regards to scalability in runtime and CATE estimation accuracy. We omit FLAME/DAME [Wang et al., 2017, Dieng et al., 2019] from this comparison since it can only handle discrete covariates.

We generate synthetic datasets of various sizes from the quadratic data generation process described in Parikh et al. [2022c] and the Supplementary Materials. We first measure the runtime scalability of LCM, MALTS, GenMatch, and AHB with respect to the number of samples, n, and number of covariates, p. To measure scaling runtime in n, we keep the number of covariates constant at 64 and increase the number of samples from 256 to 8192. To measure scaling in p, we set the number of samples to be 2048 and vary the number of covariates from 8 to 1024. The Supplementary Materials contain further information on how runtimes were measured. Figure 3 shows the runtimes for each of the AME algorithms on these various dataset sizes, highlighting the multiple-order-of-magnitude runtime disparity between LCM and other AME methods. MALTS ran out of memory (16GB RAM) for the largest dataset in each plot. We stopped increasing the dataset sizes for AHB when its runtime surpassed the longest measured runtime of all other methods.

As discussed in Section 4, LASSO is capable of recovering sparse $\boldsymbol{\beta}$s and important features in high dimensional settings. Naturally, LCM also excels at producing accurate CATE estimates as the number of irrelevant covariates grows. Figure 4 shows how LCM is robust to added noise as the number of unimportant covariates grows – unlike MALTS and GenMatch, which struggle to learn an accurate distance metric as the dimensionality of the covariate space increases. Here, we keep the number of important covariates equal to 8.

7 MODEL-TO-MATCH ADAPTATIONS

In this section, we propose three adaptations of the Model-to-Match framework that extend LCM. The first approach uses a metalearner variant of LCM and shows improvement in CATE estimation in certain settings. The second adaptation proposes the use of a tree-based outcome modeling approach in place of LASSO. The third adaptation combines prognostic score matching with LCM to yield accurate CATEs and tight match groups.

7.1 METALEARNER LCM

Metalearners leverage powerful regression tools for estimating heterogenous treatment effects [Künzel et al., 2019]. LASSO Coefficient Matching can be adapted to run similar to the T-learner outlined in Künzel et al. [2019] by learning separate distance metrics for control and treated units. The metalearner adaptation of LCM is advantageous when certain covariates have vastly different effects on the outcome

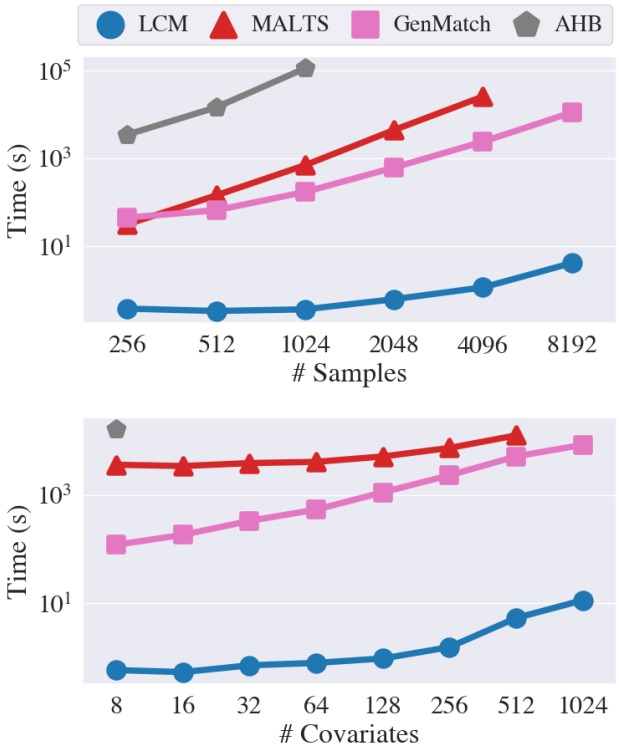


Figure 3: Scalability in n and p for LCM, MALTS, GenMatch, and AHB.

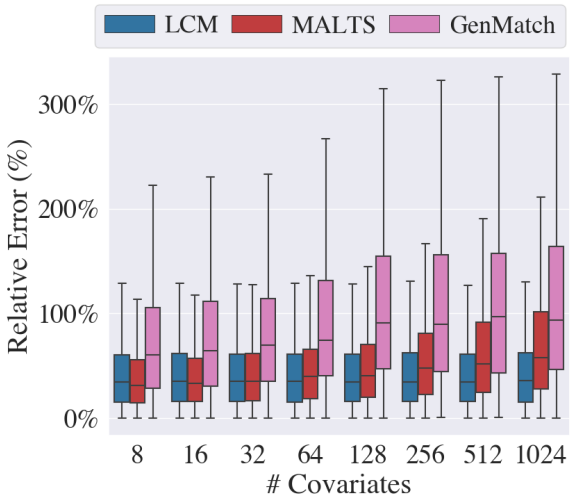


Figure 4: Absolute CATE estimation error relative to the true ATE for LCM, MALTS, and GenMatch as the number of covariates increases.

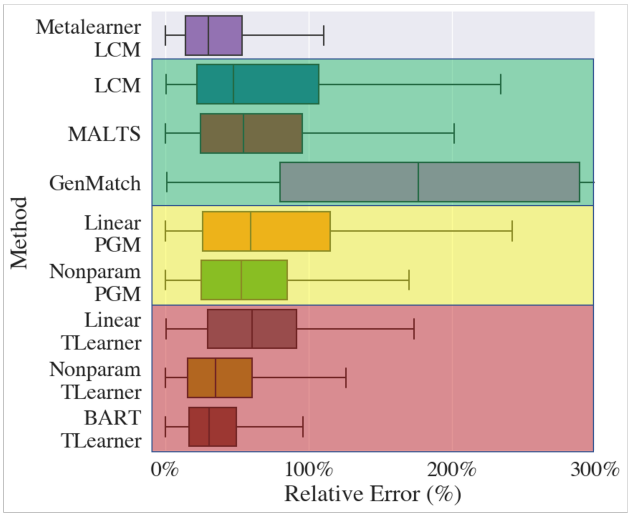


Figure 5: Absolute CATE estimation error relative to the true ATE for various methods on the **Sine** data generation process. The transparent boxes separate the methods into different categories. **Green**: Almost exact matching methods. **Yellow**: Other matching methods. **Red**: TLearner methods.

depending on if a sample received treatment or not.

For Metalearner LCM, we learn a separate distance metric, $d_{\mathcal{M}^{(t')*}}$, for each $t' \in \{0,1\}$. Specifically, for $l, r \in \{1,\ldots,p\}$ we set $\mathcal{M}_{l,l}^{(t')*} = |\hat{\beta}_l^{(t')}| \frac{1}{\|\hat{\boldsymbol{\beta}}^{(t')}\|_1}$, where $\hat{\boldsymbol{\beta}}^{(t')} = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \lambda\|\boldsymbol{\beta}\|_1 + \sum_{i\in\mathcal{S}_{n,tr}^{(t')}} (Y_i - \mathbf{X}_i\boldsymbol{\beta})^2$, and $\mathcal{M}_{l,r}^{(t')*} = 0$ when $l \neq r$. In Step (ii), for each unit $i \in \mathcal{S}_{n,est}$, we find K-nearest neighbors with replacement using the corresponding distance metric in each treatment arm.

To illustrate the advantage of the Metalearner LCM, we consider the same **Sine** data generation process used in Section 6.2. In **Sine**, covariate $X_{i,1}$ is important to the outcome under both treatment regimes ($Y_i(0)$ and $Y_i(1)$) while covariate $X_{i,2}$ is only relevant to the outcome under treatment ($Y_i(1)$). We generate 500 samples and 10 covariates. We compare LCM to the previously used matching methods along with linear and nonparametric T-Learners. Figure 5 shows estimated CATE errors for each method. Metalearner LCM improves upon LCM, which already outperforms other matching methods, and is comparable to T-Learners.

Figure 6 shows how Metalearner LCM stretches the control and treatment response surfaces differently, whereas regular LCM learns a global metric that is a linear combination of the two treatment spaces. The Metalearner variant is more suitable for problems in which accurate CATE estimation is more important than emulating a randomized experiment.

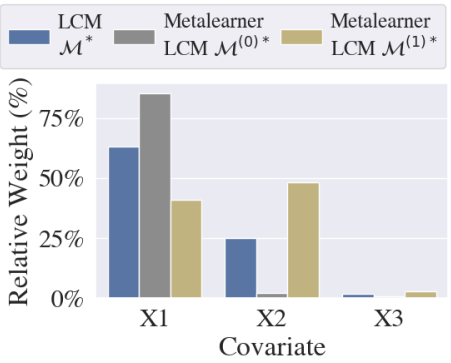


Figure 6: Relative covariate weights averaged over the η-folds for LCM $\mathcal{M}^*$, Metalearner LCM $\mathcal{M}^{(0)*}$, and Metalearner LCM $\mathcal{M}^{(0)*}$. This shows that the Metalearner LCM's distance metrics are different between treatment and control groups.

7.2 FEATURE IMPORTANCE MATCHING

LASSO can often find the important features, even if the true data generation process is nonlinear [Zhang et al., 2016]. However, in cases where it cannot, we can use any nonlinear method (decision tree, random forest, BART, AdaBoost, etc.) from which we can extract a measure of feature importance. These feature importance values can be used in place of LASSO coefficients in Algorithm 1 as weights for matching.

We demonstrate this using shallow decisions trees as the model and Gini importance as the feature importance measure [Menze et al., 2009]. We use a shallow decision tree to promote sparsity and to account for nonlinearities in the outcome space. We generate a dataset with 1000 samples and 10 covariates where only the first covariate is important: $Y_i(0) = X_{i,1}^2 + \epsilon_{i,y}$ and $Y_i(1) = X_{i,1}^2 + 10 + \epsilon_{i,y}$. A linear approach will not find this important covariate because it is symmetric around 0. The full data generation process is outlined in the Supplementary Material. Figure 7 shows that in this setting, the tree-based method creates more accurate CATE estimates than the LASSO method (LCM).

7.3 LCM-AUGMENTED-PGM

As shown in Section 6.1, LCM produces tighter matched groups on important covariates than linear and non-parametric PGM. However, PGM sometimes can estimate CATEs more accurately while not producing tight matched groups. This might occur either when the parametric prognostic model is correctly specified or when there is a strong non-linear effect that a non-parametric prognostic score can model accurately. In such situations, we propose augmenting PGM with LCM to guarantee tight matches and accurate

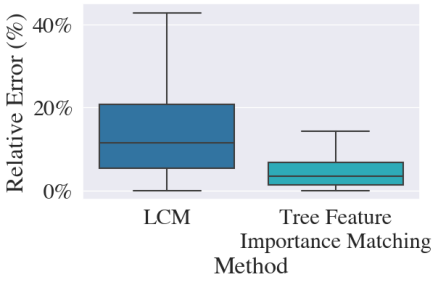


Figure 7: Absolute CATE estimation error relative to the true ATE for LCM vs. the Model-to-Match framework with classification and regression decision tree (CART) as the model and Gini feature importance as the feature importance measure.

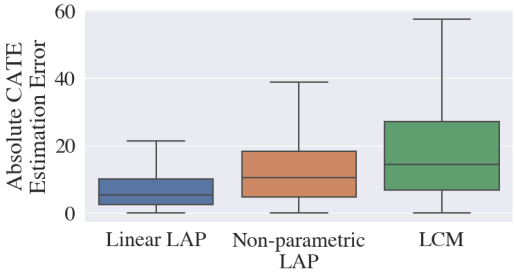


Figure 8: Absolute CATE estimation error for linear LAP (blue), non-parametric LAP (orange) and LCM (green).

CATE estimates. Our LCM-augmented-PGM (LAP) is a two stage procedure. In the first stage, we match using prognostic scores and create large matched groups. In the second stage, we match using the distance metric learned via LCM inside each PGM matched group. The first stage leverages the flexibility of outcome modeling and the second stage ensures tight matching on important covariates.

We compare LCM and LAP using the quadratic data generation process used in Section 6.3 and described in the Supplementary Material. We generate 5000 units and 20 covariates, of which the first 5 are important and the other 15 are irrelevant. Here, we first do 25 nearest neighbors matching with PGM and then perform 5 nearest neighbors matching using the LCM learned distance metric. Figure 8 shows that for this problem setup, both linear LAP and non-parametric LAP are more accurate than LCM. Further, Figure 9 shows that the matches created using non-parametric LAP are almost equally as tight as LCM's matches on the 5 important covariates and do not prioritize matching on irrelevant covariates.

8 DISCUSSION AND CONCLUSION

Model-to-Match is a fast, scalable, and auditable framework for observational causal inference. Unlike other almost-

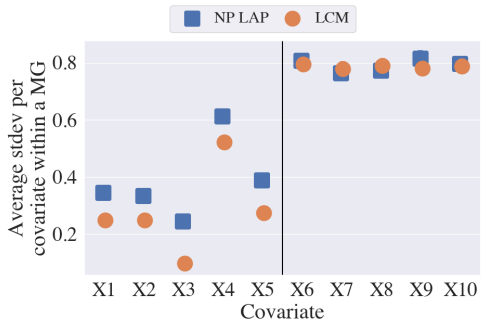


Figure 9: Average standard deviation for each covariate inside the matched groups for non-parametric LAP (NP LAP) and LCM. The smaller the standard deviation, the tighter the match on that covariate. The dataset has 20 covariates, but we only show 10 for ease of presentation. Note that X1-X5 are important and X6-X10 are unimportant.

matching-exactly approaches, Model-to-Match can scale to large datasets and high-dimensional settings and is flexible in regards to how the outcome space is modeled to learn a distance metric. We implemented Model-to-Match using LASSO as our machine learning algorithm of choice and refer to this as LCM. We show many desirable properties of LCM – including robustness to model misspecification and the ability to handle high-dimensional settings – and provide details on its consistency for CATE estimation. We provide additional experimental results in the Supplementary Material including further comparisons to non-matching CATE estimation methods and a simulation showing the advantage of LCM over equally weighted matching after feature selection.

Limitations and Future Directions. Model-to-Match is for i.i.d. data and should be extended to situations with either network interference or time-series effects. Furthermore, Model-to-Match is sensitive to the variable importance metric choice – leading to confounding bias if the correct support is not recovered. While we introduce our framework for categorical treatments, we are working on extending its application to continuous treatment regimes.

Other variations of Model-to-Match are easily possible. While we show sparse decision trees as a potential substitute to LASSO, any machine learning algorithm can be used. Furthermore, one can use other configurations in the matching and estimation steps of the framework, such as using a $\|\cdot\|_2$ norm instead of $\|\cdot\|_1$, employing a caliper matching method instead of K nearest neighbors, or choosing a different post-matching estimator instead of arithmetic average for potential outcomes. This level of flexibility makes Model-to-Match a framework that can be adapted to a variety of practical problems. In future work, we plan to both study the theory behind different Model-to-Match variations and implement our framework on large, real-world datasets such as electronic health records, genome studies, living standards measurement studies, etc.

Acknowledgements

We acknowledge funding from the National Science Foundation and Amazon under grant NSF IIS-2147061, and the National Institute on Drug Abuse under grant DA054994. Quinn Lanners thanks the National Science Foundation Artificial Intelligence for Designing and Understanding Materials - National Research Traineeship (aiM-NRT) at Duke University funded under grant DGE-2022040. Cynthia Rudin, Alexander Volfovsky and Harsh Parikh are supported by NSF grant DMS-2046880. Harsh Parikh is also partially support by Amazon Graduate Fellowship and NSF award IIS-1703431. Alexander Volfovsky is also supported by a National Science Foundation Faculty Early Career Development Award (CAREER: Design and analysis of experiments for complex social processes).

References

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2):1148 – 1178, 2019.

Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/py-why/EconML, 2019. Version 0.14.0.

Carlos Carvalho, Avi Feller, Jared Murray, Spencer Woody, and David Yeager. Assessing treatment effect variation in observational studies: Results from a data challenge, 2019. URL `https://arxiv.org/abs/1907.07592`.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and causal parameters, 2017.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

Alexis Diamond and Jasjeet S. Sekhon. Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies. *The Review of Economics and Statistics*, 95(3): 932–945, 2013.

Awa Dieng, Yameng Liu, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Interpretable almost-exact

matching for causal inference. *Proceedings of Machine Learning Research (Proceedings of AISTATS)*, 89:2445, 2019.

Vincent Dorie, Hugh Chipman, Robert McCulloch, Armon Dadgar, R Core Team, Guido U Draheim, Maarten Bosmans, Christophe Tournayre, Michael Petch, Rafael de Lucena Valle, et al. Package 'dbarts'. 2019.

Max H. Farrell. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics*, 189(1):1–23, 2015. URL https://EconPapers.repec.org/RePEc:eee:econom:v:189:y:2015:i:1:p:1-23.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

P. Richard Hahn, Jared S. Murray, and Carlos Carvalho. Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects, 2019.

Ben B. Hansen. The prognostic analogue of the propensity score. *Biometrika*, 95(2):481–488, 2008a. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/20441477.

Ben B Hansen. The prognostic analogue of the propensity score. *Biometrika*, 95(2):481–488, 2008b.

Daniel E Ho, Kosuke Imai, Gary King, and Elizabeth A Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political analysis*, 15(3):199–236, 2007.

Nathan Kallus. A Framework for Optimal Matching for Causal Inference. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 372–381, Fort Lauderdale, FL, USA, 20–22 Apr 2017.

Noémi Kreif, Susan Gruber, Rosalba Radice, Richard Grieve, and Jasjeet S Sekhon. Evaluating treatment effectiveness under model misspecification: A comparison of targeted maximum likelihood estimation with bias-corrected matching. *Statistical Methods in Medical Research*, 25(5):2315–2336, 2016. doi: 10.1177/0962280214521341.

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, February 2019. doi: 10.1073/pnas.1804597116. URL https://doi.org/10.1073/pnas.1804597116.

Almost Matching Exactly Lab. AME-ahb-r-package. https://github.com/almost-matching-exactly/AHB-R-package, 2022.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3), June 2006. doi: 10.1214/009053606000000281. URL https://doi.org/10.1214/009053606000000281.

Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1), February 2009. doi: 10.1214/07-aos582. URL https://doi.org/10.1214/07-aos582.

Bjoern H Menze, B Michael Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich, and Fred A Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10:1–16, 2009.

Marco Morucci, Vittorio Orlandi, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Adaptive hyper-box matching for interpretable individualized treatment effect estimation. In *Conference on Uncertainty in Artificial Intelligence*, pages 1089–1098. PMLR, 2020.

Harsh Parikh. AME-pymalts. https://github.com/almost-matching-exactly/MALTS, 2020.

Harsh Parikh, Cynthia Rudin, and Alexander Volfovsky. An application of matching after learning to stretch (malts). *Observational Studies*, 5(2):118–130, 2019.

Harsh Parikh, Kentaro Hoffman, Haoqi Sun, Sahar F. Zafar, Wendong Ge, Jin Jing, Lin Liu, Jimeng Sun, Aaron F. Struck, Alexander Volfovksy, Cynthia Rudin, and M. Brandon Westover. Effects of epileptiform activity on discharge outcome in critically ill patients: A retrospective cross-sectional study. 2022a.

Harsh Parikh, Carlos Varjao, Louise Xu, and Eric Tchetgen Tchetgen. Validating causal inference methods. In *International Conference on Machine Learning*, pages 17346–17358. PMLR, 2022b.

Harsh Parikh, Alexander Volfovsky, and Cynthia Rudin. Malts: Matching after learning to stretch. *Journal of Machine Learning Research*, 23(240), 2022c.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Marc T. Ratkovic. Rehabilitating the regression : Honest and valid causal inference through machine learning. 2019.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

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

Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25(1):1, 2010.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Tianyu Wang, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. FLAME: A fast large-scale almost matching exactly approach to causal inference. *arXiv preprint arXiv:1707.06315*, 2017.

Larry Wasserman and Kathryn Roeder. High-dimensional variable selection. *The Annals of Statistics*, 37(5A), October 2009. doi: 10.1214/08-aos646. URL https://doi.org/10.1214/08-aos646.

David S. Yeager. The national study of learning mindsets [united states]. 2015-2016. URL https://doi.org/10.3886/ICPSR37353.v4.

Ruoqi Yu, Dylan S. Small, David Harding, José Aveldanes, and Paul R. Rosenbaum. Optimal matching for observational studies that integrate quantitative and qualitative research. *Statistics and Public Policy*, 8(1):42–52, 2021. doi: 10.1080/2330443X.2021.1919260. URL https://doi.org/10.1080/2330443X.2021.1919260.

Yue Zhang, Soumya Ray, and Weihong Guo. On the consistency of feature selection with lasso for non-linear targets. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48, page 183–191, 2016.

Shuheng Zhou. Thresholded lasso for high dimensional variable selection and statistical estimation, 2010. URL https://arxiv.org/abs/1002.1583.

Variable Importance Matching for Causal Inference (Supplementary material)

Quinn Lanners[1] **Harsh Parikh**[2] **Alexander Volfovsky**[3] **Cynthia Rudin**[2] **David Page**[1]

[1]Dept. of Biostatistics, Duke University, Durham, NC, USA.
[2]Dept. of Computer Science, Duke University, Durham, NC, USA.
[3]Dept. of Statistical Science, Duke University, Durham, NC, USA.

9 PROOFS FOR THEOREMS IN SECTION 5

Theorem 5.1 (Closeness in $\mathbf{X}$ implies closeness in Y). Consider a p-dimensional covariate space where for $t' \in \{0,1\}$, $f^{(t')}(\mathbf{X}_i) = \mathbb{E}[Y_i|\mathbf{X} = \mathbf{X}_i, T = t'] = \mathbf{X}_i\boldsymbol{\beta}^{(t')}$. Construct $\mathcal{M} \in \mathbb{R}^{p\times p}$ where for all $l, r \in \{1, ..., p\}$ $\mathcal{M}_{l,l} = |\beta_l^{(t')}|$ and for $l \neq r$ $\mathcal{M}_{l,r} = 0$. Then, $\forall i, j$, we have that $d_{\mathcal{M}}(\mathbf{X}_i, \mathbf{X}_j) \geq \left|f^{(t')}(\mathbf{X}_i) - f^{(t')}(\mathbf{X}_j)\right|$.

Proof for Theorem 5.1.

$$d_{\mathcal{M}}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{l=1}^{p} \mathcal{M}_{l,l}|X_{i,l} - X_{j,l}| = \sum_{l=1}^{p} |\beta_l^{(t')}||X_{i,l} - X_{j,l}| \geq \left|\sum_{l=1}^{p} \beta_l^{(t')}(X_{i,l} - X_{j,l})\right|$$
$$= \left|f^{(t')}(\mathbf{X}_i) - f^{(t')}(\mathbf{X}_j)\right|.$$

QED

Theorem 5.2 (Optimality of $\mathcal{M}$). Using the setup of Theorem 5.1, let $\text{supp}(\mathbf{X}) = \mathbb{R}^p$. Consider an arbitrary diagonal Mahalanobis distance matrix $\widetilde{\mathcal{M}} \in \mathbb{R}^{p\times p}$ where $\|\widetilde{\mathcal{M}}\|_1 = \|\boldsymbol{\beta}^{(t')}\|_1$ and $\widetilde{\mathcal{M}}_{l,l} > 0$ when $|\beta_l^{(t')}| > 0$. For some $\epsilon \geq 0$ and $\mathbf{X}_1 \in \mathbb{R}^p$, define $S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1) := \{\mathbf{X}_2 : \mathbf{X}_2 \in \mathbb{R}^p, d_{\widetilde{\mathcal{M}}}(\mathbf{X}_1, \mathbf{X}_2) = \epsilon\}$. Then,

$$\sup_{\mathbf{X}_2 \in S_{\mathcal{M},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_2)| \leq \sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_3)|.$$

In what follows, we recall that a diagonal Mahalanobis distance matrix, $\widetilde{\mathcal{M}}$, is:

- diagonal: for all $l, r \in \{1, ..., p\}$, $l \neq r$, $\widetilde{\mathcal{M}}_{l,r} = 0$.
- non-negative entries: for all $l \in \{1, ..., p\}$, $\widetilde{\mathcal{M}}_{l,l} \geq 0$.

To prove this result, we first prove the following two lemmas.

Lemma 1 (Maximum Absolute Difference in Expected Outcomes under $\mathcal{M}$). Consider a p-dimensional covariate space where $\text{supp}(\mathbf{X}) = \mathbb{R}^p$ and for $t' \in \{0,1\}$, $f^{(t')}(\mathbf{X}_i) = \mathbb{E}[Y_i|\mathbf{X} = \mathbf{X}_i, T = t'] = \mathbf{X}_i\boldsymbol{\beta}^{(t')}$. Define $\mathcal{L} := \{l : \left|\beta_l^{(t')}\right| > 0\}$. Construct any diagonal Mahalanobis distance matrix, $\widetilde{\mathcal{M}}$, where $\|\widetilde{\mathcal{M}}\|_1 = \|\boldsymbol{\beta}^{(t')}\|_1$ and $\widetilde{\mathcal{M}}_{l,l} > 0$ when $|\beta_l^{(t')}| > 0$. Then, for some $\epsilon \geq 0$ and $\mathbf{X}_1 \in \mathbb{R}^p$, let $S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)$ be as defined in Theorem 5.2. We can conclude that

$$\sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_3)| = \epsilon \max_{l\in\mathcal{L}} \left\{\frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}}\right\}.$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Proof of Lemma 1.

$$\sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_3)| = \sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} \left| \sum_{l \in \mathcal{L}} \beta_l^{(t')} (X_{1,l} - X_{3,l}) \right|.$$

Note that since $\text{supp}(\mathbf{X}) = \mathbb{R}^p$, with probability strictly greater than zero there exists an $\mathbf{X}_1$ and $\mathbf{X}_3$ such that $d_{\widetilde{\mathcal{M}}}(\mathbf{X}_1, \mathbf{X}_3) = \epsilon$ and for all $l \in \mathcal{L}$, $X_{1,l} > X_{3,l}$ when $\beta_l^{(t')} > 0$ and $X_{1,l} < X_{3,l}$ when $\beta_l^{(t')} < 0$. Then,

$$\begin{aligned}
\sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} \left| \sum_{l \in \mathcal{L}} \beta_l^{(t')} (X_{1,l} - X_{3,l}) \right| &= \sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} \left\{ \sum_{l \in \mathcal{L}} \left| \beta_l^{(t')} (X_{1,l} - X_{3,l}) \right| \right\} \\
&= \sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} \left\{ \sum_{l \in \mathcal{L}} \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \widetilde{\mathcal{M}}_{l,l} \, |X_{1,l} - X_{3,l}| \right\}.
\end{aligned}$$

Note that $\left\{ \sum_{l \in \mathcal{L}} \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \widetilde{\mathcal{M}}_{l,l} \, |X_{1,l} - X_{3,l}| : \mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1) \right\}$ is maximized at $\epsilon \max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \right\}$. It is known that if the maximum value of a set is in the set, the supremum of that set equals the maximum value of that set. Therefore, we conclude that,

$$\sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} \left\{ \sum_{l \in \mathcal{L}} \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \widetilde{\mathcal{M}}_{l,l} \, |X_{1,l} - X_{3,l}| \right\} = \epsilon \max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \right\}.$$

QED

Lemma 2 Under the same setup as Lemma 1, $\max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \right\} \geq 1$.

Proof of Lemma 2. First note that $\sum_{l \in \mathcal{L}} \widetilde{\mathcal{M}}_{l,l} \leq \sum_{l=1}^{p} \widetilde{\mathcal{M}}_{l,l} = \sum_{l=1}^{p} |\beta_l^{(t')}| = \sum_{l \in \mathcal{L}} |\beta_l^{(t')}|$. There are two possible cases. In case one, $\forall l \in \mathcal{L}$, $\widetilde{\mathcal{M}}_{l,l} = \mathcal{M}_{l,l} = |\beta_l^{(t')}|$. Then $\max_{l \in \mathcal{L}} \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} = 1$. In case two, there exists $l \in \mathcal{L}$ for which $\widetilde{\mathcal{M}}_{l,l} \neq |\beta_l^{(t')}|$. But then there must exist an $l' \in \mathcal{L}$ for which $\widetilde{\mathcal{M}}_{l',l'} < |\beta_{l'}^{(t')}| \implies \max_{l \in \mathcal{L}} \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} > 1$. QED

Proof of Theorem 5.2. First note that $\mathcal{M}$ is a diagonal Mahalanobis distance matrix, $\|\mathcal{M}\|_1 = \|\boldsymbol{\beta}^{(t')}\|_1$, and $\mathcal{M}_{l,l} > 0$ when $|\beta_l^{(t')}| > 0$. The proof of the theorem then follows directly from Lemma 1 and Lemma 2.

$$\begin{aligned}
\sup_{\mathbf{X}_2 \in S_{\mathcal{M},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_2)| &= \epsilon \max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{\mathcal{M}_{l,l}} \right\} \\
&= \epsilon \max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{|\beta_l^{(t')}|} \right\} \\
&= \epsilon \\
&\leq \epsilon \max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \right\} \\
&= \sup_{\mathbf{X}_3 \in S_{\widetilde{\mathcal{M}},\epsilon}(\mathbf{X}_1)} |f^{(t')}(\mathbf{X}_1) - f^{(t')}(\mathbf{X}_3)|.
\end{aligned}$$

Where $\epsilon \leq \epsilon \max_{l \in \mathcal{L}} \left\{ \frac{|\beta_l^{(t')}|}{\widetilde{\mathcal{M}}_{l,l}} \right\}$ because of Lemma 2. QED

Theorem 5.3 (Consistency of LCM). For $t' \in \{0, 1\}$, let $f^{(t')}(\mathbf{X}_i) = \mathbb{E}[Y_i | \mathbf{X} = \mathbf{X}_i, T = t']$. Let $f^{(t')}$ be Lipschitz continuous and,

$$\text{supp}\left(f^{(t')}\right) := \left\{ j : \text{importance of } \mathbf{X}_{\cdot,j} \text{ in } f^{(t')} \text{ is } > 0 \right\}.$$

Denote $d_{\mathcal{M}^*}$ as the distance metric learned by LCM in Section 4 and let $\Gamma(\mathcal{M}^*) = \{j : \mathcal{M}^*_{j,j} > 0\}$. LCM is consistent for CATE estimation if $\text{supp}\left(f^{(0)}\right) \bigcup \text{supp}\left(f^{(1)}\right) \subseteq \Gamma(\mathcal{M}^*)$.

Proof of Theorem 5.3. First, let us introduce the concept of a smooth distance metric (defined in Parikh et al. [2022c]).

Definition 9.1 (Smooth Distance Metric)**.** $d : \mathbf{X} \times \mathbf{X} \to \mathbb{R}^+$ is a smooth distance metric if there exists a monotonically increasing bounded function $\delta_d(\cdot)$ with zero intercepts, such that $\forall i, j \in \mathcal{S}$ if $T_i = T_j = t'$ and $d(\mathbf{X}_i, \mathbf{X}_j) \leq a$ then $|\mathbb{E}[Y_i(t')|\mathbf{X}_i] - \mathbb{E}[Y_j(t')|\mathbf{X}_j]| \leq \delta_d(a)$.

Theorem 1 in [Parikh et al., 2022c] shows that matching with a smooth distance metric guarantees consistency of CATE estimates.

Recovering the correct support for the potential outcome functions implies that restricting to only variables in the recovered support, the potential outcomes are independent of the covariates: $(Y(1), Y(0)) \perp \mathbf{X} \mid \{\mathbf{X}_{\cdot,j}\}_{j \in \text{supp}(f^{(0)}) \cup \text{supp}(f^{(1)})}$. Also, note that if $\{\mathbf{X}_{i,j}\}_{j \in \text{supp}(f^{(0)}) \cup \text{supp}(f^{(1)})}$ is close to $\{\mathbf{X}_{k,j}\}_{j \in \text{supp}(f^{(0)}) \cup \text{supp}(f^{(1)})}$ then $f^{(0)}(X_i)$ is close to $f^{(0)}(X_k)$ and $f^{(1)}(X_i)$ is close to $f^{(1)}(X_k)$ by the definition of support and the Lipschitz continuity assumption. Thus, if $\text{supp}(f^{(0)}) \cup \text{supp}(f^{(1)}) \subseteq \Gamma(\mathcal{M}^*)$ then $d^*_{\mathcal{M}}$ is a smooth distance metric. This guarantees the consistency of our estimates. QED

Consistency of LASSO. Much work has been done on the consistency of LASSO for feature selection [Zhang et al., 2016]. The ability for LASSO to recover the correct support even in the case of non-linear targets makes it more robust to model misspecification. LASSO is consistent for support recovery if $f(\mathbf{X}_i, t) = \mathbb{E}[Y_i|\mathbf{X} = \mathbf{X}_i, T = t']$ satisfies one of the following conditions:

(i) $f(\mathbf{X}_i, t') = \mathbf{X}_i \boldsymbol{\beta}^{(\boldsymbol{t'})}$

(ii) $f(\mathbf{X}_i, t') = g\left(\mathbf{X}_i \boldsymbol{\beta}^{(\boldsymbol{t'})}\right)$ where $\beta_k^{(t')} \neq 0$ for $k \in \{1, .., r\}$, for some $r \leq p$, and, if $r < p$, $\beta_k^{(t)} = 0$ for $k \in \{r, ..., p\}$, and the following conditions are met:

   (a) $\mathbf{Cov}(\mathbf{X}, \mathbf{X})$ is invertible.

   (b) The eigenvalues of $\Sigma_{r,r} = \mathbf{Cov}(\mathbf{X}_{1:r}, \mathbf{X}_{1:r})$ are such that $0 < c_1 \leq \Lambda(\Sigma_{r,r}) \leq c_2 < \infty$. Where $\Lambda(\Sigma_{r,r})$ are the eigenvalues of $\Sigma_{r,r}$.

   (c) $E[Y(t')]^4 < \infty$

   (d) g is differentiable almost everywhere and for $t \sim \mathcal{N}(0, 1)$, $E(|g(t)|) < \infty$ and $E(|g'(t)|) < \infty$.

   (e) For all i, $E\left[X_i^T X_i \left|g\left(\mathbf{X}_i \boldsymbol{\beta}^{(\boldsymbol{t'})}\right)\right|^2\right] < \infty$.

10 METHOD IMPLEMENTATION FOR EXPERIMENTS

In this section we outline how we implemented each method used in our experiments. To calculate CATE estimates for all samples, we employed the same η-fold cross-fitting strategy for each method. In particular, we train models to estimate the $\widehat{Y}_i(t') = f^{(t')}(\mathbf{X}_i)$ for $t' \in \{0, 1\}$ using $S_{n,tr}$ and perform estimation on $S_{n,est}$. The only method that we did not use cross-fitting for was GenMatch, which does not use the outcome to learn it's distance metric and thus does not require a training set. All references to scikit-learn refer the Python machine learning package from Pedregosa et al. [2011].

- **LASSO Coefficient Matching**: We implemented the method described in this paper in Python. We use scikit-learn's `LassoCV` to learn $d_{\mathcal{M}^*}$ and `NearestNeighbors` with `metric='manhattan'` to perform nearest neighbor matching.
- **Linear and Nonparametric Prognostic Score Matching**: We follow the notion of a prognostic score outlined in Hansen [2008a]. In particular, we employ a *double* prognostic score matching method were we model both the control and treatment space separately as $\widehat{Y}_i(t') = f^{(t')}(\mathbf{X}_i)$ for $t' \in \{0, 1\}$. For linear PGM we use scikit-learn's `LassoCV` as our prognostic score models and for nonparametric PGM we use `GradientBoostingRegressor` for our prognostic score models. We then match with replacement on $[f^{(0)}(\mathbf{X}_i), f^{(1)}(\mathbf{X}_i]$ using scikit-learn's `NearestNeighbors` with `metric='euclidean'` to perform nearest neighbor matching. We estimated CATEs with the same mean estimator as LCM.
- **MALTS Matching**: We use the method developed in Parikh et al. [2022c] that was implemented in Python [Parikh, 2020]. We use the package's `mean` CATE estimator with `smooth_cate=False`.
- **MatchIt**: We use MatchIt's implementation of GenMatch [Ho et al., 2007]. We kept the default setting of `ratio=1`, which set $K = 1$ for matching. But we matched with replacement to be in line with LCM and the other matching methods we compared with.

- **Linear and Nonparametric TLearner**: We use the EconML TLearner implementation from Battocchi et al. [2019]. For Linear TLearner we use scikit-learn's `LassoCV` for our models and for Nonparametric TLearner we use scikit-learn's `GradientBoostingRegressor` for our models.
- **AHB**: We use the method developed in Morucci et al. [2020] that was implemented in R [Lab, 2022]. We use the package's `AHB_fast_match` implementation with the default settings.
- **Bart T-Learner**: We use the dbarts R package from Dorie et al. [2019]. We train a BART model on $S_{n,tr}$ to model $\widehat{Y}_i(t') = f^{(t')}(\mathbf{X}_i)$ for $t' \in \{0, 1\}$. We then estimate CATEs for each $j \in S_{n,est}$ as $f^{(1)}(\mathbf{X}_j) - f^{(0)}(\mathbf{X}_j)$.
- **Linear DoubleML**: We use the `econml.dml.DML` class in the econml Python package from Battocchi et al. [2019]. We fit a model on $S_{n,tr}$ setting `model_y=WeightedLassoCV`, `model_t=LogisticRegressionCV`, and `model_final=LassoCV`. We then estimate CATEs for each $j \in S_{n,est}$ using the `.effect()` method.
- **Causal Forest DoubleML**: We use the `econml.dml.CausalForestDML` class in the econml Python package from Battocchi et al. [2019]. We fit a model on $S_{n,tr}$ setting `model_y=WeightedLassoCV` and `model_t=LogisticRegressionCV`. We then estimate CATEs for each $j \in S_{n,est}$ using the `.effect()` method.
- **Causal Forest**: We use the implementation of causal forest from the grf R package from Battocchi et al. [2019]. We fit a model on $S_{n,tr}$ with the default package settings. We then used the fit model to estimate CATEs for each $j \in S_{n,est}$.

11 EXPERIMENTAL DETAILS FOR SECTION 6 AND SECTION 7

In this section, we describe the data generating processes used and provide further details regarding the setup of each experiment conducted in this paper. The source code necessary to reproduce all of the experiments in this paper is located in the GitHub repository: `https://github.com/almost-matching-exactly/variable_imp_matching`.

11.1 DATA GENERATION PROCESSES

Here we outline the data generation processes (DGPs) not fully outlined in the main text.

Sine and Exponential DGPs. *Used in Sections 6.2 and 7.1*. We generate the covariates and treatment assignments for the Sine and Exponential DGPs in a similar manner. For both, we generate data as follows:

$$\begin{aligned} X_{i,1}, \ldots, X_{i,p} &\overset{iid}{\sim} \text{Uniform}(-\alpha, \beta) \\ \epsilon_{i,y} &\overset{iid}{\sim} \mathcal{N}(0, \sigma^2), \epsilon_{i,t} \overset{iid}{\sim} \mathcal{N}(0, 1) \\ T_i &= \mathbb{1}\left[\text{expit}\Big(X_{i,1} + X_{i,2} + \epsilon_{i,t}\Big) > 0.5\right] \\ Y_i &= T_i Y_i(1) + (1 - T_i) Y_i(0) + \epsilon_{i,y}, \end{aligned}$$

where expit is the logistic sigmoid: $\text{expit}(x) = \frac{1}{1+e^{-x}}$.

For **Sine** we set $\alpha = \beta = \pi$, $\sigma^2 = 0.1$ and calculate the potential outcomes as

$$Y_i(0) = \sin(X_{i,1}),\ Y_i(1) = \sin(X_{i,1}) - \sin(X_{i,2}).$$

For **Exponential** we set $\alpha = \beta = 3$, $\sigma^2 = 1$ and calculate the potential outcomes as

$$Y_i(0) = 2e^{X_{i,1}} - \sum_{j=2}^{3} e^{X_{i,j}},\ Y_i(1) = 2e^{X_{i,1}} - \sum_{j=2}^{3} e^{X_{i,j}} + e^{X_{i,4}}.$$

Quadratic DGP. *Used in Sections 6.3 and 7.3*. This quadratic data generation process is also described in Parikh et al. [2022c]. This DGP includes both linear and quadratic terms. For each sample, let $\mathbf{X}_i$ be a p-dimensional vector where the first $k \leq p$ covariates are relevant and $\kappa \leq k$ is the number of covariates relevant to determining the treatment choice. The DGP is outlined below.

$$X_{i,p} \overset{iid}{\sim} \mathcal{N}(1, 1.5),\ \epsilon_{i,y}\epsilon_{i,t} \overset{iid}{\sim} \mathcal{N}(0,1),\ \ s_1, \ldots, s_{|k|} \overset{iid}{\sim} \text{Uniform}\{-1, 1\}$$
$$\alpha_j | s_j \overset{iid}{\sim} \mathcal{N}(10 s_j, 9),\ \beta_1, \ldots, \beta_{|k|} \overset{iid}{\sim} \mathcal{N}(1, 0.25)$$

$$Y_i(0) = \sum_{j \le k} \alpha_j X_{i,j}$$

$$Y_i(1) = \sum_{j \le k} \alpha_j X_{i,j} + \sum_{j \le k} \beta_j X_{i,j} + \sum_{j \le k} \sum_{j' \le k} X_{i,j} X_{i,j'}$$

$$T_i = \mathbb{1}\left[\text{expit}\Big(\sum_{j \le \kappa} X_{i,j} - \kappa + \epsilon_{i,t} \Big) > 0.5 \right]$$

$$Y_i = T_i Y_i(1) + (1 - T_i) Y_i(0) + \epsilon_{i,y}$$

Where $\text{expit}(x) = \frac{1}{1+e^{-x}}$.

Basic Quadratic DGP. *Used in Section 7.2*. This DGP is a quadratic DGP centered at zero. We generate each sample as shown.

$$X_{i,1}, \ldots, X_{i,10} \overset{iid}{\sim} \mathcal{N}(0, 2.5),\ \epsilon_{i,y} \overset{iid}{\sim} \mathcal{N}(0,1),\ T_i \sim \text{Bernoulli}(0.5)$$
$$Y_i(0) = X_{i,1}^2,\ Y_i(1) = X_{i,1}^2 + 10$$
$$Y_i = T_i Y_i(1) + (1 - T_i) Y_i(0) + \epsilon_{i,y}$$

11.2 EXPERIMENTAL DETAILS

In Table 1 we provide details on the experiments shown in this paper. We include additional notes for selected experiments below:

- Section 6.1: Accuracy and Auditability: We included the school id as a categorical covariate in our dataset. After preprocessing the categorical covariates, we had 6 continuous covariates and 98 binary covariates that we used as input to each model. We used only two splits due to the small occurrence rate of many of the categorical values. We repeated the cross-fitting process 50 times to smooth out treatment effect estimates for each method. All of the results in this section are for the combined 50 iterations.
- Section 6.3: Scalability: The matchit package only performs k:1 matching, so we kept K=1 for GenMatch (which is the default value). Reported runtimes were measured on a Slurm cluster with VMware, where each VM was an Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz. For measuring runtime, we ran each method 20 times on each dataset size. We report the average runtime for each method on each dataset. The variability across the 20 runs was negligible so we ommitted bars showing the standard deviation from the final plot. Each individual runtime measurement was ran on a separate Slurm job that was allocated a single core with 16GB RAM.
- Section 7.3: LCM-Augmented-PGM: For ease of implementation, we did not perform cross-fitting for this experiment. Rather, we just used half of the samples (2500) for training and the other half of the samples (2500) for estimation.

12 ADDITIONAL EXPERIMENTAL RESULTS

In this section, we include additional experimental results using LCM. We first discuss further findings from experiments in Section 6 and Section 7. We then show results of additional experiments comparing LCM to non-matching methods and matching methods with equal weights after feature selection.

Section 6.1: Accuracy and Auditability. Figure 10 in this document is an expanded plot of Figure 1(a) in the main text. The supplementary material's Figure 10 includes S3, X1, and all other effect modifiers X2, C1=1, C1=13, and C1=14. As mentioned in the caption of Figure 1(a) in the main text, S3 indicates the self-reported prior achievements of students and X1 indicates school-level average mindset score of the students. X2 is a school-level continuous covariate that measures the school's achievement level and C1 is a categorical covariate for race/ethnicity. We measure closeness in continuous covariates using the same mean absolute difference metric used in Figure 1(a) in the main text. Whereas, we measure

Table 1: Details of Experiments in Sections 6 and 7. The *Additional Information* column indicates if further details for that experiment are included in Section 11.2.

Section	Dataset	# Samples	# Covariates	K	η	Additional Notes
6.1: Accuracy and Auditability	ACIC 2018 Learning Mindset Dataset	10,000	10	10	2	Y
6.2: Nonlinear Outcome	Sine	5000	100	10	10	
	Exponential	5000	100	10	10	
6.3: Scalability	Linear + Quadratic	Varies	Varies	10 (1 for GenMatch - see notes)	2	Y
7.1: Metalearner LCM	Sine	500	10	10	5	
7.2: Feature Importance Matching	Simple Quadratic	500	10	10	5	
7.3: LCM-Augmented-PGM	Linear + Quadratic	5000	20	25 using PGM followed by 5 using LCM	N/A	Y

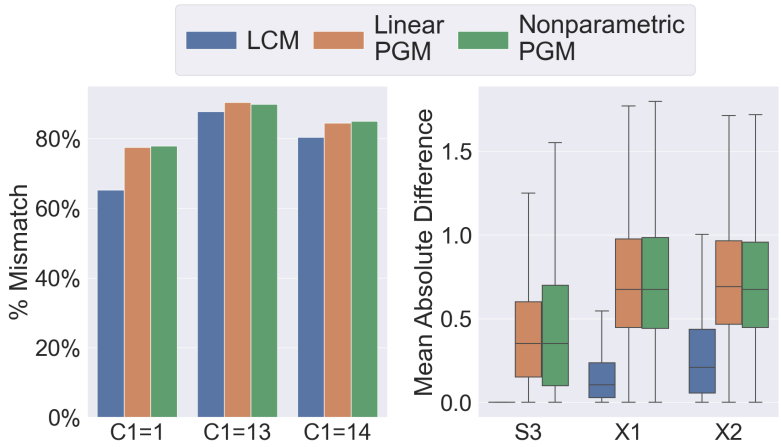


Figure 10: Closeness in important covariates for matched groups produced by LCM, linear PGM, and nonparametric (NP) PGM. Smaller values imply better and tighter matches.

closeness in categorical covariates as the percent of samples in a match group that do not have the same label as the query unit (% Mismatch). LCM matches much more tightly on all of the continuous covariates. For categorical covariates, while LCM matches tighter than PGM methods, it struggles compared to continuous covariates. We theorize this is due to the low occurrence rate of these features. In particular, C1=1 in 9.5%, C1=13 in 1.8% and C1=14 in 6.2% of samples. Therefore, it is difficult to find matches that have the same C1 value and are also similar in all of the other important covariates. LCM sometimes prioritizes matching almost-exactly on other covariates at the expense of these rare categorical covariates.

Carvalho et al. [2019] also states that although XC (Urbanicity) is not an effect modifier it is strongly related to X1 (student's fixed mindsets - summarized at the school level) and X2 (school achievement level) which are true effect modifiers. Because of this, seven of the eight methods that are summarized in Carvalho et al. [2019] identified XC as an effect modifier. Carvalho et al. [2019] further shows that, in this dataset, marginally the true cates for XC=3 are much lower than other values of XC. We show in Figure 11 that LCM also identifies this trend in XC.

For Section 6.1, we did not compare to other almost-matching-exactly methods (i.e. MALTS, AHB, GenMatch) due to the large size of the dataset. The ACIC 2018 Learning Mindset Dataset has 50,000 samples and >100 covariates after encoding the categorical features. Results from Section 6.3 highlight how intractable it would be to run other AME methods on a dataset of this size.

Section 6.2: Nonlinear Outcomes. Figure 12 shows CATE estimation accuracy for the same experiment in Section 6.2

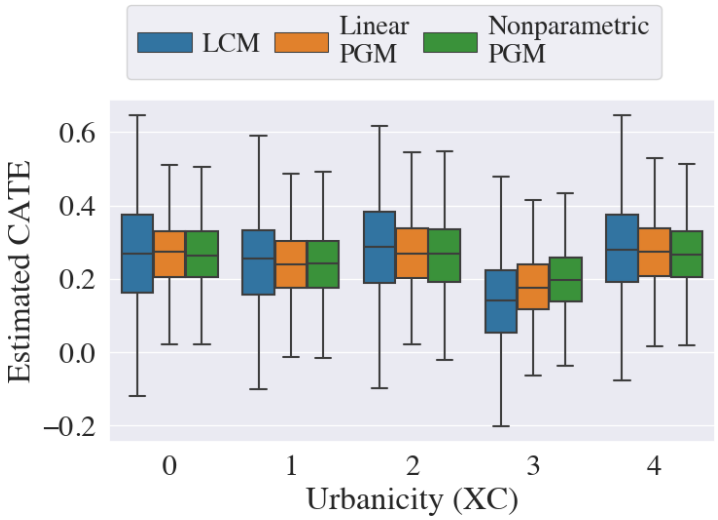


Figure 11: Marginal CATE estimates produced by LCM, Linear PGM, and Nonparametric PGM for the categorical school-level covariate of urbanicty (XC).

with the number of covariates increased to 500 for both the **Sine** and **Exponential** datasets. Given that we used 10 splits for this experiment, the training set in each fold had 500 samples. Note that LCM's accuracy does not suffer in this extremely high-dimensional setting where the number of samples equals the number of covariates. These results further highlight the ability of LCM to scale to very high-dimensional data even in the case of nonlinear outcome functions.

Section 7.1: Metalearner LCM. For the Metalearner LCM, here we show the effect of learning unique distance metrics for calculating control vs treated KNNs. We measure the distance between query unit's covariate values and the values of the ten nearest neighbors' of each treatment type. In particular, we calculate the mean absolute difference between a query unit's value and the values of its ten nearest neighbors. As explained in Section 7.1, X1 is a relevant covariate to the outcome under both treatment regimes, whereas X2 is only relevant to the outcome under treatment. X3 is unimportant in both setting and shown as a reference point. Figure 13 shows that while LCM's nearest neighbors are equally close on X0 and X1 in both treatment spaces, Metalearner LCM considers X2 as unimportant when calculating KNNs who are in the control group. This highlights how Metalearner LCM is able to adapt to outcome spaces that are different under different treatment regimes.

LCM vs Machine Learning Methods. Previous almost-matching-exactly literature has established that AME methods perform as well as (and often better than) machine learning methods like BART, causal forest, and double machine learning for estimating CATEs [Parikh et al., 2022c, Morucci et al., 2020, Wang et al., 2017]. For this reason, this paper focuses on comparing LCM to matching methods and particularly other AME methods. However, here we include an experiment comparing the CATE estimation accuracy of LCM to various machine learning methods on a high-dimensional non-linear dataset.

We use the Quadratic DGP with 25 relevant covariates, 2 of which are relevant to the treatment choice, and 125 irrelevant covariates. We generate 2500 samples and set $\eta = 5$. We run LCM with two configurations. *LCM Mean* is run with $K = 10$ and uses a mean estimator inside the match groups. *LCM Linear* is run with $K = 40$ and uses linear regression as the estimator inside the match groups. We compare to state-of-the-art machine learning methods double machine learning (DML), causal forest, and BART TLearner. Figure 14 shows that LCM Mean performs on par with the machine learning methods on this dataset, further highlighting the accuracy our method. LCM Linear improves upon LCM Mean, showing that we can achieve better accuracy with more sophisticated estimators if we are willing to increase the size of the match groups.

LCM vs Feature Selection. Here we show CATE estimation accuracy of LCM compared to matching equally on the covariates after feature selection. To compare with LCM, we estimate CATEs using feature selection by simply following the same steps as LCM but replacing the $\mathcal{M}^*$ with an $\mathcal{M} \in \mathbb{R}^{p \times p}$ such that $\mathcal{M}_{l,l} = 1$ when $\mathcal{M}^*_{l,l} > 0$ and $\mathcal{M}_{l,l} = 0$ when $\mathcal{M}^*_{l,l} = 0$. We refer to this method as *LASSO FS*. We also compare to an *Oracle* feature selector in which we assume that we know which covariates are important and match equally only on the important covariates.

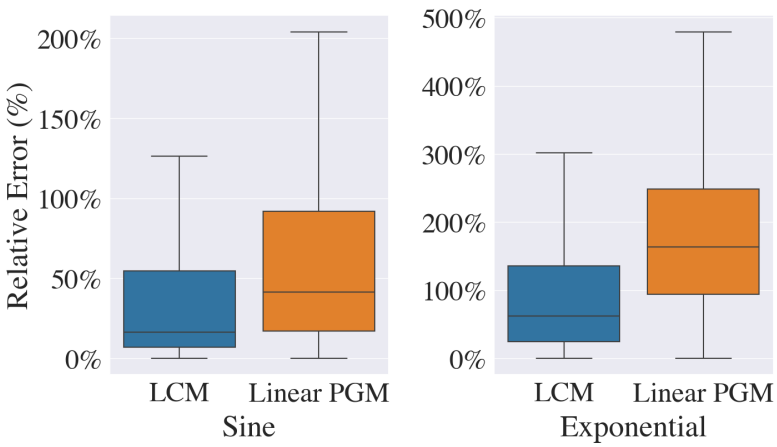


Figure 12: Comparing LCM's and Linear PGM's performances for high-dimensional nonlinear synthetically generated datasets **Sine** and **Exponential**.

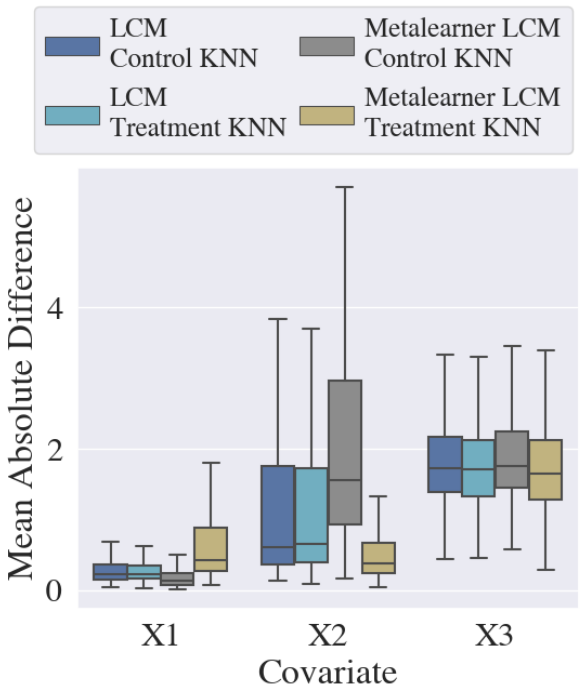


Figure 13: Measure of how tightly the KNN groups are for LCM versus Metalearner LCM under different treatment regimes.

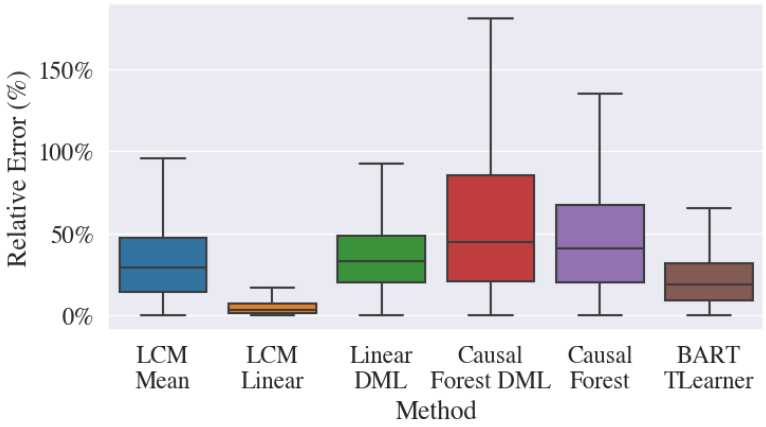


Figure 14: Estimated CATE absolute error relative to the true ATE for LCM Mean, LCM Linear, and state-of-the-art machine learning methods. DML stands for double machine learning.

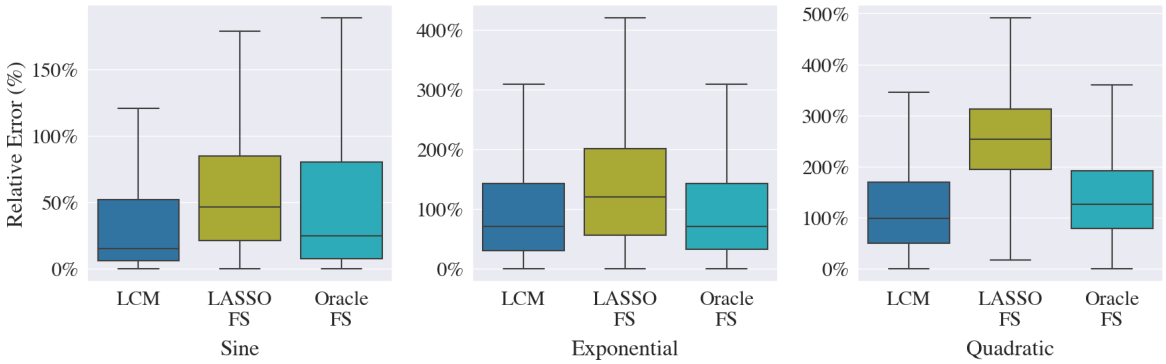


Figure 15: Estimated CATE absolute error relative to the true ATE for LCM and matching equally on covariates after LASSO and Oracle feature selection.

We run our analysis on three of the data generation processes used earlier in this paper. Namely, we run on the **Sine**, **Exponential**, and **Quadratic** DGPs described in Section 11.1. We generate 5000 samples and 100 covariates for each DGP and have two important covariates for **Sine**, four important covariates for **Exponential**, and five important covariates for **Quadratic**. All tests set $\eta = 5$ and $K = 10$. Figure 15 shows that LCM outperforms LASSO feature selection and performs on par with an Oracle feature selector. This highlights how using the relative weights of feature importance values in a distance metric, and thus matching tighter on covariates that more heavily contribute to the outcome, ultimately leads to more accurate CATE estimates.