Interpretable Almost-Exact Matching for Causal Inference

Awa Dieng, Yameng Liu, Sudeepa Roy, Cynthia Rudin, Alexander Volfovsky*
Duke University
Durham, NC 27708
{awa.dieng, alexander.volfovsky}@duke.edu, {ymliu, sudeepa, cynthia}@cs.duke.edu

Abstract

Matching methods are heavily used in the social and health sciences due to their interpretability. We aim to create the highest possible quality of treatment-control matches for categorical data in the potential outcomes framework. The method proposed in this work aims to match units on a weighted Hamming distance, taking into account the relative importance of the covariates; the algorithm aims to match units on as many relevant variables as possible. To do this, the algorithm creates a hierarchy of covariate combinations on which to match (similar to downward closure), in the process solving an optimization problem for each unit in order to construct the optimal matches. The algorithm uses a single dynamic program to solve all of the units' optimization problems simultaneously. Notable advantages of our method over existing matching procedures are its high-quality interpretable matches, versatility in handling different data distributions that may have irrelevant variables, and ability to handle missing data by matching on as many available covariates as possible.

1 INTRODUCTION

In observational causal inference where the scientist does not control the randomization of individuals into treatment, an ideal approach matches each treatment unit to a control unit with identical covariates. However, in high dimensions, few such "identical twins" exist, since it becomes unlikely that any two units have identical covariates in high dimensions. In that case, how might we construct a match assignment that would lead to accurate estimates of conditional average treatment effects (CATEs)?

For categorical variables, we might choose a Hamming distance to measure similarity between covariates. Then, the goal is to find control units that are similar to the treatment units on as many covariates as possible. However, the fact that not all covariates are equally important has serious implications for CATE estimation. Matching methods generally suffer in the presence of many irrelevant covariates (covariates that are not related to either treatment or outcome): the irrelevant variables would dominate the Hamming distance calculation, so that the treatment units would mainly be matched to the control units on the irrelevant variables. This means that matching methods do not always pass an important sanity check in that irrelevant variables should be irrelevant. To handle this issue with irrelevant covariates, in this work we choose to match units based on a *weighted* Hamming distance, where the weights can be learned from machine learning on a hold-out training set. These weights act like variable importance measures for defining the Hamming distance.

The choice to optimize matches using Hamming distance leads to a serious computational challenge: how does one compute optimal matches on Hamming distance? In this work, we define a matched group for a given unit as the solution to a constrained discrete optimization problem, which is to find the weighted Hamming distance of each treatment unit to the nearest control unit (and vice versa). There is one such optimization problem for each unit, and we solve all of these optimization problems efficiently with a single dynamic program. Our dynamic programming algorithm has the same basic monotonicity property (downwards closure) as that of the apriori algorithm (Agrawal and Srikant, 1994) used in data mining for finding frequent itemsets. However, frequency of itemsets is irrelevant

* Equal contribution from all authors.

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

here, instead the goal is to find a largest (weighted) set of covariates that both a treatment and control unit have in common. The algorithm, Dynamic Almost Matching Exactly – DAME – is efficient, owing to the use of bit-vector computations to match units in groups, and does not require an integer programming solver.

A more general version of our formulation (Full Almost Matching Exactly) adaptively chooses the features for matching in a data-driven way. Instead of using a fixed weighted Hamming distance, it uses the hold-out training set to determine how useful a *set* of variables is for prediction out of sample. For each treatment unit, it finds a set of variables that (i) allows a match to at least one control unit; (ii) together have the best out-of-sample prediction ability among all subsets of variables for which a match can be created (to at least one control unit). Again, even though for each unit we are searching for the best subset of variables, we can solve all of these optimization problems at once with our single dynamic program.

2 RELATED WORK

As mentioned earlier, exact matching is not possible in high dimensions, as "identical twins" in treatment and control samples are not likely to exist. Early on, this led to techniques that reduce dimension using propensity score matching (Rubin, 1973b,a, 1976; Cochran and Rubin, 1973), which extend to penalized regression approaches (Schneeweiss et al., 2009; Rassen and Schneeweiss, 2012; Belloni et al., 2014; Farrell, 2015). Propensity score matching methods project the entire dataset to one dimension and thus cannot be used for estimating CATE (conditional average treatment effect), since units within the matched groups often differ on important covariates. In "optimal matching," (Rosenbaum, 2016), an optimization problem is formed to choose matches according to a pre-defined distance measure, though as discussed above, this distance measure can be dominated by irrelevant covariates, leading to poor matched groups and biased estimates. Coarsened exact matching (Iacus et al., 2012, 2011) has the same problem, since again, the distance metric is pre-defined, rather than learned. Recent integer-programming-based methods considers extreme matches for all possible reasonable distance metrics, but this is computationally expensive and relies on manual effort to create the ranges (Morucci et al., 2018; Noor-E-Alam and Rudin, 2015); in contrast we use machine learning to create a single good match assignment.

In the framework of *almost-exact matching* (Wang et al., 2017), each matched group contains units that are close on covariates that are important for predicting outcomes. For example, Coarsened Exact Matching (Iacus et al., 2012, 2011) is almost-exact if one were to use an oracle (should one ever become available) that bins covariates according to importance for estimating causal effects. DAME's predecessor, the FLAME algorithm (Wang et al., 2017) is an almost-exact matching method that adapts the distance metric to the data using machine learning. It starts by matching "identical twins," and proceeds by eliminating less important covariates one by one, attempting to match individuals on the largest set of covariates that produce valid matched groups. FLAME can handle huge datasets, even datasets that are too large to fit in memory, and scales well with the number of covariates, but removing covariates in exactly one order (rather than all possible orders as in DAME) means that many high-quality matches will be missed.

DAME tends to match on more covariates than FLAME; the distances between matched units are smaller in DAME than in FLAME, thus its matches are distinctly higher quality. This has implications for missing data, where DAME can find matched groups that FLAME cannot.

3 ALMOST MATCHING EXACTLY (AME) FRAMEWORK

Consider a dataframe $D = [X, Y, T]$ where $X \in \{0, 1, \ldots, k\}^{n \times p}$, $Y \in \mathbb{R}^n$, $T \in \{0, 1\}^n$ respectively denote the categorical covariates for all units, the outcome vector and the treatment indicator (1 for treated, 0 for control). The j-th covariate X of unit i is denoted $x_{ij} \in \{0, 1, \ldots, k\}$. Notation $\mathbf{x}_i \in \{0, 1, \ldots, k\}^p$ indicates covariates for the ith unit, and $T_i \in \{0, 1\}$ is an indicator for whether or not unit i is treated.

Throughout we make SUTVA and ignorability assumptions (Rubin, 1980). The goal is to match treatment and control units on as many relevant covariates as possible. Relevance of covariate j is denoted by $w_j \geq 0$ and it is determined using a hold-out training set. w_j's can either be fixed beforehand or adjusted dynamically inside the algorithm (see Full-AME in Section 5).

For now, assuming that we have a fixed nonnegative weight w_j for each covariate j, we would like to find a match for each treatment unit t that *matches at least one control unit on as many relevant covariates as possible*. Thus we consider the following problem:

Almost Matching Exactly with Fixed Weights (AME): *For each treatment unit t,*

$$\boldsymbol{\theta}^{t\star} \in \operatorname{argmax}_{\boldsymbol{\theta} \in \{0,1\}^p} \boldsymbol{\theta}^T \mathbf{w} \text{ such that}$$
$$\exists\, \ell \text{ with } T_\ell = 0 \text{ and } \mathbf{x}_\ell \circ \boldsymbol{\theta} = \mathbf{x}_t \circ \boldsymbol{\theta},$$

where $\circ$ denotes Hadamard product. The solution to the AME problem is an indicator of the optimal set of

covariates for the matched group of treatment unit t. The constraint says that the optimal matched group contains at least one control unit. When the solution of the AME problem is the same for multiple treatment units, they form a single matched group. For treatment unit t, the **main matched group** for t contains all units ℓ so that $\mathbf{x}_t \circ \boldsymbol{\theta}^{t*} = \mathbf{x}_\ell \circ \boldsymbol{\theta}^{t*}$. If any unit ℓ (either control or treatment) within t's main matched group has its own different main matched group, then t's matched group is an **auxiliary matched group** for ℓ. In this case, ℓ could have been matched to other units on more covariates than it was matched to t. Estimation of CATE for a unit should always be done on the main matched group for that unit.

The formulation of the AME and main matched group is symmetric for control units. There are two straightforward (but inefficient) approaches to solving the AME problem for all units.

AME Solution 1 (quadratic in n, linear in p): Brute force pairwise comparison of treatment points to control points. (Detailed in the appendix.)

AME Solution 2 (order $n \log n$, exponential in p): Brute force iteration over all 2^p subsets of the p covariates. (Detailed in the appendix.)

If n is in the millions, the first solution, or any simple variation of it, is practically infeasible. A straightforward implementation of the second solution is also inefficient. However, a monotonicity property (downward closure) allows us to prune the search space so that the second solution can be modified to be completely practical. The `DAME` algorithm does not enumerate all $\boldsymbol{\theta}$'s, monotonicity reduces the number of $\boldsymbol{\theta}$'s it considers.

Proposition 3.1. (Monotonicity of $\boldsymbol{\theta}^*$ in AME solutions) *Fix treatment unit t. Consider feasible $\boldsymbol{\theta}$, meaning $\exists\ \ell$ with $T_\ell = 0$ and $\mathbf{x}_\ell \circ \boldsymbol{\theta} = \mathbf{x}_t \circ \boldsymbol{\theta}$. Then,*

- *Any feasible $\boldsymbol{\theta}'$ such that $\boldsymbol{\theta}' < \boldsymbol{\theta}$ elementwise will have $\boldsymbol{\theta}'^T \mathbf{w} \leq \boldsymbol{\theta}^T \mathbf{w}$.*
- *Consequently, consider feasible vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Define $\tilde{\boldsymbol{\theta}}$ as the elementwise $\min(\boldsymbol{\theta}, \boldsymbol{\theta}')$. Then $\tilde{\boldsymbol{\theta}}^T \mathbf{w} < \boldsymbol{\theta}^T \mathbf{w}$, and $\tilde{\boldsymbol{\theta}}^T \mathbf{w} < \boldsymbol{\theta}'^T \mathbf{w}$.*

These follow from the fact that the elements of $\boldsymbol{\theta}$ are binary and the elements of $\mathbf{w}$ are non-negative. The first property means that if we have found a feasible $\boldsymbol{\theta}$, we do not need to consider any $\boldsymbol{\theta}'$ with fewer 1's as a possible solution of the AME for unit t. Thus, the `DAME` algorithm starts from $\boldsymbol{\theta}$ being all 1's (consider all covariates). It systematically drops one element of $\boldsymbol{\theta}$ to zero at a time, then two, then three, ordered according to values of $\boldsymbol{\theta}^T \mathbf{w}$. The second property implies that we must evaluate both $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ as possible AME solutions before evaluating $\tilde{\boldsymbol{\theta}}$. Conversely, a new subset of variables defined by $\tilde{\boldsymbol{\theta}}$ cannot be considered unless all of its supersets have been considered. These two properties form the basis of the `DAME` algorithm.

The algorithm must be stopped early to avoid creating low quality matches. A useful stopping criterion is if the weighted sum of covariates $\boldsymbol{\theta}^T \mathbf{w}$ used for matching becomes too low (perhaps lower than a prespecified percentage of the total sum of weights $\|\mathbf{w}\|_1$).

Note that matching does not produce estimates, it produces a partition of the covariate space, based on which we can estimate CATEs. Within each main matched group, we use the difference of the average outcome of the treated units and the average outcome of the control units as an estimate of the CATE value, given the covariate values for that group. Smoothing the CATE estimates could be useful after matching.

4 DYNAMIC ALMOST MATCHING EXACTLY (DAME)

We call a *covariate-set* any set of covariates. We denote by $\mathcal{J}$ the original set of all covariates from the input dataset, where $p = |\mathcal{J}|$. When we *drop* a set of covariates s, it means we will match on $\mathcal{J} \setminus s$. For any covariate-set s, we associate an *indicator-vector* $\boldsymbol{\theta}_s \in \{0,1\}^p$ defined as follows:

$$\boldsymbol{\theta}_{s,j} = \mathbb{1}_{\{j \notin s\}} \quad \forall\ j \in \{1, .., p\} \tag{1}$$

that is, the value is 1 if the covariate is *not in* s implying that it is being used for matching.

Algorithm 1 gives the pseudocode of the `DAME` algorithm. It uses the monotonicity property stated in Proposition 3.1 and ideas from the *apriori algorithm for association rule mining* (Agrawal and Srikant, 1994). Instead of looping over all possible 2^p vectors to solve the AME, it considers a covariate-set s for being dropped only if satisfies the monotonicity property of Proposition 3.1. For example, if $\{1\}$ has been considered for being dropped to form matched groups, it would not process $\{1,2,3\}$ next because the monotonicity property requires $\{1,2\}$, $\{1,3\}$, and $\{2,3\}$ to have been considered previously for being dropped.

The `DAME` algorithm uses the `GroupedMR` (*Grouped Matching with Replacement*) subroutine given in Algorithm 2 to form all valid main matched groups having at least one treated and one control unit. `GroupedMR` takes a given subset of covariates and finds all subsets of treatment and control units that have identical values of those covariates. We use an efficient implementation of the `group-by` operation in the algorithm from Wang et al. (2017) that uses *bit-vectors*. To keep track of main matched groups, `GroupedMR` takes the entire set of units D as well as the set of unmatched units from

Algorithm 1: The `DAME` algorithm

Input : Data D, pre-computed weight vector w for all covariates (from machine learning)

Output : $\{D^m_{(h)}, \mathcal{MG}_{(h)}\}_{h\geq 1}$ all matched units and all the matched groups from all iterations h

Notation: h: iterations, $D_{(h)}$ (resp. $D^m_{(h)}$) = unmatched (resp. matched) units at the end of iteration h, $\mathcal{MG}_{(h)}$ = matched groups at the end of iteration h, $\Lambda_{(h)}$ = set of active covariate-sets at the end of iteration h that are eligible to be dropped to form matched groups, $\Delta_{(h)}$ = set of covariate-sets at the end of iteration h that have been processed (i.e., have been considered to be dropped and for formulation of matched groups).

Initialize: $D_{(0)} = D, D^m_{(0)} = \varnothing, \mathcal{MG}_{(0)} = \varnothing, \Lambda_{(0)} = \{\{1\}, ..., \{p\}\}, \Delta_{(0)} = \varnothing$, $h = 1$

while *there is at least one treatment unit to match in* $D_{(h-1)}$ **do**

- `(find the 'best' covariate-set to drop from the set of active covariate-sets)`
- Let $s^*_{(h)} \in \arg\max_{s\in\Lambda_{h-1}} \boldsymbol{\theta}_s^T \mathbf{w}$ ($\boldsymbol{\theta}_s \in \{0,1\}^p$ denotes the indicator-vector of s as in (1))
- **if** *early stopping condition is met* **then**
  - **Exit while loop**
- $(D^m_{(h)}, \mathcal{MG}_{(h)}) =$ `GroupedMR`$(D, D_{(h-1)}, \mathcal{J} \setminus s^*_{(h)})$ `(find matched units and main groups)`
- $Z_{(h)} =$ `GenerateNewActiveSets`$(\Delta_{(h-1)}, s^*_{(h)})$ `(generate new active covariate-sets)`
- $\Lambda_{(h)} = \Lambda_{(h-1)} \setminus \{s^*_{(h)}\}$ `(remove` $s^*_{(h)}$ `from the set of active sets)`
- $\Lambda_{(h)} = \Lambda_{(h)} \cup Z_{(h)}$ `(update the set of active sets)`
- $\Delta_{(h)} = \Delta_{(h-1)} \cup \{s^*_{(h)}\}$ `(update the set of already processed covariate-sets)`
- $D_{(h)} = D_{(h-1)} \setminus D^m_{(h-1)}$ `(remove matches)`
- $h = h + 1$

return $\{D^m_{(h)}, \mathcal{MG}_{(h)}\}_{h\geq 1}$

Algorithm 2: Procedure `GroupedMR`

Input : Data D, unmatched Data $D^{um} \subseteq D = (X, Y, T)$, subset of indexes of covariates $\mathcal{J}^s \subseteq \{1, ..., p\}$

Output : Newly matched units D^m using covariates indexed by $\mathcal{J}^s$ where groups have at least one treated and one control unit, and main matched groups for D^m

$M_{raw} =$ `group-by` $(D, \mathcal{J}^s)$ (`form groups on` D `by exact matching on` J^s)

$M =$ `prune`(M_{raw}) (`remove groups without at least one treatment and one control unit`)

$D^m =$ `Subset of` D^{um} `where the covariates match with some group in` M (`find newly matched units and their main matched groups`)

return $\{D^m, M\}$ (`newly matched units and main matched groups`)

the previous iteration $D_{(h-1)}$ as input along with the covariate-set $\mathcal{J} \setminus s^*_{(h)}$ to match on in this iteration. Instead of matching only the unmatched units in $D_{(h-1)}$ using the group-by procedure, it matches all units in D to allow for matching with replacement as in the AME objective. It keeps track of the main matched groups for the unmatched units $D_{(h-1)}$.

`DAME` keeps track of two sets of covariate-sets: (1) The set of **processed sets** Δ contains the covariate-sets whose main matched groups (if any exist) have already been formed. That is, Δ contains s if matches have been constructed on $\mathcal{J} \setminus s$ by calling the `GroupedMR` procedure. (2) The set of **active sets** Λ contains the covariate-sets s that are eligible to be dropped according to Proposition 3.1. For any iteration h, $\Lambda_{(h)} \cap \Delta_{(h)} = \varnothing$, i.e., the sets are disjoint, where $\Lambda_{(h)}, \Delta_{(h)}$ denote the states of Λ, Δ at the end of iteration h. Due to the monotonicity property stated in Proposition 3.1, if $s \in \Lambda_{(h)}$, then each proper subset $r \subset s$ belonged to $\Lambda_{(h')}$ in an earlier iteration $h' < h$. Once an active set $s \in \Lambda_{(h-1)}$ is chosen as the optimal subset to drop (i.e., s is $s^*_{(h)}$ in iteration h), s is excluded from $\Lambda_{(h)}$ (it is no longer active) and is included in $\Delta_{(h)}$ as a processed set. In that sense, the active sets are generated and included in $\Lambda_{(h)}$ in a hierarchical manner similar to the apriori algorithm. A set s is included in $\Lambda_{(h)}$ only if all of its proper subsets of one less size $r \subset s$, $|r| = |s| - 1$, have been processed.

The procedure `GenerateNewActiveSets` gives an efficient implementation of generation of new active sets in each iteration of `DAME`, and takes the currently processed sets $\Delta = \Delta_{(h-1)}$ and a newly processed set $s = s^*_{(h)}$ as input. Let $|s| = k$. In this procedure, $\Delta^k \subseteq \Delta \cup \{s\}$ denotes the set of all processed covariate-sets in Δ of size k, and also includes s. Inclusion of s in Δ^k may lead to generation of a new active set r of size $k+1$ only if all of r's subsets of size k (one less) have been previously processed. The new active sets triggered by inclusion of s in Δ^k would be supersets r of s of size $k+1$ if all subsets $s' \subset r$ of size $|s'| = k$ belong to Δ^k. To generate such candidate supersets r, we can append s with all covariates appearing in some covariate-set in Δ except those in s. However, this naive approach would iterate over many superfluous candidates for active sets. Instead, `GenerateNewActiveSets` safely prunes some such candidates that cannot be valid active sets using **support** of each covariate e in Δ^k, which is the number of sets in Δ^k containing e. Indeed, for any covariate that is not frequent enough in Δ^k, the monotonicity property ensures that any covariate-set that contains

that covariate cannot be active. The following proposition shows that this pruning step does not eliminate any valid active set (proof is in the appendix):

Proposition 4.1. *If for a superset r of a newly processed set s where $|s| = k$ and $|r| = k+1$, all subsets s' of r of size k have been processed (i.e. r is eligible to be active after s is processed), then r is included in the set Z returned by* `GenerateNewActiveSets`.

The explicit verification step of whether all possible subsets of r of one less size belongs to Δ^k is necessary, i.e., the above optimization only prunes some candidate sets that are guaranteed not to be active. For instance, consider $s = \{2,3\}$, $k = 2$, and $\Delta^2 = \{\{1,2\},\{1,3\},\{3,5\},\{5,6\}\} \cup \{\{2,3\}\}$. For the superset $r = \{2,3,5\}$ of s, all of $2,3,5$ have support of ≥ 2 in Δ^2, but this r cannot become active yet, since the subset $\{2,5\}$ of r does not belong to Δ^2.

Finally, the following theorem states the correctness of the `DAME` algorithm (proof is in the appendix).

Theorem 4.2. ***(Correctness)*** *The* `DAME` *algorithm solves the AME problem.*

Once the problem is solved, the main matched groups can be used to estimate treatment effects, by considering the difference in outcomes between treatment and control units in each group, and possibly smoothing the estimates from the matched groups to prevent overfitting of treatment effect estimates.

5 Almost Matching Exactly with Adaptive Weights

We now generalize the AME framework so the weights are adjusted adaptively for each subset of variables. The weights are chosen using machine learning on a hold-out training set. Let us consider a trivial variation of the AME problem with fixed weights and then generalize it to handle adaptive weights.

Almost Matching Exactly with Fixed Weights, Revisited: We will use squared rewards w_j^2 this time. For a given treatment unit u with covariates $\mathbf{x}_u$, compute the following, which is the maximum sum of rewards $\{w_j^2\}_{j=1,..,p}$ we can attain for a valid matched group (that contains at least one control unit):

$$\boldsymbol{\theta}^{u*} \in \text{argmax}_{\boldsymbol{\theta}\in\{0,1\}^p} \boldsymbol{\theta}^T(\mathbf{w}\circ\mathbf{w}) \quad \text{s.t.}$$
$$\exists \ell \text{ with } T_\ell = 0 \text{ and } \mathbf{x}_\ell \circ \boldsymbol{\theta} = \mathbf{x}_u \circ \boldsymbol{\theta}. \qquad (2)$$

The solution to this is an indicator of the optimal set of covariates to match unit u on. For treatment unit u, again, the **main matched group** for u contains all units ℓ so that $\mathbf{x}_u \circ \boldsymbol{\theta}^{u*} = \mathbf{x}_\ell \circ \boldsymbol{\theta}^{u*}$. Now we provide the (more general) adaptive version of AME.

Algorithm 3: Procedure `GenerateNewActiveSets`

Input : s a newly dropped set of size k,
Δ the set of previously processed sets
Initialize: $Z = \varnothing$ (`stores new active sets`)
$\Delta^k = \{\delta \in \Delta \mid size(\delta) = k\} \cup \{s\}$ (`compute all subsets of` Δ `of size` k `and also include` s)
$\Gamma = \{\alpha \mid \alpha \in \delta \text{ and } \delta \in \Delta^k\}$ (`get all the covariates contained in sets in` Δ^k)
$\mathcal{S}_e$ = support of covariate e in Δ^k
$\Omega = \{\alpha \mid \alpha \in \Gamma \text{ and } \mathcal{S}_\alpha \geq k\} \setminus s$ (`get the covariates not in` s `that have enough support`)
if $\{\forall e \in s : \mathcal{S}_e \geq k\}$ *(if all covariates in s have enough support in Δ^k)* **then**
  for *all* $\alpha \in \Omega$ *(generate new active set)* **do**
    $r = s \cup \{\alpha\}$
    if *all subsets* $s' \subset r$, $|s'| = k$, *belong to* Δ^k **then**
      add r to Z (`add newly active set` r `to` Z)
return Z

Example (follow line number correspondence)
1. $s = \{2,3\}$, $k = 2$,
$\Delta = \{\{1\},\{2\},\{3\},\{5\},\{1,2\},\{1,3\},\{1,5\}\}$
2. $Z = \varnothing$
3. $\Delta^2 = \{\{1,2\},\{1,3\},\{2,3\},\{1,5\}\}$
4. $\Gamma = \{1,2,3,5\}$
5. $\mathcal{S}_1 = 3, \mathcal{S}_2 = 2, \mathcal{S}_3 = 2, \mathcal{S}_5 = 1$
6. $\Omega = \{1,2,3\} \setminus \{2,3\} = \{1\}$
7. $True$: both 1 and 2 have support ≥ 2
8. $\alpha = 1$ (only one value)
9. $r = \{2,3\} \cup \{1\} = \{1,2,3\}$
10. $True$ (subsets of r of size 2 are $\{1,2\},\{1,3\},\{2,3\}$)
11. $Z = \{\{1,2,3\}\}$
12. **return** $Z = \{\{1,2,3\}\}$

Full Almost Matching Exactly (Full-AME): Denote $\boldsymbol{\theta} \in \{0,1\}^p$ as an indicator vector for a subset of covariates to match on. Define the matched group for unit u with respect to covariates $\boldsymbol{\theta}$ as the units that match u exactly on the covariates $\boldsymbol{\theta}$:

$$\mathcal{MG}_{\boldsymbol{\theta}}(u) = \{v : \mathbf{x}_v \circ \boldsymbol{\theta} = \mathbf{x}_u \circ \boldsymbol{\theta}\}.$$

The usefulness of a set of covariates $\boldsymbol{\theta}$ is now determined by *how well they can be used together to make out-of-sample predictions.* Specifically, the prediction error $\texttt{PE}(\theta)$ is defined with respect to a class of functions $\mathcal{F}$ as: $\texttt{PE}_{\mathcal{F}}(\boldsymbol{\theta}) = \min_{f\in\mathcal{F}} \mathbb{E}(f(X \circ \boldsymbol{\theta}, T) - Y)^2$, where the expectation is taken over X, T and Y. Its empirical counterpart is defined with respect to a separate random sample from the distribution, used as a training set $\{\mathbf{x}_i^{tr}, T_i^{tr}, y_i^{tr}\}_{i\in \text{ training}}$, specifically:

$$\widehat{\texttt{PE}}_{\mathcal{F}}(\boldsymbol{\theta}) = \min_{f\in\mathcal{F}} \sum_{i\in \text{ training}} (f(\mathbf{x}_i^{tr} \circ \boldsymbol{\theta}, T_i^{tr}) - y_i^{tr})^2.$$

The training set is only used to calculate prediction error, not for matching. Using this, the best prediction error we could hope to achieve for a nontrivial matched group containing treatment unit u uses the following covariates for matching:

$$\boldsymbol{\theta}_u^* \in \arg\min_{\theta} \widehat{\mathrm{PE}}_{\mathcal{F}}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \exists \ell \in \mathcal{MG}_{\boldsymbol{\theta}}(u) \text{ where } T_\ell = 0$$

The **main matched group** for u is defined as $\mathcal{MG}_{\boldsymbol{\theta}_u^*}(u)$. The goal of the Full-AME problem is to find the main matched group $\mathcal{MG}_{\boldsymbol{\theta}_u^*}(u)$ for all units u.

The class of functions $\mathcal{F}$ can include nonlinear functions. We can use variable importance measures for prediction on $\widehat{\mathrm{PE}}_{\mathcal{F}}$ such as permutation importance (also called model reliance) to determine the variable's weight. If $\mathcal{F}$ includes linear models, the weight w_j for feature j would be the absolute value of feature j's coefficient.

The Full-AME problem reduces to the fixed-squared-weights version under specific conditions, such as when $\mathcal{F}$ is a single function f, which is linear with fixed linear weights $(\mathbf{w}, w_T)$ and $f(\mathbf{x}\circ\boldsymbol{\theta}, T) = (\mathbf{w}\circ\boldsymbol{\theta})^T(\mathbf{x}\circ\boldsymbol{\theta}) + w_T T$, where $\mathbf{w}$ is the ground-truth coefficient vector that generates y, and $\widehat{\mathrm{PE}}_{\boldsymbol{\theta}}$ is determined by the sum of w_j^2 weights for covariates determined by the feature-selector vector $\boldsymbol{\theta}$. This reduction is discussed formally by Wang et al. (2017).

In order to solve Full-AME, a step is needed in Algorithm 1 at the top of the while loop that updates the weights for each covariate-set we could choose at that iteration. In particular, we let

$$s_{(h)}^* \in \arg\min_{s \in \Lambda_{(h-1)}} \widehat{\mathrm{PE}}(\theta_s),$$

where $\Lambda_{(h-1)}$ is the active set of covariates, and the predictive error is computed over the training set with respect to a pre-specified class of models, $\mathcal{F}$. In the implementation in this paper we consider linear functions fit separately on the treated and the control units in the training set using ridge regression (that is, we add a ridge penalty to Eq (5)).

5.1 Early Stopping of DAME

It is important that DAME be *stopped early* when the quality of matches produced falls. In dropping covariates, its prediction error $\widehat{\mathrm{PE}}_{\mathcal{F}}$ should never increase too far above its original value using all the covariates. This ensures the quality of every matched group: the covariates $\boldsymbol{\theta}_u^*$ for every matched group thus obey $\widehat{\mathrm{PE}}_{\mathcal{F}}(\boldsymbol{\theta}_u^*) < \min_{\boldsymbol{\theta}} \widehat{\mathrm{PE}}_{\mathcal{F}}(\boldsymbol{\theta}) + \epsilon$, where the choice of ϵ (perhaps 5%) determines stopping. As such the while loop in Algorithm 1 should not only check whether there are more units to match, but also whether the predictive error has increased too much.

5.2 Hybrid FLAME-DAME

The DAME algorithm solves the Full-AME problem, whereas FLAME (Wang et al., 2017) approximates its solution. This is because FLAME uses backwards feature selection, whereas DAME calculates the solution without approximation. For problems with many features, we can use FLAME to remove the less relevant features, and then switch to DAME when we start to remove some of the more influential features. This hybrid algorithm scales substantially better, possibly without any noticeable loss in the quality of matches.

Matching-after-learning-to-stretch (MALTS) (Parikh et al., 2018) has been combined with FLAME and DAME to handle mixed real and categorical covariates.

5.3 Other Estimands

While CATEs are the most granular estimands, aggregate estimands such as Average Treatment Effect (ATE) and Average Treatment Effect on the Treated (ATT) may be of interest. Since DAME matches with replacement, standard techniques (e.g., frequency weights) should be used (Stuart, 2010; Abadie et al., 2004).

6 SIMULATIONS

We present results under several data generating processes. We show that DAME produces higher quality matches than popular matching methods such as 1-PSNNM (propensity score nearest neighbor matching) and Mahalanobis distance nearest neighbor matching, and better treatment effect estimates than black box machine learning methods such as Causal Forest (which is not a matching method, and is not interpretable). The 'MatchIt' R-package (Ho et al., 2011) was used to perform 1-PSNNM and Mahalanobis distance nearest neighbor matching ('Mahalanobis'). For Causal Forest, we used the 'grf' R-package (Athey et al., 2019). DAME also improves over FLAME (Wang et al., 2017) with regards to the quality of matches. Other matching methods (optmatch, cardinality match) do not scale to large problems and thus needed to be omitted.

Throughout this section, the outcome is generated with $y = \sum_i \alpha_i x_i + T\sum_{i=1} \beta_i x_i + T\cdot U \sum_{i,\gamma,\gamma>i} x_i x_\gamma$ where $T \in \{0,1\}$ is the binary treatment indicator. This generation process includes a baseline linear effect, linear treatment effect, and quadratic (nonlinear) treatment effect. We vary the distribution of covariates, coefficients (α's, β's, U), and the fraction of treated units. We report conditional average treatment effects on the treated.

6.1 Presence of Irrelevant Covariates

A basic sanity check for matching algorithms is how sensitive they are to irrelevant covariates. To that end, we run experiments with a majority of the covariates being irrelevant to the outcome. For important covariates $1 \leq i \leq 5$ let $\alpha_i \sim N(10s, 1)$ with $s \sim \text{Uniform}\{-1, 1\}$, $\beta_i \sim N(1.5, 0.15)$, $x_i \sim \text{Bernoulli}(0.5)$. For unimportant covariates $5 < i \leq 15$, $x_i \sim \text{Bernoulli}(0.1)$ in the control group and $x_i \sim \text{Bernoulli}(0.9)$ in the treatment group so there is little overlap between treatment and control distributions. This simulation generates 15000 control units, 15000 treatment units, 5 important covariates and 10 irrelevant covariates. **Results:** In Figure 1, DAME (even with early stopping) runs to the end and matches on all units because the stopping criteria is never met. In this figure, DAME finds all high-quality matches even after important covariates are dropped. In contrast, FLAME achieves the optimal result before dropping any important covariates and generates some poor matches after dropping important covariates. However, even FLAME's worst case scenario is better than the comparative methods, all of which perform poorly in the presence of irrelevant covariates. Causal Forest is especially ill suited for this case.

6.2 Exponentially Decaying Covariates

An advantage of DAME over FLAME is that it produces more high quality matches before resorting to lower quality matches. To test this, we considered covariates of decaying importance, letting the α parameters decrease exponentially as $\alpha_i = 64 \times \left(\frac{1}{2}\right)^i$. We evaluated performance when $\approx 30\%$ and 50% of units were matched. **Results:** As Figure 2 shows, DAME matches on more covariates, yielding better estimates than FLAME.

6.3 Imbalanced Data

Imbalance is common in observational studies: there are often substantially more control than treatment units. The data for this experiment has covariates with decreasing importance. A fixed batch of 2000 treatment and 40000 control units were generated. We sampled from the controls to construct different imbalance ratios: 40000 in the most imbalanced case (Ratio 1), then 20000 (Ratio 2), and 10000 (Ratio 3). **Results:** Table 1 reveals that FLAME and DAME outperform the nearest neighbor matching methods. DAME is distinctly better than FLAME. Additionally, DAME has an average of 4 covariates not matched on, with $\approx 84\%$ of units matched on all but 2 covariates, whereas FLAME averages 7 covariates not matched on and only $\approx 25\%$ units matched on all but 2 covariates. Detailed results are in the longer version (Liu et al., 2018).

Table 1: MSE for different imbalance ratios

	Mean Squared Error (MSE)		
	Ratio 1	Ratio 2	Ratio 3
DAME	**0.47**	**0.83**	**1.39**
FLAME	0.52	0.88	1.55
Mahalanobis	26.04	48.65	64.80
1-PSNNM	246.08	304.06	278.87

6.4 Run Time Evaluation

We compare the run time of DAME with a brute force solution (AME Solution 1 described in Section 3). All experiments were run on an Ubuntu 16.04.01 system with Intel Core i7 Processor (Cores: 8, Speed: 3.6 GHz), 8 GB RAM. **Results:** As shown in Figure 3, FLAME provides the best run-time performance because it incrementally reduces the number of covariates, rather than solving Full-AME. On the other hand, as shown in the previous simulations, DAME produces high quality matches that the other methods do not. It solves the AME much faster than brute force. The run time for DAME could be further optimized through simple parallelization of the checking of active sets.

6.5 Missing Data

Missing data problems are complicated in matching. Normally one would impute missing values, but matches become less interpretable when matching on imputed values. If we match only on the raw values, DAME has an advantage over FLAME because it can simply match on as many non-missing relevant covariates as possible. When data are imputed, DAME still maintains an advantage over FLAME, possibly because it can match on more raw covariate values and fewer imputed values. Details of the experiments are in the longer version (Liu et al., 2018).

7 BREAKING THE CYCLE OF DRUGS AND CRIME

Breaking The Cycle (BTC) (Harrell et al., 2006) is a social program conducted in several U.S. states designed to reduce criminal involvement and substance abuse among current offenders. We study the effect of participating in the program on reducing non-drug future arrest rates. The details of the data and our results are in Appendix D. We compared CATE predictions of DAME and FLAME to double check the performance of a black box support vector machine (SVM) approach that predicts positive, neutral, or negative treatment effect for each individual. The result is that DAME and the SVM approach agreed on most of the exactly

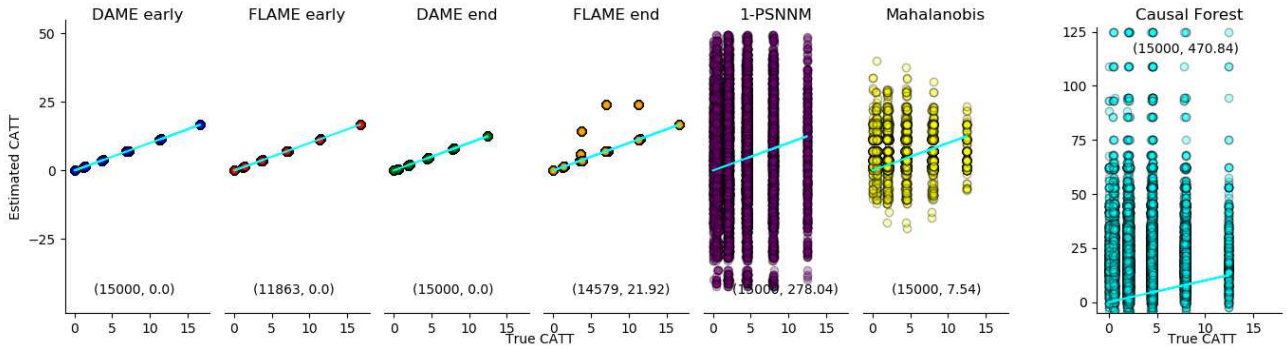

Figure 1: Estimated CATT vs. True CATT (Conditional Average Treatment Effect on the Treated). DAME and FLAME perfectly estimate the CATTs before dropping important covariates. DAME matches all units without dropping important covariates, but FLAME needs to stop early in order to avoid poor matches. All other methods are sensitive to irrelevant covariates and give poor estimates. The two numbers on each plot are the number of matched units and MSE.

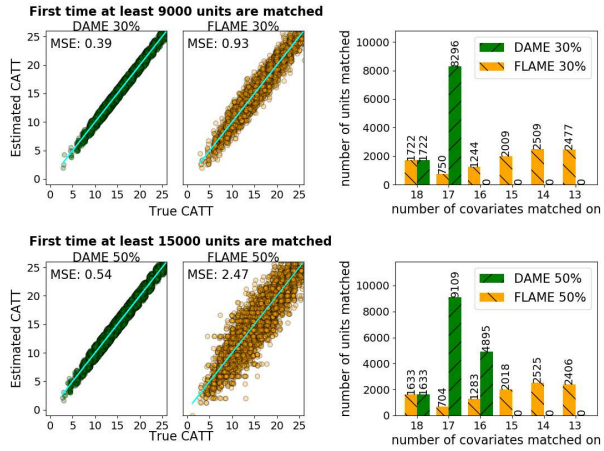

Figure 2: DAME makes higher quality matches early on. Rows correspond to stopping thresholds (top row 30%, bottom row 50%). DAME matches on more covariates than FLAME, yielding lower MSE from matched groups.

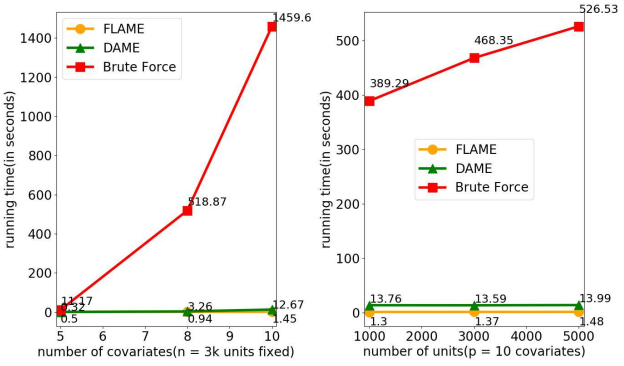

Figure 3: Run-time comparison between DAME FLAME, and brute force. *Left:* varying number of units. *Right:* varying number of covariates.

matched units. All of the units for which exact matching predicted approximately zero treatment effect all have a "neutral" treatment effect predicted label from the SVM. Most other predictions were similar between the two methods. There were only few disagreements between the methods. Upon further investigation, we found that the differences are due to the fact that DAME is a matching method and not a modeling method; the estimates could be smoothed afterwards if desired to create a model. In particular, one of the two disagreeing predictions between the SVM and DAME has a positive treatment CATE prediction, but it was closer in Hamming distance to units predicted to have negative treatment effects. With smoothing, its predicted CATE may have also become negative.

8 CONCLUSIONS

DAME produces matches that are of high quality. Its estimates of individualized treatment effects are as good as the (black box) machine learning methods we have tried. Other methods can match individuals together whose covariates look nothing alike, whereas the matches from DAME are interpretable and meaningful, because they are almost exact; units are matched on covariates that together can be used to predict outcomes accurately. Code is publicly available at: `https://github.com/almost-matching-exactly/DAME` .

Acknowledgements:

This work was supported in part by NIH award 1R01EB025021-01, NSF awards IIS-1552538 and IIS-1703431, a DARPA award under the L2M program, and a Duke University Energy Initiative Energy Research Seed Fund (ERSF).

References

A. Abadie, D. Drukker, J. L. Herr, and G. W. Imbens. Implementing matching estimators for average treatment effects in stata. *The Stata Journal*, 4(3): 290–311, 2004.

R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, 1994.

S. Athey, J. Tibshirani, and S. Wager. Generalized Random Forests. *The Annals of Statistics*, 47(2): 1148–1178, 2019.

A. Belloni, V. Chernozhukov, and C. Hansen. Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2):608–650, 2014.

W. G. Cochran and D. B. Rubin. Controlling bias in observational studies: A review. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 417–446, 1973.

M. H. Farrell. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics*, 189(1):1–23, 2015.

S. Goh and C. Rudin. A minimax surrogate loss approach to conditional difference estimation. *ArXiv e-prints: arXiv:1803.03769*, Mar. 2018.

A. V. Harrell, D. Marlowe, and J. Merrill. Breaking the cycle of drugs and crime in Birmingham, Alabama, Jacksonville, Florida, and Tacoma, Washington, 1997-2001. Ann Arbor, MI: Inter-university Consortium for Political and Social Research, 2006-03-30, 2006.

D. Ho, K. Imai, G. King, and E. Stuart. Matchit: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software, Articles*, 42(8):1–28, 2011.

S. M. Iacus, G. King, and G. Porro. Multivariate matching methods that are monotonic imbalance bounding. *Journal of the American Statistical Association*, 106 (493):345–361, 2011.

S. M. Iacus, G. King, and G. Porro. Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20:1–24, 2012.

Y. Liu, A. Dieng, S. Roy, C. Rudin, and A. Volfovsky. Interpretable almost-exact matching for causal inference. *arXiv e-prints: arXiv:1806.06802*, Jun 2018.

M. Morucci, M. Noor-E-Alam, and C. Rudin. Hypothesis tests that are robust to choice of matching method. *ArXiv e-prints, arXiv:1812.02227*, Dec. 2018.

M. Noor-E-Alam and C. Rudin. Robust nonparametric testing for causal inference in observational studies. *Optimization Online*, Dec, 2015.

H. Parikh, C. Rudin, and A. Volfovsky. MALTS: Matching After Learning to Stretch. *arXiv e-prints: arXiv:1811.07415*, Nov 2018.

J. A. Rassen and S. Schneeweiss. Using high-dimensional propensity scores to automate confounding control in a distributed medical product safety surveillance system. *Pharmacoepidemiology and Drug Safety*, 21(S1):41–49, 2012.

P. R. Rosenbaum. Imposing minimax and quantile constraints on optimal matching in observational studies. *Journal of Computational and Graphical Statistics*, 26(1), 2016.

D. B. Rubin. Matching to remove bias in observational studies. *Biometrics*, 29(1):159–183, Mar. 1973a.

D. B. Rubin. The use of matched sampling and regression adjustment to remove bias in observational studies. *Biometrics*, 29(1):185–203, Mar. 1973b.

D. B. Rubin. Multivariate matching methods that are equal percent bias reducing, I: Some examples. *Biometrics*, 32(1):109–120, Mar. 1976.

D. B. Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371): 591–593, 1980.

S. Schneeweiss, J. A. Rassen, R. J. Glynn, J. Avorn, H. Mogun, and M. A. Brookhart. High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. *Epidemiology (Cambridge, Mass.)*, 20(4):512, 2009.

E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science: a Review Journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

T. Wang, S. Roy, C. Rudin, and A. Volfovsky. FLAME: A fast large-scale almost matching exactly approach to causal inference. *arXiv e-prints: arXiv:1707.06315*, July 2017.