Chapter 8

Appendix

8.1 Naïve AME solutions

AME Solution 1 (quadratic in n, linear in p): For all treatment units t, we (i) iterate over all control units c, (ii) find the vector $\boldsymbol{\theta}_{tc} \in \{0,1\}^p$ with value 1 if there is a match on the values of the corresponding covariates, and 0 otherwise, (iii) find the control unit(s) with the highest value of $\boldsymbol{\theta}_{tc}^T\mathbf{w}$, and (iv) return them as the main matched group for the treatment unit t (and compute the auxiliary group). Whenever a previously matched unit α is matched to a previously unmatched unit η, record the η's main matched group as an auxiliary group for the previously matched unit α. When all units are 'done' (all units are either matched already or cannot be matched) then stop, and compute the CATE for each treatment and control unit using its main matched group. If a unit belongs to auxiliary matched groups then its outcome is used for computing both its own CATE (in its own main matched group) and the CATEs of units for whom it is in an auxiliary group (e.g., α will be used to compute η's estimated CATE). This algorithm is polynomial in both n and p, however, the quadratic time complexity in n also makes this approach impractical for large datasets (for instance, when we have more than a million units with half being treatment units).

AME Solution 2 (order $n \log n$, exponential in p:) This approach solves the AMER problem simultaneously for all treatment and control units for a fixed weight vector $\mathbf{w}$. First, (i) enumerate every $\boldsymbol{\theta} \in \{0,1\}^p$ (which serves as an indicator for a subset of covariates), (ii) order the $\boldsymbol{\theta}$'s according to $\boldsymbol{\theta}^T\mathbf{w}$, (iii) call `GroupedMR` for every $\boldsymbol{\theta}$ in the predetermined order, (iv) the first time each unit is matched during a `GroupedMR` procedure, mark that unit with a 'done' flag, and record its corresponding main matched group and, to facilitate matching with replacement, (v) whenever a previously matched unit is matched to a previously unmatched unit, record this main matched group as an auxiliary group. When all units are 'done' (all units are either matched already or cannot be matched) then stop, and compute the CATE for each treatment and control unit using its main matched group. Each unit's outcome will be used to estimate CATEs for every auxiliary group that it is a member of, as before. Although this approach exploits the efficient 'group by' function (e.g., provided in database (SQL) queries), which can be implemented in $O(n \log n)$ time by sorting the units, iterating over all possible vectors $\boldsymbol{\theta} \in \{0,1\}^p$ makes this approach unsuitable for practical purposes (exponential in p).

8.2 Proof of Proposition 4.0.1

Proposition 4.0.1 *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`*.*

Proof. Suppose all subsets of r of size k are already processed and belong to Δ^k. Let f be the covariate in $r \setminus s$. Clearly, f would appear in Δ^k, since at least one subset $s' \neq s$ of r of size k would contain f, and $s' \in \Delta^k$. Further all covariates in r, including f and those in s will have support at least k in Δ^k. To see this, note that there are $k+1$ subsets of r of size k, and each covariate in r appears in exactly k of them. Hence $f \in \Omega$, which the set of high support covariates. Further, the 'if' condition to check minimum support for all covariates in s is also satisfied. In addition, the final ‘if’ condition to eliminate false positives is satisfied too by assumption (that all subsets of r are already processed). Therefore r will be included in Z returned by the procedure. □

8.3 Proof of Theorem 4.0.2

Theorem 4.0.2 ***(Correctness)*** *The* `DAME` *algorithm solves the AME problem.*

Proof. Consider any treatment unit t. Let s be the set of covariates in its main matched group returned in `DAME` (the while loop in `DAME` runs as long as there is a treated unit and the stopping criteria have not been met, and the `GroupedMR` returns the main matched group for every unit when it is matched for the first time). Let $\boldsymbol{\theta}_s$ be the indicator vector of s (see Eq. 4.1). Since the `GroupedMR` procedure returns a main matched group only if it is a *valid* matched group containing at least one treated and one control unit (see Algorithm 2), and since all units in the matched group on s have the same value of covariates in $\mathcal{J} \setminus s$, there exists a unit ℓ with $T_\ell = 0$ and $\mathbf{x}_\ell \circ \boldsymbol{\theta}_s = \mathbf{x}_t \circ \boldsymbol{\theta}_s$.

Hence it remains to show that the covariate set s in the main matched group for t corresponds to the maximum weight $\boldsymbol{\theta}^T\mathbf{w}$. Assume that there exists another covariate-set r such that $\boldsymbol{\theta}_r^T\mathbf{w} > \boldsymbol{\theta}_s^T\mathbf{w}$, there exists a unit ℓ' with $T_{\ell'} = 0$ and $\mathbf{x}_{\ell'} \circ \boldsymbol{\theta}_r = \mathbf{x}_t \circ \boldsymbol{\theta}_r$, and gives the maximum weight $\boldsymbol{\theta}_r^T\mathbf{w}$ over all such r.

(i) r cannot be a (strict) subset of s, since `DAME` ensures that all subsets are processed before a superset is processed to satisfy the downward closure property in Proposition 3.0.1.

(ii) r cannot be a (strict) superset of s, since it would violate the assumption that $\boldsymbol{\theta}_r^T\mathbf{w} > \boldsymbol{\theta}_s^T\mathbf{w}$ for non-negative weights.

(iii) Assume that r and s are incomparable (there exist covariates in both $r \smallsetminus s$ and $s \smallsetminus r$). Suppose the active set s was chosen in iteration h. If r was processed in an earlier iteration $h' < h$, since r forms a valid matched group for t, it would give the main matched group for t violating the assumption.

Given (i)–(iii) we argue that r must be active at the start of iteration h, and will be chosen as the best covariate set in iteration h, leading to a contradiction.

Note that we start with all singleton sets as active sets in $\Delta_{(0)} = \{\{1\}, \cdots, \{p\}\}$ in the DAME algorithm. Consider any singleton subset $r_0 \subseteq r$ (comprising a single covariate in r). Due to the downward closure property in Proposition 3.0.1, $\boldsymbol{\theta}_{r_0}^T \mathbf{w} \geq \boldsymbol{\theta}_r^T \mathbf{w} > \boldsymbol{\theta}_s^T \mathbf{w}$. Hence all of the singleton subsets of r will be processed in earlier iterations $h' < h$, and will belong to the set of processed covariate sets $\Lambda_{(h-1)}$.

Repeating the above argument, consider any subset $r' \subseteq r$. It holds that $\boldsymbol{\theta}_{r'}^T \mathbf{w} \geq \boldsymbol{\theta}_r^T \mathbf{w} > \boldsymbol{\theta}_s^T \mathbf{w}$. All subsets r' of r will be processed in earlier iterations $h' < h$ starting with the singleton subsets of r. In particular, all subsets of size $|r| - 1$ will belong to $\Lambda_{(h-1)}$. As soon as the last of those subsets is processed, the procedure GenerateNewActiveSets will include r in the set of active sets in a previous iteration $h' < h$. Hence if r is not processed in an earlier iteration, it must be active at the start of iteration h, leading to a contradiction.

Hence for all treatment units t, the covariate-set r giving the maximum value of $\boldsymbol{\theta}_r^T \mathbf{w}$ will be used to form the main matched group of t, showing the correctness of the DAME algorithm. □

8.4 Proof of Lemma 5.2.1

Since the result is exactly symmetric when non-instrumented units are matched we prove it only for the case when instrumented units are matched. Assume $\mathbf{w} \in \mathbb{R}^p$. For a given unit i with $z_i = 1$, suppose we could find a $\boldsymbol{\theta}^*$ as defined in the AME-IV problem. Let us define another unit k with $z_k = 0$, and $\mathbf{x}_k \circ \boldsymbol{\theta}^* = \mathbf{x} \circ \boldsymbol{\theta}^*$, by definition of $\mathcal{MG}(\boldsymbol{\theta}^*, \mathbf{x}_i)$ it must be that $\mathbf{x}_k \in \mathcal{MG}(\boldsymbol{\theta}^*, \mathbf{x}_i)$. So $\mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_k]} = J - \boldsymbol{\theta}^*$, where J is a vector of length p that has all entries equals to 1.

Assume there is another unit j with $z_j = 0$, and $j \neq k$.

If $j \in \mathcal{MG}(\boldsymbol{\theta}^*, \mathbf{x}_i)$, then $\mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_j]} = J - \boldsymbol{\theta}^*$. So

$$\mathbf{w}^T \mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_k]} = \mathbf{w}^T (J - \boldsymbol{\theta}^*) = \mathbf{w}^T \mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_j]}$$

If $j \notin \mathcal{MG}(\boldsymbol{\theta}^*, \mathbf{x}_i)$, let us define $\boldsymbol{\theta}^j = J - \mathbb{1}_{[\mathbf{x} \neq \mathbf{x}_j]}$, obviously $\boldsymbol{\theta}^j \neq \boldsymbol{\theta}^*$. Since $\boldsymbol{\theta}^* \in \arg\max_{\boldsymbol{\theta} \in \{0,1\}^p} \boldsymbol{\theta}^T \mathbf{w}$, we have:

$$\begin{aligned}
\mathbf{w}^T \mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_k]} &= \mathbf{w}^T (J - \boldsymbol{\theta}^*) \\
&= \mathbf{w}^T - \mathbf{w}^T \boldsymbol{\theta}^* \\
&< \mathbf{w}^T - \mathbf{w}^T \boldsymbol{\theta}^j \\
&= \mathbf{w}^T (J - \boldsymbol{\theta}^j) \\
&= \mathbf{w}^T \mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_j]}.
\end{aligned}$$

Therefore,

$$k \in \underset{\substack{j=1,\dots,n \\ Z_j = 0}}{\arg\min}\, \mathbf{w}^T \mathbb{1}_{[\mathbf{x}_i \neq \mathbf{x}_j]}.$$

This concludes the proof.

8.5 Asymptotic Variance and Confidence Intervals for LATE Estimates

To construct estimators for the variance of $\hat{\lambda}$ we use an asymptotic approximation, that is, we will try to estimate the asymptotic variance of $\hat{\lambda}$, rather than its small sample variance. The strategy we use to do this is the same as [IR15], with the difference that our data is grouped: we adapt their estimators to grouped data using canonical methods for stratified sampling. In order to define asymptotic quantities for our estimators, we must marginally expand the definitions of potential outcomes introduced in our paper. In practice, while our framework has been presented under the assumption that the potential outcomes and treatments are fixed, we now relax that assumption and instead treat $y_i(1), y_i(0), t_i(1), t_i(0)$ as realizations of random variables $Y_i(1), Y_i(0), T_i(1), T_i(0)$, which are drawn from some unknown distribution $f(Y_i(1), Y_i(0), T_i(1), T_i(0))$. In this case the SUTVA assumption requires that each set of potential outcomes and treatments is independently drawn from the same distribution for all units. As usual, lowercase versions of the symbols above denote observed realizations of the respective random variables.

Recall as well that in this scenario we have a set of m matched groups $\mathcal{MG}_1, \ldots \mathcal{MG}_m$ indexed by ℓ, such that each unit is only in one matched group. We denote the number of units in matched group ℓ that have $z_i = 1$ with n_ℓ^1 and the number of units in matched group ℓ with $z_i = 0$ with n_ℓ^0. Finally the total number of units in matched group ℓ is $n_\ell = n_\ell^0 + n_\ell^1$.

We make all the assumptions listed in Section 5.1 but we must require a variant of (A3), to be used instead of it. This assumption is:

(A3') $\Pr(Z_i = 1 | i \in \mathcal{MG}_\ell) = \Pr(Z_k = 1 | k \in \mathcal{MG}_\ell) = \frac{n_\ell^1}{n_\ell}, \forall i, k.$

That is, if two units are in the same matched group, then they have the same probability of receiving the instrument. This probability will be equal to the ratio of instrument 1 units to all units in the matched group because we hold these quantities fixed. Note that this more stringent assumption holds when matches are made exactly, and is common in variance computation for matching estimators (see, for example, [KKM+16]).

We keep our exposition concise and we do not give explicit definitions for our variance estimands. These are all standard and can be found in [IR15].

We have to start from estimating variances of observed potential outcomes and treatments within each matched group. We do so with the canonical approach:

$$
\begin{aligned}
\hat{s}^2_{\ell 0} &= \frac{1}{n^0_\ell - 1} \sum_{i \in \mathcal{MG}_\ell} \left(y_i(1 - z_i) - \frac{1}{n^0_\ell} \sum_{i \in \mathcal{MG}_\ell} y_i(1 - z_i) \right)^2 \\
\hat{s}^2_{\ell 1} &= \frac{1}{n^1_\ell - 1} \sum_{i \in \mathcal{MG}_\ell} \left(y_i z_i - \frac{1}{n^1_\ell} \sum_{i \in \mathcal{MG}_\ell} y_i z_i \right)^2 \\
\hat{r}^2_{\ell 0} &= \frac{1}{n^0_\ell - 1} \sum_{i \in \mathcal{MG}_\ell} \left(t_i(1 - z_i) - \frac{1}{n^0_\ell} \sum_{i \in \mathcal{MG}_\ell} t_i(1 - z_i) \right)^2 \\
&= 0 \\
\hat{r}^2_{\ell 1} &= \frac{1}{n^1_\ell - 1} \sum_{i \in \mathcal{MG}_\ell} \left(t_i z_i - \frac{1}{n^1_\ell} \sum_{i \in \mathcal{MG}_\ell} t_i z_i \right)^2,
\end{aligned}
$$

where: $\hat{s}^2_{\ell 0}$ is an estimator for the variance of potential responses for the units with instrument value 0 in matched group ℓ, $\hat{s}^2_{\ell 1}$ for the variance of potential responses for the units with instrument value 1 in matched group ℓ, $\hat{r}^2_{\ell 0}$ for the variance of potential treatments the units with instrument value 0 in matched group ℓ, and $\hat{r}^2_{\ell 1}$ is an estimator for the variance of potential treatments for the units with instrument value 1 in matched group ℓ. The fact that $\hat{r}^2_{\ell 0} = 0$ follows from Assumption A4.

We now move to variance estimation for the two ITTs. Conservatively biased estimators for these quantities are given in [IR15]. These estimators are commonly used in practice and simple to compute, hence why they are often preferred to unbiased but more complex alternative. We repeat them below:

$$\widehat{Var}(\widehat{ITT}_y) = \sum_{\ell=1}^{m} \left(\frac{n_\ell}{n}\right)^2 \left(\frac{\hat{s}_{\ell 1}^2}{n_\ell^1} + \frac{\hat{s}_{\ell 0}^2}{n_\ell^0}\right)$$

$$\widehat{Var}(\widehat{ITT}_t) = \sum_{\ell=1}^{m} \left(\frac{n_\ell}{n}\right)^2 \frac{\hat{r}_{\ell 1}^2}{n_\ell^1}.$$

To estimate the asymptotic variance of $\hat{\lambda}$ we also need estimators for the covariance of the two ITTs both within each matched group, and in the whole sample. Starting with the former, we can use the standard sample covariance estimator for $Cov(\widehat{ITT}_{y\ell}, \widehat{ITT}_{t\ell})$:

$$\begin{aligned}\widehat{Cov}(\widehat{ITT}_{y\ell}, \widehat{ITT}_{t\ell}) = & \frac{1}{n_\ell^1(n_\ell^1 - 1)} \\ & \times \sum_{i \in \mathcal{MG}_\ell} \left(y_i z_i - \frac{1}{n_\ell^1} \sum_{i \in \mathcal{MG}_\ell} y_i z_i\right) \\ & \quad \times \left(t_i z_i - \frac{1}{n_\ell^1} \sum_{i \in \mathcal{MG}_\ell} t_i z_i\right).\end{aligned}$$

The reasoning behind why we use only units with instrument value 1 to estimate this covariance is given in [IR15], and follows from A4. We can use standard techniques for covariance estimation in grouped data to combine the estimators above into an overall estimator for $Cov(\widehat{ITT}_y, \widehat{ITT}_t)$ as follows:

$$\widehat{Cov}(\widehat{ITT}_y, \widehat{ITT}_t) = \sum_{\ell=1}^{m} \left(\frac{n_\ell}{n}\right)^2 \widehat{Cov}(\widehat{ITT}_{y\ell}, \widehat{ITT}_{t\ell}).$$

Once all these estimators are defined, we can use them to get an estimate of the asymptotic variance of $\hat{\lambda}$. This quantity is obtained in [IR15] with an application of the delta method to convergence of the two ITTs. The final estimator for the asymptotic variance of $\hat{\lambda}$, which we denote by σ^2, is given by:

$$\begin{aligned}\hat{\sigma}^2 = \frac{1}{\widehat{ITT}_t^2}\widehat{Var}(\widehat{ITT}_y) + \frac{\widehat{ITT}_y^2}{\widehat{ITT}_t^4}\widehat{Var}(\widehat{ITT}_t) \\ - 2\frac{\widehat{ITT}_y}{\widehat{ITT}_t^3}\widehat{Cov}(\widehat{ITT}_y, \widehat{ITT}_t).\end{aligned}$$

Using this variance, $1 - \alpha\%$ asymptotic confidence intervals can be computed in the standard way.

8.6 The FLAME-IV Algorithm

Basic Matching Requirement (R1): There should be at least one instrumented and one noninstrumented unit in each matched group.

Algorithm 4 presents the matching algorithm for FLAME-IV. Initially, the input with n units is given as $D = (X, Y, T, Z)$, where X (and $n \times p$ matrix) denotes the covariates, Y (an $n \times 1$ vector) is the outcome, T (an $n \times 1$ vector) is the treatment, and Z is the instrument. The covariates are indexed with $J = 1, \cdots, p$.

Let $\mathcal{MG}_l$ represent a set of all matched groups at iteration h of the FLAME-IValgorithm. At iteration h of the algorithm, FLAME-IVcomputes a subset of the matched groups $\mathcal{MG}_h$ such that, for each matched group $mg \in \mathcal{MG}_l$, there is at least one treated and one control unit. Note that it is possible for $\mathcal{MG}_h = \varnothing$, in which case no matched groups are returned in that iteration. M_u denotes the iteration when a unit u is matched. Overloading notation, let M_{mg} denote the iteration when a matched group mg is formed. Hence if a unit u belongs to a matched group mg, $M_u = M_{mg}$ (although not every u with $M_u = M_{mg}$ is in mg).

We use $D_h \subseteq D$ to denote the unmatched units and $J_h \subseteq J$ to denote the remaining

Algorithm 4: FLAME-IV Algorithm

```
Input  : (i) Input data $D = (X, Y, T, Z)$. (ii) holdout training set
         $D^H = (X^H, Y^H, T^H, Z^H)$.
Output : A set of matched groups $\{\mathcal{MG}_h\}_{h\geq 1}$ and ordering of covariates
         $j_1^*, j_2^*, ..,$ eliminated.
Initialize $D_0 = D = (X, Y, T, Z), J_0 = \{1, ..., p\}, h = 1, run = True$, $\mathcal{MG} = \varnothing$. ($h$
 is the index for iterations, $j$ is the index for covariates)
$(D_0^m, D_0 \setminus D_0^m, \mathcal{MG}_1) = GroupedMR(D_0, J_0)$.
while $run = True$ and $D_{h-1} \setminus D_{h-1}^m \neq \varnothing$ (we still have data to match) do
    $D_h = D_{h-1} \setminus D_{h-1}^m$ (remove matches)
    for $j \in J_{h-1}$ (temporarily remove one feature at a time and compute match
     quality) do
        $(D_h^{mj}, D_h \setminus D_h^{mj}, \mathcal{MG}_{temp}^j) = GroupedMR(D_h, J_{h-1} \setminus j)$.
        $D^{Hj} = [X^H(:, J_{h-1} \setminus j), Y^H, T^H, Z^H]$
        $q_{hj} = MQ(D_h^{mj}, D^{Hj})$
    if other stopping conditions are met, then
        $run = False$ (break from the while loop)
    $j_h^\star \in \arg\min_{j \in J_{h-1}} q_{hj}$: (choose feature to remove)
    $J_h = J_{h-1} \setminus j_h^\star$ (remove feature $j_h^\star$)
    $D_h^m = D_h^{mj^\star}$ and $\mathcal{MG}_h = \mathcal{MG}_{temp}^{j_h^\star}$ (newly matched data and groups)
    $h = h + 1$
return $\{\mathcal{MG}_h, D_h^m, J_h\}_{h\geq 1}$ (return all the matched groups and covariates
 used)
```

variables when iteration $h+1$ of the while loop starts (*i.e.*, after iteration h ends). Initially $J_0 = J$. While the algorithm proceeds, the algorithm drops one covariate $\pi(h)$ in each iteration (whether or not there are any valid non-empty matched groups), and therefore, $J_h = J \setminus \{\pi(j)_{j=1}^h\}$, $|J_h| = p - h$. All matched groups $mg \in \mathcal{MG}_h$ in iteration h use J_{h-1} as the subset of covariates on which to match.

The first call to `GroupedMR`: First we initialize the variables D_0, J_0, h, and run. The variable run is true as long as the algorithm is running, while $h \geq 1$ denotes an iteration. After the initialization step, the subroutine `GroupedMR` (see Algorithm 5) finds all of the exact matches in the data $D = D_0$ using *all* features $J = J_0$, such that each of the matched groups $mg \in \mathcal{MG}_1$ contains at least one instrumented and one uninstrumented observation (*i.e.*, satisfies constraint (R1)). The rest of the iterations

```
Algorithm 5: GroupedMR procedure
Input   : Unmatched Data $D^{um} = (X, Y, T, Z)$, subset of indexes of covariates
          $J^s \subseteq \{1, ..., p\}$.
Output  : Newly matched units $D^m$ using covariates indexed by $J^s$ where
          groups obey (R1),the remaining data as $D^{um} \setminus D^m$ and matched
          groups for $D^m$.
$M_{raw}$=group-by $(D^{um}, J^s)$ (form groups by exact matching on $J^s$)
$M$=prune($M_{raw}$) (remove groups not satisfying (R1))
$D^m$=Get subset of $D^{um}$ where the covariates match with $M$ (recover newly matched units)
return $\{D^m, D^{um} \setminus D^m, M\}$.
```

in the algorithm aim to find the best possible matches for the rest of the data by selectively dropping covariates as discussed in the previous section.

The while loop and subsequent calls to GroupedMR: At each iteration of the **while** loop, each feature is temporarily removed (in the **for** loop over j) and evaluated to determine if it is the best one to remove by running GroupedMR and computing the matched quality MQ. Since GroupedMR does not consider feature j (one less feature from the immediately previous iteration), there are fewer constraints on the matches, and it is likely that there will be new matches returned from this subroutine.

We then need to determine whether a model that excludes feature j provides sufficiently high quality matches and predictions. We would not want to remove j if doing so would lead to poor predictions or if it led to few new matches. Thus, MQ is evaluated by temporarily removing each j, and the j^* that is chosen for removal creates the most new matches and also does not significantly reduce the prediction quality. The algorithm always chooses the feature with largest MQ to remove, and remove it. After the algorithm chooses the feature to remove, the new matches and matched groups are stored. The remaining unmatched data are used for the next iteration $h + 1$.

Stopping Conditions: If we run out of unmatched data, the algorithm stops. There are also a set of **early-stop conditions** we use to stop algorithm in advance.

Early-Stop Conditions:

(1) There are no more covariates to drop.

(2) Unmatched units are either all instrumented or uninstrumented.

(3) The matching quality drops by 5% or more than the matching quality of exact matching.

Finally, the matched groups are returned along with the units and the features used for each set of matched groups formed in different iterations.

The key component in the Basic FLAME-IV algorithm (Algorithm 4) is the `GroupedMR` procedure (Algorithm 5). The steps of `GroupedMR` can be easily implemented in Java, Python, or R. In the next two subsections we give two efficient implementations of `GroupedMR`, using database queries and bit vector techniques.

8.7 Implementation of `GroupedMR` using Database (SQL) Queries

In this implementation, we keep track of matched units globally by keeping an extra column in the input database D called `is_matched`. For every unit, the value of `is_matched =` ℓ if the unit is matched in a valid group with at least one instrumented and one uninstrumented unit in iteration ℓ of Algorithm 4, and `is_matched = 0` if the unit is still unmatched. Therefore instead of querying the set of unmatched data D^{um} at each iteration (as in the input of Algorithm 5), at each iteration we query the full database D, and consider only the unmatched units for matching by checking the predicate `is_matched = 0` in the query. Let $A_1, \cdots, A_p$ be the covariates in J_s. The SQL query is described below:

```
WITH tempgroups AS
```

```
   (SELECT $A_1$, $A_2$, $\cdots$, $A_p$
    /*matched groups identified by covariate values*/
    FROM D
    WHERE $is\_matched$ = 0
     /*use data that are not yet matched*/
    GROUP BY $A_1$, $A_2$, $\cdots$, $A_p$
     /*create matched groups with identical covariates*/
    HAVING SUM(Z) >= 1 AND
        SUM(Z) <= COUNT(*)-1
    /*groups have >=1 instrumented, but not all instrumented*/
    ),
UPDATE D
SET $is\_matched$ = $\ell$
WHERE  EXISTS
    (SELECT $D.A_1$, $D.A_2$, $\cdots$, $D.A_p$
     FROM tempgroups S
     /*set of covariate values for valid groups*/
     WHERE $S.A_1$ = $D.A_1$
     AND $S.A_2$ = $D.A_2$
     AND $\cdots$ AND $S.A_p$ = $D.A_p$)
    AND $is\_matched$ = 0
```

The *WITH clause* computes a temporary relation *tempgroups* that computes the combination of values of the covariates forming 'valid groups' (*i.e.*, groups with at least one instrumented and one noninstrumented unit) on unmatched units. The *HAVING clause* of the SQL query discards groups that are invalid – since instruments Z takes binary values $0, 1$, for any valid group the sum of Z values will be strictly > 0

and < total number of units in the group. Then we update the population table D, where the values of the covariates of the existing units match with those of a valid group in *tempgroups*. Several optimizations of this basic query are possible and are used in our implementation. Setting the `is_matched` value to level ℓ (instead of a constant value like 1) helps us compute the conditional LATE for each matched group efficiently.

8.8 Implementation of `GroupedMR` using Bit Vectors

In this section we discuss an bit-vector implementation to the `GroupedMR` procedure discussed above. We will assign unit u's covariates to a single integer b_u. Unit u's covariates, appended with the instrumental variable indicator, will be assigned an integer b_u^+. Let us discuss how to compute b_u and b_u^+. Suppose $|J_s| = q$, and the covariates in J_s are indexed (by renumbering from J) as 0 to $q-1$. If the j-th covariate is $k_{(j)}$-ary ($k_{(j)} \geq 2$), we first rearrange the q covariates such that $k_{(j)} \geq k_{(j+1)}$ for all $0 \leq j \leq q-2$. Thus the (reordered) covariate values of unit u, $(a_{q-1}, a_{q-2}, \ldots, a_0)$, is represented by the number $b_u = \sum_{j=0}^{q-1} a_j k_{(j)}^j$. Together with the instrument indicator value $Z = z$, the set $(a_{q-1}, a_{q-2}, \ldots, a_0, z)$ for unit u is represented by the number $b_u^+ = z + \sum_{j=0}^{p-1} a_j k_{(j)}^{j+1}$. Since the covariates are rearranged so that $k_{(j)} \leq k_{(j+1)}$ for all $0 \leq j \leq q-2$, two units u and u' have the same covariate values if and only if $b_u = b_{u'}$. For each unit u, we count how many times b_u and b_u^+ appear, and denote them as c_u and c_u^+ respectively. (The counting is done by NumPy's `unique()` function.) To perform matching, we compute the b_u, b_u^+, c_u, c_u^+ values for all units and mark a unit as matched if its c_u value and c_u^+ value differ.

Proposition 8.8.1 guarantees the correctness of the bit-vector implementation.

Proposition 8.8.1. *A unit u is matched if and only if $c_u \neq c_u^+$, since the two counts*

Table 8.1: Example population table illustrating the *bit-vector* implementation.

1st variable	2nd variable	Z	b_u	b_u^+	c_u	c_u^+	matched?
0	2	0	6	18	1	1	No
1	1	0	4	11	2	1	Yes
1	0	1	1	3	1	1	No
1	1	1	4	12	2	1	Yes

Notes: Here the second unit and the fourth unit are matched to each other while the first and third units are left unmatched.

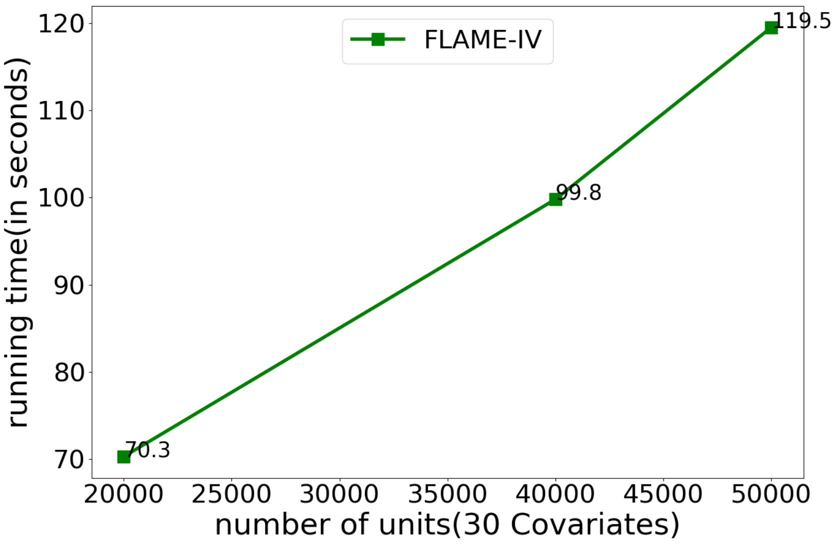

Figure 8.1: Running Time for FLAME-IV on large dataset.

b_u *and* b_u^+ *differ iff the same combination of covariate values appear both as an instrumented unit and an uninstrumented unit.*

An example of this procedure is illustrated in Table 8.1. We assume in this population the 0-th variable is binary and the next variable is ternary. In this example, the number b_1 for the first unit is $0 \times 2^0 + 2 \times 3^1 = 6$; the number b_1^+ including its treatment indicator is $0 + 0 \times 2^1 + 2 \times 3^2 = 18$. Similarly we can compute all the numbers b_u, b_u^+, c_u, c_u^+, and the matching results are listed in the last column in Table 8.1. subsectionMore Running Time Results on Large Dataset Figure 8.1 shows the results of running time for FLAME-IV on a larger dataset. The running time is still very short(< 2 min) on the large dataset for FLAME-IV. Full matching can not handle a dataset of this size.

Table 8.2: Two sample matched groups generated by FLAME on the application data described in Section 8.9.

Territory	Last Election PS Vote Share	Last Election Turnout	Population (in thousands)	Share Male	Share Unemployed	Treated	Instrumented
Matched Group 1							
Plouguenast et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	1
Lorrez-le-Bocage-Préaux et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	1
La Ferté-Macé et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	1
Mundolsheim et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	1	1
Paris, 7e arrondissement	(0.01, 0.05]	(0.77, 0.88]	(1,800, 2,250]	(0.47, 0.57]	(0.1, 0.2]	0	1
Sainte-Geneviève et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	0
Cranves-Sales et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	0
Hem et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	1
Legé et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	1
Moûtiers et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	0
Paris, 7e arrondissement	(0.01, 0.05]	(0.77, 0.88]	(1,800, 2,250]	(0.47, 0.57]	(0.1, 0.2]	0	1
Craponne-sur-Arzon et environs	(0.01, 0.05]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0, 0.1]	0	0
Matched Group 2							
Nantes	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.47, 0.57]	(0.1, 0.2]	1	1
Alès	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.37, 0.47]	(0.2, 0.3]	1	1
Sin-le-Noble	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.47, 0.57]	(0.2, 0.3]	1	1
Grand-Couronne et environs	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.47, 0.57]	(0.1, 0.2]	1	1
Dreux	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.47, 0.57]	(0.2, 0.3]	1	1
Vosges	(0.19, 0.22]	(0.77, 0.88]	(0, 450]	(0.47, 0.57]	(0.1, 0.2]	0	0
Arras et environs	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.37, 0.47]	(0.1, 0.2]	1	1
Montargis et environs	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.37, 0.47]	(0.2, 0.3]	1	1
Marseille, 3e arrondissement	(0.19, 0.22]	(0.66, 0.77]	(450, 900]	(0.47, 0.57]	(0.1, 0.2]	1	1
Nantes	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.47, 0.57]	(0.1, 0.2]	1	1
Mâcon et environs	(0.19, 0.22]	(0.66, 0.77]	(0, 450]	(0.37, 0.47]	(0.1, 0.2]	1	1

Notes: The columns are a subset of the covariates used for matching.
Territory was not used for matching. Original covariates are continuous and were coarsened into 5 bins. Last election PS vote share was coarsened into 10 bins. Labels in the cells represent lower and upper bounds of the covariate bin each unit belongs to.
The two groups have relatively good match quality overall.

8.9 Sample Matched Groups

Sample matched groups are given in Table 8.2. These groups were produced by FLAME-IVon the data from [Pon18], introduced in Section 8.9. The algorithm was ran on all of the covariates collected in the original study except for territory. Here we report some selected covariates for the groups. The first group is comprised of electoral districts in which previous turnout was relatively good but PS vote share was low. This suggest that existing partisan splits are being taken into account by FLAME-IVfor matching. Municipalities in the second group have slightly lower turnout at the previous election but a much larger vote share for PS. Note also that treatment adoption is very high in the second group, while low in the first: this suggest that the instrument is weak in Group 1 and strong in Group 2.