# Why Propensity Scores Should Not Be Used for Matching: Supplementary Appendix

Gary King[*] Richard Nielsen[†]

January 17, 2019

**Abstract**

This paper is the Supplementary Appendix to Gary King and Richard Nielsen, "Why Propensity Scores Should Not Be Used For Matching," copy at j.mp/psnot

[*] Institute for Quantitative Social Science, Harvard University, 1737 Cambridge Street, Cambridge MA 02138; GaryKing.org, king@harvard.edu, (617) 500-7570.

[†] Department of Political Science, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139; mit.edu/∼rnielsen, rnielsen@mit.edu, (857) 998-8039.

# 1 PSM Approximates Random Matching

In a simple simulation, we provide intuition for how relatively balanced data makes PSM, but not MDM or CEM, highly sensitive to trivial changes in the covariates, often producing nonsensical results that approximate random matching. In the left panel of Figure 1, we generate data with 12 observations and two covariates, with one covariate plotted by the other. The data are well balanced between treated (black disks) and control (open circles) units. From these initial data, we generate 10 data sets, where we add to each of the 12 observations a small amount of random error drawn from a normal distribution with mean zero and variance 0.001. This error is so small relative to the scale of the covariates that the new points are visually indistinguishable from the original points (in fact, the graph plots all 10 sets of 12 points nearly on top of one another, but it only appears that one set is there). Next, we run CEM and MDM; in both cases, as we would expect, the treated units are matched to the nearest control in every one of the 10 data sets (as portrayed by the pair of points linked by curved solid lines).

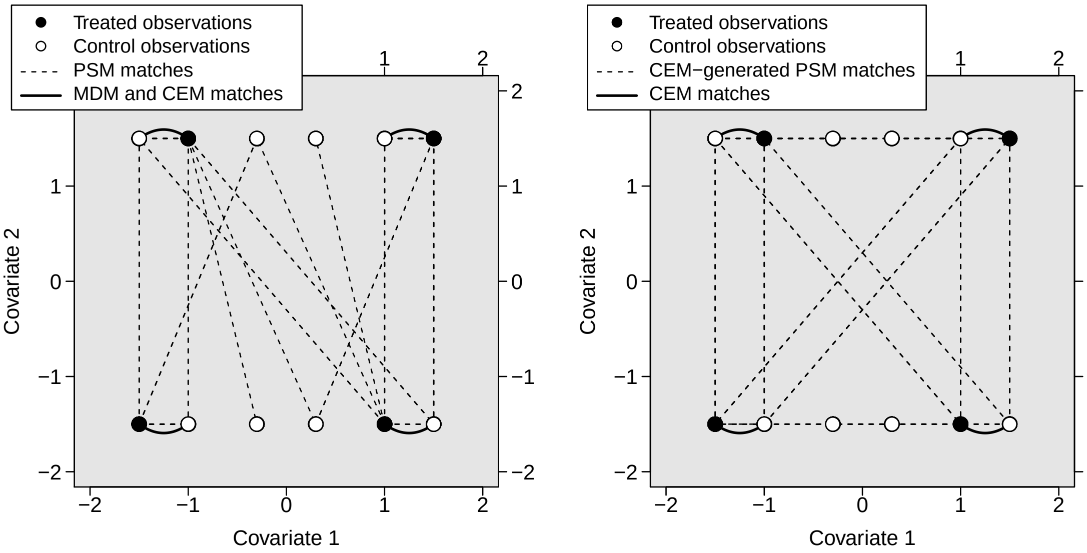

Figure 1: Ten data sets (differing from each other by imperceptibly small amounts of random error) with 4 treated units (black disks) and 8 control units (open circles). CEM and MDM match the closest control units to each treated (curved black lines). The two-step procedures match different control units for each data set, as can be seen for PSM (dashed lines, left panel) and PS-CEM (dashed lines, right panel). (The four open circles in the middle of the right panel are never matched; lines are passing through them on the way to show how other points are matched.)

However, when we run PSM on each of the 10 data sets generated for Figure 1, the four treated units are each matched to *different* control units (as portrayed by the maze of dashed lines connecting the black disks to different open circles). PSM is approximating random matching in this situation because it is unable to distinguish treated and control units; it is blind to the space of $X$ that is not represented in $\hat{\pi}$.

Perhaps the problem is our estimation of the propensity scores? It is possible that fitting a logistic regression to twelve data points results in poorly estimated propensity scores because of finite sample bias in maximum likelihood estimators. We do not generally advocate logistic regression for twelve observations, and we only use such a small sample here for clarity in the simulation. However, the estimates of the propensity scores are not the problem. By construction, the we know the propensity scores are $0.\overline{3}$. The estimated propensity scores across all 10 simulations (120 observations) range from 0.332899 to 0.333768, so the estimation is good. Moreover, we obtain the same result if we replace the estimated propensity scores with the known propensity scores. The problem with propensity scores in this example is not about estimation.

Finally, we illustrate how the paradox results from PSM's two-step procedure. We do this by developing a (similarly unnecessary and ill-advised) two-step "propensity score CEM" (PS-CEM) algorithm: to do this, we use CEM to compute a nonparametric estimate of the propensity score (i.e., the proportion of treated units within each coarsened stratum; see Iacus, King, and Porro 2011) and, second, without running CEM as usual, we match directly on the nonparametric estimate of the propensity score. The right panel in Figure 1 is constructed the same way as the left panel except that instead of the dashed lines representing propensity score matches, they represent PS-CEM matches. The result is almost as bad as PSM. The dashed lines in the right panel show how in the different (but highly similar) data sets, the two-step PS-CEM procedure matches control units (circles) close to and also distant from treated (closed disks) units. This suggests that ignoring $X$ and only matching based on the scalar propensity score generates the PSM paradox.

# 2 PSM Extensions Also Ignore Information

Most of our analysis has focused on the simplest version of PSM, which could be called *greedy nearest neighbor* matching. Our content analysis in Section 3.2 shows that the vast majority of applied papers (94%) use this simple version of PSM, but numerous extensions to PSM have been proposed in the methodological literature. We show here that these extensions to PSM do not avoid the problems we have identified. Of course, it is unsurprising that methods that seek to build on the PSM framework inherit the basic properties of and problems with PSM, even though they clearly each accomplish the more specific goals they set out to solve.

To do this, we replicate the simulation in Section 4 with six additional approaches. We first describe each briefly with citations for readers interested in further details. Two of these are adjustments to the matching procedure that introduce no new information about the covariates beyond what is contained in the propensity score:

1. **Optimal Matching** (Rosenbaum, 1989) offers an alternative to the greedy matching without replacement of the simplest version of PSM. Greedy matching without replacement matches control units to treated in order of availability, potentially resulting in poor matches for the some treated units that could have been better if considered in a different order. Optimal matching uses a network flow algorithm to construct a matched sample that minimizes the average distance between all matched pairs simultaneously.

2. **Optimal Full Matching** (Hansen, 2004; Rosenbaum, 1989) extends optimal matching from one-to-one to one-to-many groupings of treated and control units.

Three other methods incorporate additional information about the covariates along with the propensity score.

3. The **Covariate Balancing Propensity Score** (Imai and Ratkovic, 2014) approach estimates propensity scores while simultaneously optimizing covariate balance. Poorly estimated propensity scores can lead to bias. The covariate balancing propensity

score offer some robustness to poorly estimated propensity scores by using moment conditions to estimate propensity scores that are good balancing scores.

4. **Genetic Matching with the Propensity Score**. Genetic matching (Diamond and Sekhon, 2012) is a generalization of Mahalanobis distance matching that uses a genetic algorithm to estimate optimal weights for each covariate to maximize balance in the matched sample (by maximizing the p-values from paired t-tests and Kolmogorov-Smirnov tests of the covariates). When the propensity score is included as a matching covariate and receives positive weight, genetic matching is a generalization of propensity score matching as well.

5. **Mahalanobis Distance Matching with the Propensity Score as a Matching Variable**. We estimate the propensity score as normal and include it as an additional matching covariate in greedy nearest neighbor Mahalanobis distance matching. This is similar to genetic matching with the propensity score, but the contribution of the propensity score to MDM is not optimally weighted.

Finally we also include one modification of MDM with no propensity score.

6. **Genetic Matching** (ibid.) without the propensity score is a generalization of Mahalanobis distance matching that optimally weights the covariates to maximize balance (maximizing the p-values from paired t-tests and Kolmogorov-Smirnov tests of the covariates).

We also consider **PSM with a bias adjustment** proposed by Abadie and Imbens (2011), but this is an adjustment to the estimation after matching, rather than to the matching procedure itself. We suppress this from the figure because it would reveal no change at all. In any case, PSM with a bias adjustment does not do any better than simple PSM at avoiding the PSM paradox.

We use the same simulation set-up described more fully in Section 4: We simulate 1,000 data sets, each of with data from three separate sources: (1) a matched pair randomized experiment, (2) a completely randomized experiment, (3) observations that, when

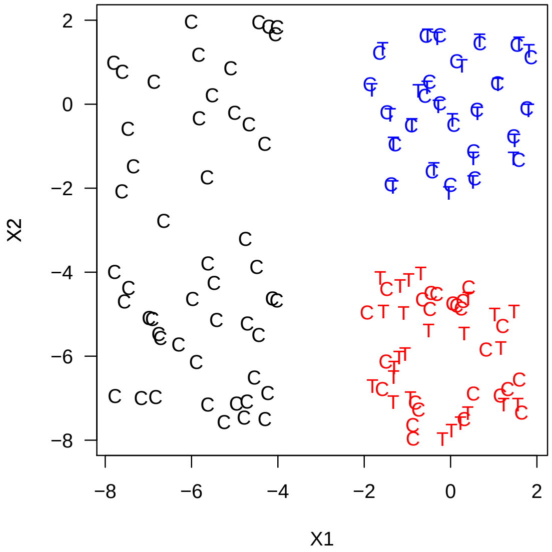

Figure 2: One (of 1,000) randomly generated data sets from a matched pair randomized experiment (in blue), a completely randomized experiment (in red), and control units from an imbalanced observational data set (in black).

added to the first two components, make the entire collection an imbalanced observational data set. Figure 2 shows one of these 1,000 data sets, with the matched pair randomized data in blue, the completely randomized data in red, and the imbalanced controls in black. Each row of pixels in the figure is a separate simulation.

We then study whether each method prunes individual observations in the correct order: starting with those at the highest level of imbalance (data set 3) to the lowest (data set 1). Ideally black is removed first, then red, then blue.

The first two panels of Figure 3 repeat the results for Mahalanobis Distance Matching and Propensity Score Matching reported in the main text. We calculate one additional statistic to aid comparison with the other variants: the proportion of observations from data set 1 that were incorrectly pruned before all observations from data sets 2 and 3 were pruned. For MDM, this proportion is 0.913 with a bootstrapped 95% confidence interval of [0.911, 0.916]. That is, approximately 8.7% of observations in data set 1 were removed too early. In comparison, the proportion of data set 1 observations correctly retained by PSM was much lower: 0.576 [0.573, 0.580]. PSM does slightly better than random guessing, but not much. Figure 4 shows these proportions graphically.

The results for the six alternative matching algorithms described above are shown in

Figures 3 and 4. Adaptations of the PSM algorithm to allow optimal matching or optimal full matching produce are no less blind to the difference between completely randomized and fully blocked data than simple PSM (see Figures 3 and 4). The covariate balancing propensity score method also offers no noticeable improvement. Matching methods based on MDM perform much better, though incorporating the propensity score hurts performance marginally, roughly in proportion to how much weight is put on the propensity score. That is, if we had 100 covariates and also the propensity score, the propensity score would not hurt as much as if there were only 2 covariates. This is as expected because the problems we identify occur with PSM and other methods that try to funnel all information about matching through the propensity score.

These results suggest that the problems we identify with PSM are not limited to the simplest cases. The blindness that PSM has to the great advantage of fully blocked data over completely randomized data is not changed even for methods that attempt to fix other, unrelated problems with PSM.

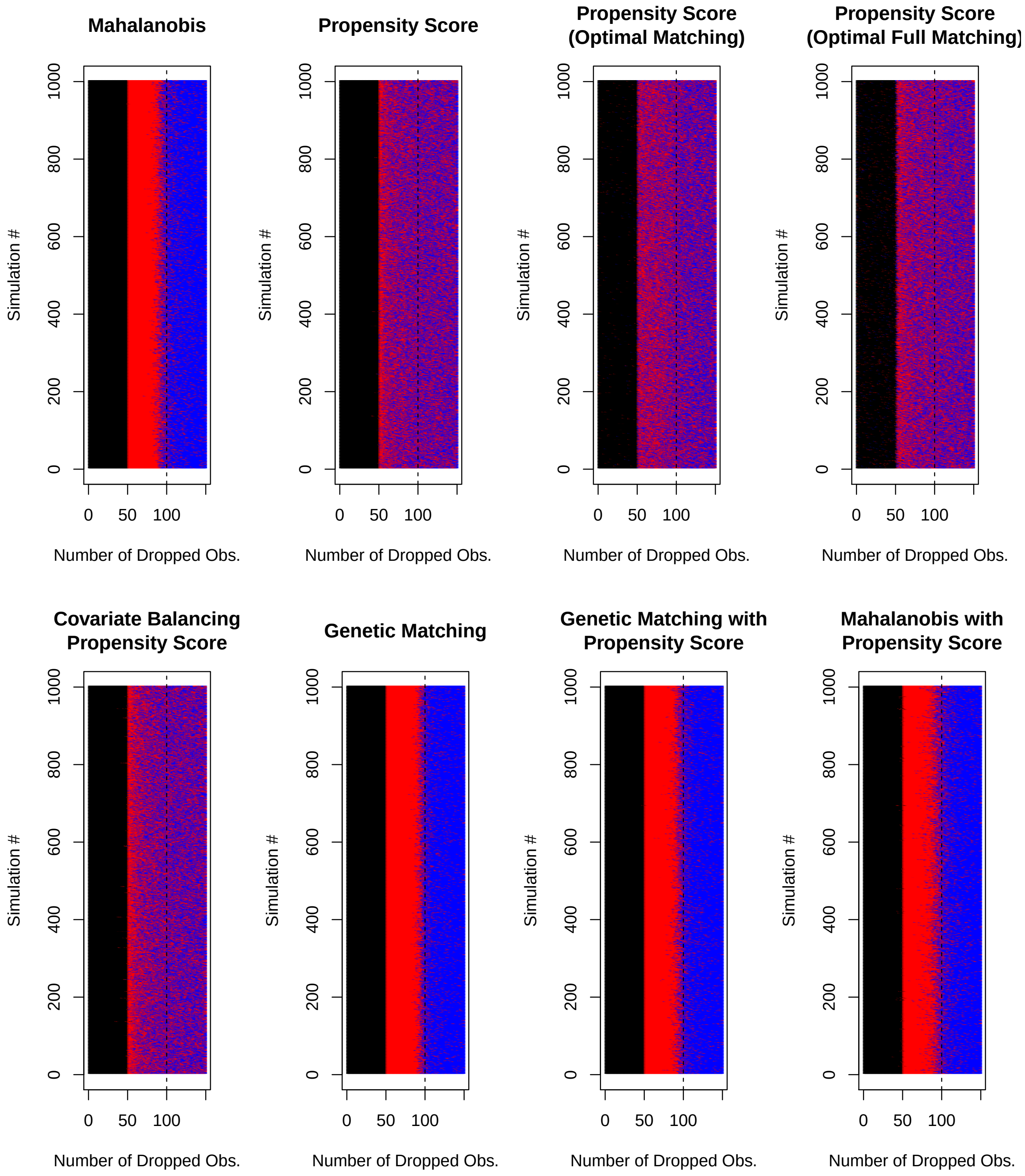

Figure 3: Variants of PSM, that build on the method, cannot find fully blocked experiments hidden in observational data. In each panel, each of 1,000 simulations is represented by a separate row of pixels, color-coded by experiment type to indicate the order (from left to right) in which observations are pruned. Ideally, black is removed first, then red, then blue. Mahalanobis, Mahalanobis with Propensity Scores, Genetic Matching, and Genetic Matching with Propensity scores generally suceed. Propensity Scores, Optimal Matching with Propensity Scores, Optimal Full matching with Propensity Scores, and the Covariate Balancing Propensity Score fail, exactly as predicted depending on how much the method depends on PSM.

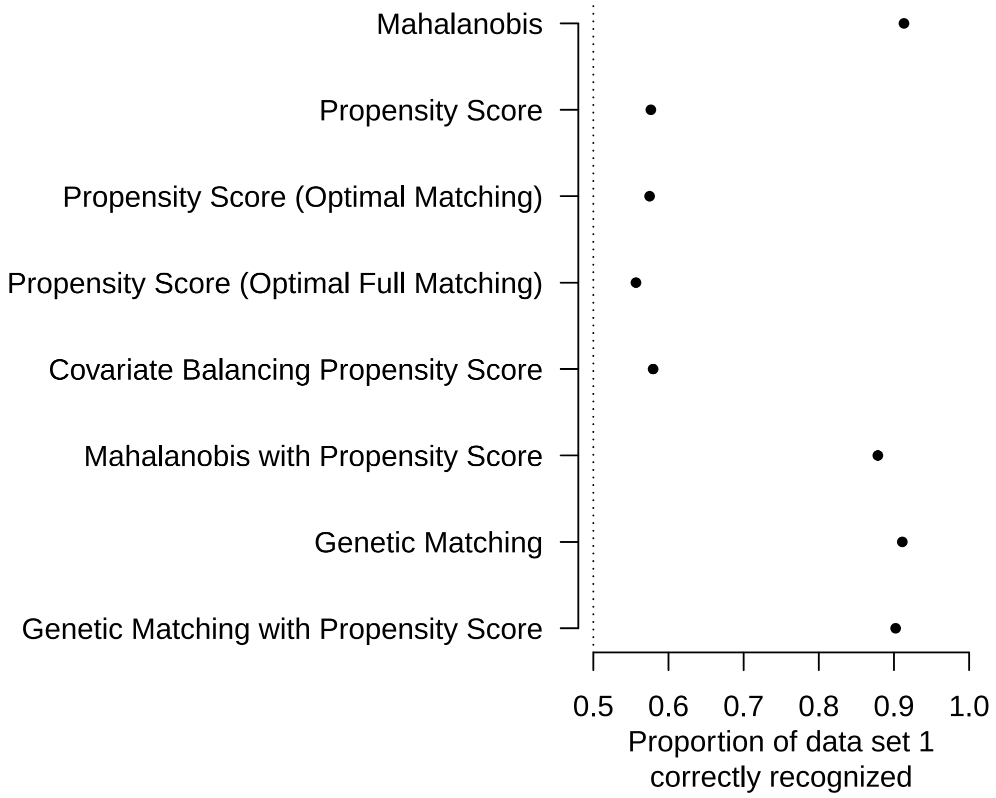

Figure 4: Comparing the performance of eight matching methods for finding fully blocked experiments hidden in observational data. The x-axis shows the average proportion of high-quality block-pair matches discarded before all non-block-pair matches are discarded by each method across 1000 simulations. Bootstrapped confidence intervals are shown, but are so small that they are difficult to see.

# 3 Damage Caused In Data Generated to Fit PSM Theory

We now study different types of simulations generated consistent with PSM theory, varying the number of covariates, levels of imbalance, and matching method.

## 3.1 Data Generation Processes

We generate data by following Gu and Rosenbaum (1993). Covariates are drawn from a multivariate normal (meeting the data requirements of EPBR) with variances of 1 and covariances of 0.2. Data sets with high, medium, and low levels of balance result from setting the control group mean vector to (0,0,0) and different treated group mean vectors to (0.1,0.1,0.1), (1,1,1), and (2,2,2), respectively. We draw 250 treated and 250 control units, which is equivalent for our purposes to generating a larger pool of controls and pruning down to 250 to achieve 1-to-1 matching with the treated units. We then prune from that point by calipering off additional units.

We draw 50 random data sets, for each of the nine combinations of 1, 2, and 3 covariates and low, medium, and high levels of imbalance. (Analyses with more covariates and higher levels of imbalance predictably produce even more dramatic patterns than presented here and so we do not present them.) For each data set generated, we analyze the same data with PSM, CEM, and MDM, following the procedures from Section 5.3. We repeat the procedure for each level of pruning and for each of the 50 data sets, average, and put a point on a graph we present below. All our results apply to estimating both SATT and FSATT; for simplicity, we present results here for the latter, which is the most commonly recommended and used approach.

For the third data generation process, we use the same simulated covariates as above, but define the treatment assignment vector using a true propensity score equation, and during estimation use the knowledge of the correct specification. We find nearly identical results for all three data generation processes, and also when rerunning them with a wide range of different parameter values; as such we only present results from the first process.

## 3.2 Results

Figure 5 gives the results for the methods in three rows (PSM, MDM, and CEM from top to bottom) and different numbers of covariates in separate columns (1,2,3, from left to right). Each of the nine graphs in the figure gives results from data generated with low (dotted line), medium (dashed line), and high (solid line) levels of initial imbalance. For graphical clarity, individual matching solutions do not appear and instead we average over the 50 simulations in each graph for a given matched sample size and level of imbalance. The PSM paradox is revealed whenever one of the lines increase from left to right (i.e., with imbalance increasing as the number of observations drops).

As expected, the usual curse of dimensionality reduces the performance of all three matching methods, as can be seen by the level of imbalance increasing from graphs at the left to the graphs at the right in any one row. Also as expected, the second and third rows of the Figure 5 show that CEM and MDM do not suffer from the paradox in these data: for all the lines in all the graphs in these rows, imbalance never increases as more observations are pruned, just as we would want to be the case.

However, for PSM in the first row, three important patterns emerge, all of which occur when the propensity score paradox kicks in (where a line changes direction from heading downward to where it starts heading upwards). First, no paradox emerges with PSM and one covariate (top left graph) because PSM does no dimension reduction; in this case, the propensity score is merely a rescaling of a scalar $X$. Second, the paradox point appears earlier, that is with fewer observations pruned, the more covariates are included in the propensity score regression (as we go from left to right in the top row of Figure 5). This problem is worse for 3 than 2 covariates and, although we do not show it, the paradox intensifies with more covariates. Third, the paradox kicks in earlier as the data become more balanced, approximating a completely randomized experiment. This can be seen by comparing the dotted (well balanced) and solid (least balanced) data in the two graphs at the top right, and noting that the point where the paradox starts moves to the left for better balanced data.

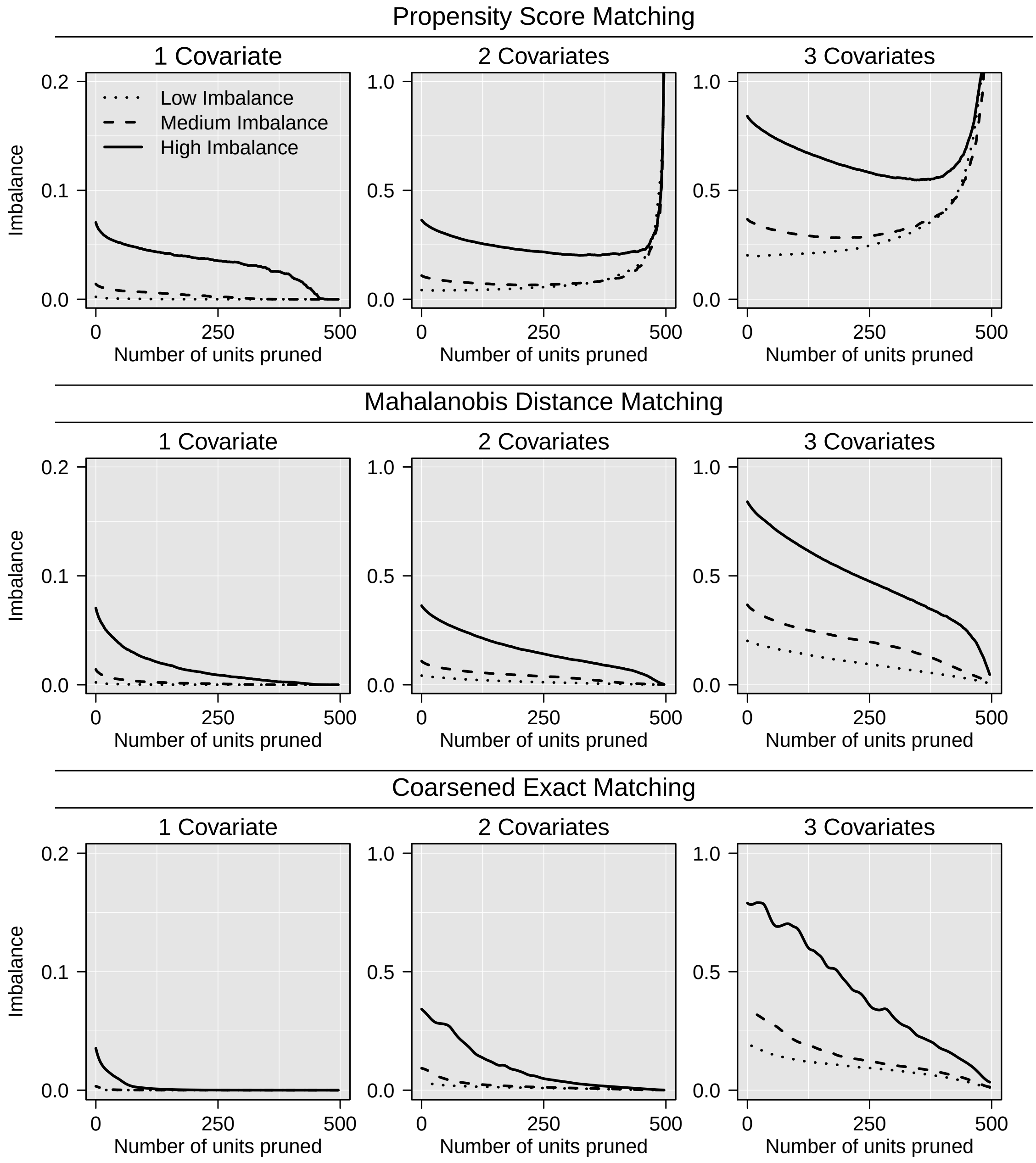

Figure 5: For PSM, MDM, and CEM in rows, and 1, 2, and 3 covariates in columns, these graphs give average values from 50 simulations with low (dotted), medium (dashed), and high (solid) levels of initial imbalance between the treated and control groups. The paradox is revealed for portions of lines that systematically increase. This can be seen for PSM with more than one covariate but not for CEM and MDM.

# 4 The Paradox With Other Methods

In the first simulation, we contrive a data set where nature is malicious. We begin by generating 100 values of a single covariate $X$ deterministically, in pairs along the number line, skipping every third value, as $X = 1, 2, 4, 5, 7, 8, \ldots, 145, 146, 148, 149$. We then assign observations with even values of $X$ to receive treatment and those with odd values to receive control. If we stopped here, each treated unit would match best to the control observation 1 unit away and $T$ would be independent of $X$ in sample (and where both treated and control units of $X$ have a mean of 75). Then to each value of $X$, we add a tiny amount of jitter drawn from a uniform on the interval $[-0.00001, 0.00001]$. This results in some pairs being slightly better matches than others, although solely due to random jitter. We then introduce confounding (which can be productively fixed via matching) by taking the three treated units with the lowest values of $X$ and reassigning them to control, and taking the three control units with the highest $X$ values and assign them to treatment. For example, this creates a substantial difference between the mean value of $X$ for the treated (≈83) and control (≈67). We generate the outcome variable as $Y = T + 0.01X + \epsilon$, where $\epsilon \sim N(0, 1)$.

The resulting data set has important levels of imbalance (and confounding) due to the units at the low and high values of $X$. The rest of the data will have matches that are effectively at random. The idea is that any method of matching will first prune the extreme (imbalanced) observations first for good reason and then start pruning at random.

We measure model dependence by first estimating the regression of $Y$ on a constant, $T$, and elements of one of the subsets of $\{X, X^2, X^3, X^4, X^5\}$, and then repeat for all the other subsets. Then our measure is the range of estimates of the coefficient on $T$ across all these regressions. Results appear in Figure 6, with model dependence plotted vertically and the number of treated units pruned by MDM horizontally.

Thus, MDM first prunes the six extreme values of $X$ which causes model dependence to drop. After that point, when all pairs differ by pure randomness, MDM continues to prune without accomplishing anything of value. Matching in this way does not overcome the fact that pruning itself increases imbalance, and so the overall imbalance line starts

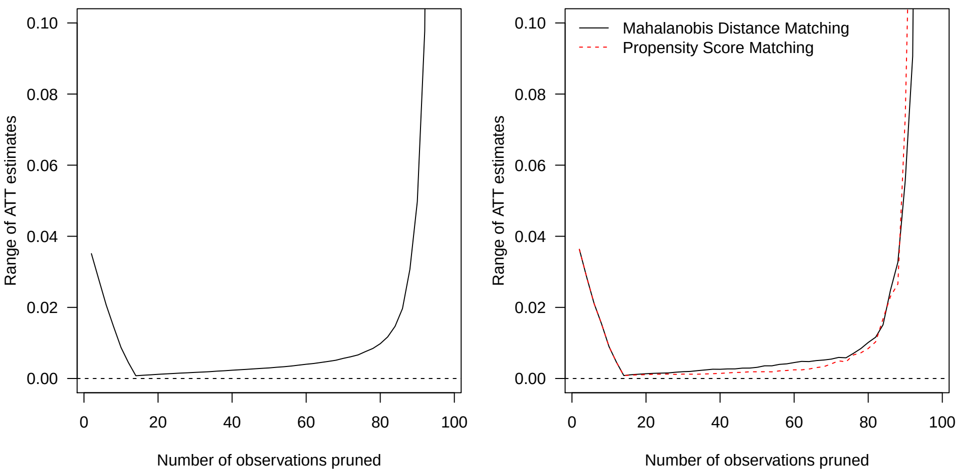

Figure 6: Left: The paradox with Mahalanobis Distance Matching. Right: Propensity Score Matching can outperform Mahalanobis Distance Matching, but only when matching is doing damage anyway.

heading upward.

We also tried to modify this simulation to create a situation where PSM outperforms MDM. However, even in this highly artificial data set, we were only able to induce better performance from PSM relative to MDM *when pruning at all was doing damage to the data set*. To do this, we increased the radius of random jitter around $X$ from $[-0.00001, 0.00001]$ to $[-0.01, 0.01]$. As the jitter increases, PSM performs "better" — or really in this situation less worse — because it prunes at random, while MDM matches the observations with jittered $X$ values that happen to be close first. However, MDM and PSM perform equally well at removing the six observations with extreme values of $X$. It is only once these six observations are removed that PSM outperforms MDM, but both methods are producing increasingly model dependent data sets at this point.

In more than two dozen real data sets and thousands of simulations we designed for this purpose, we have not seen PSM "outperform" MDM while also reducing imbalance. To be clear, we do not have a mathematical proof with such an impossibility theorem, but if it is possible it seems exceedingly unlikely in practice.

For a second illustration, we create a very small data set in high dimensional space so that points are so far spread out that few good matches are available. This is easy to see in MDM since Mahalanobis distances in this situation have the characteristic property of differing by tiny, essentially random amounts, only after many digits to the right of the decimal point. Thus, we generate a small data set, $n = 200$, with $k$ covariates, for $k = 2, 3, 4, 5, 10$. For each $k$, we generate 100 data sets with covariates drawn from independent standard normals with means drawn from a uniform on the interval $[-10, 10]$. Then, for units designated as control, we add an independent draw for each covariate from a normal with mean zero and standard deviation 5.

We then define a set $\mathcal{M}$ of linear regression models that includes all possible specifications that include subsets of covariates, squared terms, and interactions, with squared terms and interactions included only if the main effects are included. We draw one model from $\mathcal{M}$ to define the true data generating process. We use this one true model to generate $Y$ as a linear function of the treatment times its effect of 100, the covariates with coefficients drawn from a uniform distribution on the interval of [0,500], a constant term of 500, and a normal error term with mean 0 and standard deviation 500.

For each of the 100 data sets and each sample size, we run PSM and MDM, using all main effects only. To compute model dependence for a (matched) data set, we draw 1,000 models from $\mathcal{M}$, estimate the treatment effect for each as the coefficient on the treatment variable, and then compute the variance across these estimates. In order to have a comparable measure, the subset of 1,000 models is fixed across all runs (within a fixed $k$). We then average the standardized estimates of model dependence within each run, over the 100 runs, and plot scaled estimates.

Figure 7 gives our results, in parallel to previous figures, so that number of units pruned is on the horizontal axis and model dependence on the vertical axis. With PSM in red and MDM in blue, one panel appears for each $k$. Four patterns are apparent. First, PSM has higher levels of model dependence than MDM throughout all five graphs. Second, the advantage of MDM over PSM increases in all five graphs as more observations are pruned. Third, the PSM paradox is evident in all five graphs. And finally, a paradox, with more units pruned leading to higher levels of imbalance, also affects MDM in 10 dimensional space in the last graph (and to some small extent right at the end of the some of the others).

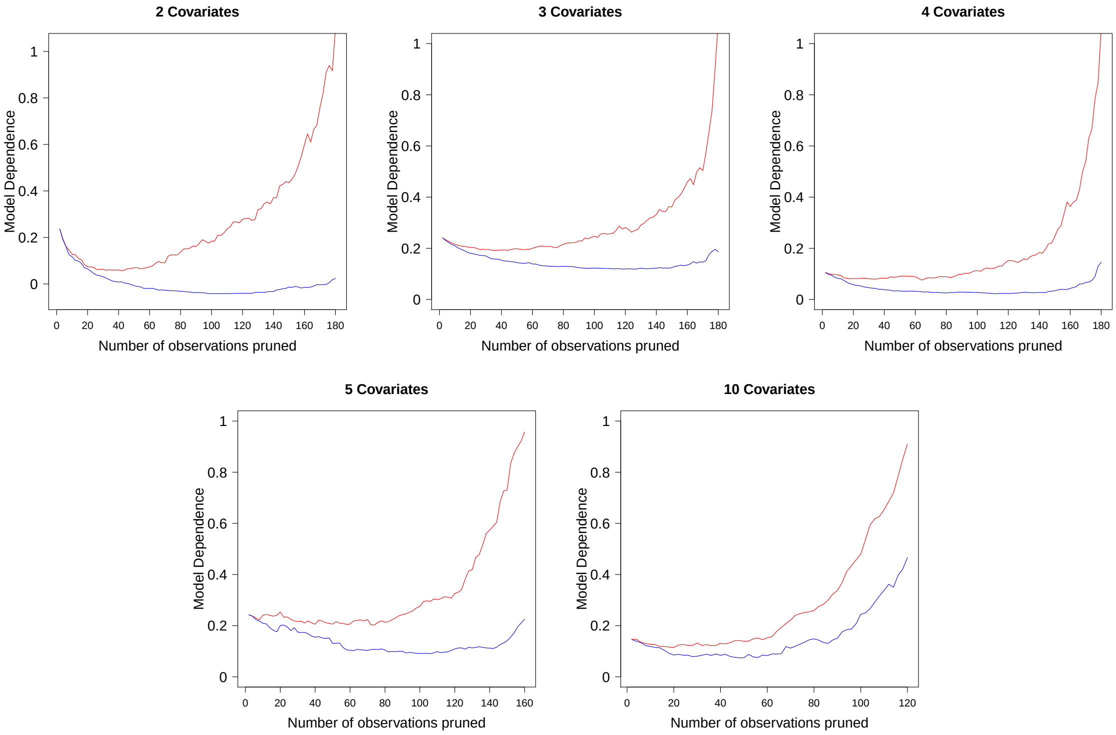

Figure 7: Model Dependence by Number of Covariates, with PSM in red and MDM in blue.

# References

Abadie, Alberto and Guido W. Imbens (2011): "Bias-corrected matching estimators for average treatment effects". In: *Journal of Business & Economic Statistics*, no. 1, vol. 29.

Diamond, Alexis and Jasjeet S Sekhon (2012): "Genetic matching for estimating causal effects: a general multivariate matching method for achieving balance in observational studies". In: *Review of Economics and Statistics*, no. 3, vol. 95, pp. 932–945.

Gu, X.S. and Paul R. Rosenbaum (1993): "Comparison of multivariate matching methods: structures, distances, and algorithms". In: *Journal of Computational and Graphical Statistics*, vol. 2, pp. 405–420.

Hansen, Ben B. (2004): "Full Matching in an Observational Study of Coaching for the SAT". In: *Journal of the American Statistical Association*, no. 467, vol. 99, pp. 609–618.

Iacus, Stefano M., Gary King, and Giuseppe Porro (2011): "Multivariate Matching Methods that are Monotonic Imbalance Bounding". In: *Journal of the American Statistical Association*, vol. 106, pp. 345–361. URL: `j.mp/matchMIB`.

Imai, Kosuke and Marc Ratkovic (2014): "Covariate balancing propensity score". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, no. 1, vol. 76, pp. 243–263.

Rosenbaum, Paul R. (1989): "Optimal matching for observational studies". In: *Journal of the American Statistical Association*, vol. 84, 1024–1032.