Counterbalance Matching First Thoughts

JP & Dylan Small

2025-06-22

Literature Review

Paul Rosenbaum lays out a method for "thick description" in his 2001 paper, which details a method for enriching our understanding of matched data by including a richer, narrative-like description of individual subjects. Specifically, he proposes investigating particularly-close matches to better-understand variables we match on, allowing us to clarify (or omit) misunderstood, mis-defined, or otherwise problematic variables.

Then in Yu et. al's 2021 paper, an optimal matching algorithm for thick description is proposed: specifically, match in a globally-optimal way with the **constraint** that we generate F closest matches, say $F = 30$. From there, we can generate a "thick description" of each subject in the F closest matches, and use that to understand possible nuances in the covariates.

However, there may be even more information gained through thick description considering counterbalanced matches, or matched pairs that are intentionally dissimmilar in the covariates. This is particularly interesting in the case where a low-propensity individual receives the treatment and a high-propensity individual does not! We explore this notion with our method.

Optimal Counterbalance Matching for Thick Description

We propose a method for counterbalancing matches, where we generate F counterbalanced matches, or matched pairs that are intentionally dissimmilar in the covariates. We set up the general form of the method, then define our algorithms and possible implementations.

Problem Setup

In dataset $\mathcal{D}$, we have the following variables:

- i: index of the i^{th} subject
- x_i: vector of covariates for the i^{th} subject
- y_i: outcome for the i^{th} subject
- z_i: treatment status for the i^{th} subject ($z_i = 1$ if the i^{th} subject receives the treatment, $z_i = 0$ if control)

Let T represent the set of the treated indices, and C represent the set of the control indices. Symbolically,

$$T = \{i \in \mathcal{D} : z_i = 1\}$$
$$C = \{j \in \mathcal{D} : z_j = 0\}$$

Note that $T \cup C = \mathcal{D}$ and $T \cap C = \emptyset$.

We then calculate some measure of similarity between subjects i and j based on their pre-treatment covariates (x_i, x_j). We define these scores as δ_{ij} for $i \in T, j \in C$. There are $|T| \times |C|$ pairs of subjects, and so we have a matrix of similarity scores

$$\boldsymbol{\delta} \in \mathbb{R}_{\geq 0}^{|T| \times |C|}$$

where $\delta_{ij} = \delta(i,j)$. Then analogous to how Yu et. al sought to find the F **closest** pairs of subjects, we set out to find the F **furthest** pairs of subjects, where we define "furthest" as the pair with the **lowest** similarity score. Then using the order statistics of δ_{ij}'s $\in \boldsymbol{\delta}$, we identify the F **furthest** pairs of subjects, then match the remaining subjects in a globally-optimal way. We formally define these algorithms next.

Algorithm

Finding the Similarity Score Threshold

First, we must identify a similarity score threshold κ such that the number of counterbalanced matches (above the threshold) is F. We define that here:

1. Calculate similarity scores δ_{ij} for $i \in T, j \in C$
2. Set the number of desired counterbalanced matches F
3. Make a guess κ as a guess of the true similarity score threshold Δ_F above which are F counterbalanced matches
4. Set $\delta'_{ij} = 1$ if $\delta_{ij} \leq \kappa$, and $\delta'_{ij} = 0$ if $\delta_{ij} > \kappa$.
5. Find an optimal matching which minimizes $\sum_{i,j} \delta'_{ij}$, call this minimum F'
6. If $F' < F$, then our proposed threshold κ is too high, and we repeat steps 4 - 6 with a greater κ. If $F' > F$, then our proposed threshold κ is too low, and we repeat steps 4 - 6 with a lower κ.
7. Repeat steps 4 - 6 until $F' = F$

Finding the Matches

Now, we are ready to define the algorithm for optimally matching the subjects. First we define the following additional variables:

- κ: the similarity score threshold we found in the previous section
- β: a massive penalization term defined as $\beta = 50 \times \max_{i \in T, j \in C} \delta_{ij}$
- δ''_{ij}: new scores defined as

$$\delta''_{ij} = \begin{cases} \delta_{ij} & \text{if } \delta_{ij} > \kappa \\ \delta_{ij} + \beta & \text{if } \delta_{ij} \leq \kappa \end{cases}$$

  This new score penalizes massively the scores below the threshold κ, ensuring that the F counterbalanced matches are paired.

We then find an optimal matching which minimizes $\sum_{i,j} \delta''_{ij}$. We will now move on to possible implementations of this algorithm.

Implementation 1: Using the Mahalanobis Distance

We can use the Mahalanobis distance to calculate the similarity scores δ_{ij}. The Mahalanobis distance for pair (i,j) is defined as

$$\delta_{ij} = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)}$$

where Σ is the covariance matrix of the covariates. We generally estimate this using the sample covariance matrix defined as

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

where $\bar{x}$ is a vector of the sample means of the covariates.

When we implement our matching algorithm, we will get F counterbalanced matches with very different covariates. This can be useful for understanding the nuances of covariates and how they relate to the likelihood of receiving treatment.

Implementation 2: Using Propensity Scores

We calculate propensity scores p_i for each subject i as the conditional probability of receiving treatment $T_i = 1$ given their covariates x_i. We typically estimate this using a logistic regression model.

Once we have these propensity scores, we can calculate the similarity scores δ_{ij} as the absolute difference in propensity scores between subjects i and j.

$$\delta_{ij} = |p_i - p_j|$$

Then, once we implement our matching algorithm, we will get F counterbalanced matches with very different propensity scores. This functions very similarly to the Mahalanobis distance implementation, but may be more interpretable in higher-dimensional settings.

The Problem with Using Only Propensity Scores

Note: if we implement our method with propensity scores, it is **VERY LIKELY** that we will find counterbalanced matches where the treated subject has a much higher propensity to receive treatment than the control subject. This makes sense, but it may not provide us with the most interesting or potentially-insightful matches for thick description. To address this, we will create a new metric and define our algorithm slightly differently.

Optimal Counterbalance Matching Using Surprise Scores

Defining the Surprise Score

Let $p_i \in [0, 1]$ be the propensity score for subject i. We want to define a new metric, a **surprise score** s_i for subject i as how **surprising** it is that subject i received a particular treatment outcome (ex: treatment or control).

For example, if subject i has a propensity score of 0.1 but received treatment ($z_i = 1$), we want our surprise score to be high. The opposite is also true: if subject i has a propensity score of 0.9 but received control ($z_i = 0$), we want our surprise score to be high as well. We'd also like these scores to be on a similar scale to the propensity scores, and so we define our surprise score s_i as

$$s_i = |p_i - z_i|$$

or the absolute difference between the propensity score and the treatment outcome. Note that this is a valid metric, as it does not depend on the observed outcome y_i. We now modify our algorithm to accomodate the use of these scores.

Algorithm

Our modified algorithm is as follows:

1. Calculate the propensity score p_i for each subject i as the conditional probability of receiving treatment $T_i = 1$ given their covariates x_i.

$$p_i = \mathbb{P}(T_i = 1 | x_i)$$

2. Calculate similarity scores δ_{ij} for $i \in T, j \in C$ using the Mahalanobis distance from Implementation 1.

$$\delta_{ij} = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)}$$

where Σ is the covariance matrix of the covariates.

3. Calculate the surprise score s_i for each subject i as the absolute difference between the propensity score and the treatment outcome
$$s_i = |p_i - z_i|$$

4. Sort the subjects by their surprise scores s_i and select the top 50%. Call this set S, defined formallly as
$$S = \{i \in \mathcal{D} : s_i \geq s_{(\lceil |\mathcal{D}|/2 \rceil)}\}$$
where $|\mathcal{D}|$ is the number of subjects in the dataset and $s_{(i)}$ is the i^{th} order statistic of the surprise scores. Note that this extracts the top 50% of subjects by surprise score.

5. Set the number of desired close matches (via Yu et. al) or counterbalanced matches (via our method) F, depending on the insight we want to gain.

6. Depending on the method selected, find these F matches from S, then match the remaining subjects in a globally-optimal way.

   This algorithm ensures that we find F close (or counterbalanced) matches who also have high surprise scores, allowing us to understand how certain covariates relate to the likelihood of receiving treatment.