Acknowledgment

Funds: This work was partially supported by the project TIN2015-68854-R (FEDER Founds) of the Spanish Ministry of Economy and Competitiveness.

We would like to thank the anonymous reviewers whose comments helped improve and clarify this manuscript.

Bibliography

R. R. Andrawis, A. F. Atiya, and H. El-Shishiny. Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. *International Journal of Forecasting*, 27(3):672 – 688, 2011. URL https://doi.org/10.1016/j.ijforecast.2010.09.005. [p]

S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Syst. Appl.*, 39(8):7067–7083, 2012. URL https://doi.org/10.1016/j.eswa.2012.01.039. [p]

S. F. Crone, M. Hibon, and K. Nikolopoulos. Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, 27 (3):635–660, 2011. URL https://doi.org/10.1016/j.ijforecast.2011.04.001. [p]

O. Dag and C. Yozgatligil. GMDH: An R package for short term forecasting via GMDH-type neural network algorithms. *The R Journal*, 8(1):379–386, 2016. URL https://doi.org/10.32614/RJ-2016-028. [p]

M. Hibon and T. Evgeniou. To combine or not to combine: Selecting among forecasts and their combinations. *International Journal of Forecasting*, 21(1):15–24, 2005. URL https://doi.org/10.1016/j.ijforecast.2004.05.002. [p]

R. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(1):1–22, 2008. URL https://doi.org/10.18637/jss.v027.i03. [p]

R. J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2014. URL https://www.otexts.org/book/fpp. [p]

R. J. Hyndman and B. Billah. Unmasking the Theta method. *International Journal of Forecasting*, 19(2): 287 – 290, 2003. URL https://doi.org/10.1016/S0169-2070(01)00143-1. [p]

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, pages 679–688, 2006. URL https://doi.org/10.1016/j.ijforecast.2006.03.001. [p]

N. Kourentzes. *Nnfor: Time Series Forecasting with Neural Networks*, 2017. URL https://CRAN.R-project.org/package=nnfor. R package version 0.9.2. [p]

M. Kuhn. Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5): 1–26, 2008. URL https://doi.org/10.18637/jss.v028.i05. [p]

F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera. A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, 2017. URL https://doi.org/10.1007/s10462-017-9593-z. [p]

F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera. Dealing with seasonality by narrowing the training set in time series forecasting with kNN. *Expert Systems with Applications*, 103:38–48, 2018. URL https://doi.org/10.1016/j.eswa.2018.03.005. [p]

D. Shaub and P. Ellis. *forecastHybrid: Convenient Functions for Ensemble Time Series Forecasts*, 2019. URL https://CRAN.R-project.org/package=forecastHybrid. R package version 4.2.17. [p]

X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, 2007. URL https://doi.org/10.1007/s10115-007-0114-2. [p]

W. Yan. Toward automatic time-series forecasting using neural networks. *IEEE Trans. Neural Netw. Learning Syst.*, 23(7):1028–1039, 2012. URL https://doi.org/10.1109/TNNLS.2012.2198074. [p]

G. Zhang, B. Eddy Patuwo, and M. Y. Hu. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 1998. URL `https://doi.org/10.1016/S0169-2070(97)00044-7`. [p]

Francisco Martínez
Computer Science Department
University of Jaén
Spain
`fmartin@ujaen.es`

María P. Frías
Statistics and Operations Research Department
University of Jaén
Spain
`mpfrias@ujaen.es`

Francisco Charte
Computer Science Department
University of Jaén
Spain
`fcharte@ujaen.es`

Antonio J. Rivera
Computer Science Department
University of Jaén
Spain
`arivera@ujaen.es`

rollmatch: An R Package for Rolling Entry Matching

by Kasey Jones, Rob Chew, Allison Witman, Yiyan Liu

Abstract The gold standard of experimental research is the randomized control trial. However, interventions are often implemented without a randomized control group for practical or ethical reasons. Propensity score matching (PSM) is a popular method for minimizing the effects of a randomized experiment from observational data by matching members of a treatment group to similar candidates that did not receive the intervention. Traditional PSM is not designed for studies that enroll participants on a rolling basis and does not provide a solution for interventions in which the baseline and intervention period are undefined in the comparison group. Rolling Entry Matching (REM) is a new matching method that addresses both issues. REM selects comparison members who are similar to intervention members with respect to both static (e.g., race) and dynamic (e.g., health conditions) characteristics. This paper will discuss the key components of REM and introduce the **rollmatch** R package.

Introduction

In experimental studies, scientists design research protocols to empirically test their hypotheses of causal relationships between one or more independent variables and an outcome variable. To isolate the effects of a treatment while mitigating confounding introduced by allocation or selection bias, researchers randomly assign treatments whenever possible. In certain scenarios, it is not always feasible to randomize who receives an intervention, due to cost, coordination, or ethical considerations (Resnik, 2008). This situation is particularly common in disciplines that study human behavior and health, including public policy, international development, medicine, and several social sciences disciplines.

To help address this methodological barrier, researchers have developed quasi-experimental designs to estimate the causal impact of an intervention where subjects are not randomly assigned a treatment (Campbell and Stanley (1996); Meyer (1995); Shadish et al. (2001)). Propensity score matching (Rosenbaum and Rubin (1983); Dehejia and Wahba (2002)) is a popular quasi-experimental method that attempts to mimic randomization by matching units that received the treatment with units having similar or identical observable covariates who did not receive treatment. This matching procedure helps create more meaningful comparisons because variables that might contribute to individuals receiving the treatment are controlled for. A propensity score, "the conditional probability of assignment to a particular treatment given a vector of observed covariates" (Rosenbaum and Rubin, 1983), is used to assess similarities between an individual receiving the treatment and potential matches. Though historically researchers have used logistic or probit regression to model propensity scores, machine learning classification methods are becoming attractive alternatives, due to their ability to deal implicitly with interactions and nonlinearities and empirical evidence supporting their ability to accurately predict outcomes (Lee et al. (2010); Westreich et al. (2010)).

PSM falls under the larger umbrella of causal inference methods and is used within the Neyman-Rubin causal modeling framework (Rubin, 1978). Under this framework, obtaining unbiased causal estimates requires two standard assumptions. First, assignment to treatment must be independent of potential outcomes. And second, we assume that all treated individuals receive the same treatment and treatment of one person does not affect the outcome of another.

Rolling Entry Matching

Traditional propensity score matching designs are cross-sectional in nature, matching on covariates before the intervention and measuring outcomes after the intervention to analyze the effect of a treatment at a specific point in time. While effective in many situations, this approach inherently assumes that covariates do not change in a time window relevant for the analysis, or if they do, that these changes will not also affect the outcome variable. In many areas such as health care and epidemiology, relevant time-varying covariates are not uncommon and cause difficulties for traditional matching approaches. Longitudinal settings can also add complexity when exposures or treatments can vary with time or when the treatment entry date is undefined for the control group (Stuart, 2010).

Rolling entry matching (Witman et al., 2018) is a propensity score matching method designed for longitudinal or panel studies where participants to be treated are enrolled on a rolling basis, a common practice in health care interventions where delaying treatment may impact patient health. We can use rolling entry matching to retrospectively select comparison group members who are similar to intervention members with respect to both static characteristics (e.g., race) and dynamic characteristics that change over time (e.g., health conditions). Incorporating time-varying characteristics into the matching procedure is important for health care interventions because a participant's health and medicinal utilization often predict entry into an intervention.

REM is also effective when there is no intervention start date for the comparison group. For certain studies, the comparison group never actually receives an intervention. While this is not a problem for PSM methods in a pre and post setting, matching individuals based on when they *could* have started an intervention is complicated in longitudinal settings with non-uniform intervention start dates. REM address both the rolling entry and missing intervention start date issues.

Typical propensity score methods are not designed to handle rolling entry because the baseline period for potential comparison individuals needs to be different for each treatment participant. To illustrate, consider two hypothetical people: (1) Sue, who started taking a prescription (the treatment), and (2) Jan, who is similar to Sue in static characteristics but does not take this medicine. If Sue started her pills in March, we might compare Sue and Jan's data from February. If Sue started her pills in June, we might compare Sue and Jan's data from May. This is done because Jan could have started taking pills in any month. REM helps in making these comparisons by turning a single comparison individual into multiple psuedo-comparison individuals, one for each unique intervention period occurring in the dataset.

Rolling entry matching requires a quasi-panel dataset and is performed in three phases. The quasi-panel dataset should consist of all available data for both treatment and control subjects and should be longitudinal.

1. **Reduce Data:** The quasi-panel dataset is reduced based on two specifications. First, all treatment observations are filtered to observations whose current time period equals the treatments entry period minus some value. For example, if Sue was treated in May and we want to look back 1 time period, we would filter to Sue's data from April. This value is called `lookback`. And second, after filtering treatment observations, we filter the control observations to those who share a time period with any treatment individual. Continuing our example, we would keep all control data with a time value equal to April.

   The `lookback` value has a default value of 1, as researchers usually consider only the time period directly before entering the study (i.e. `lookback = 1`). In certain studies, researchers would want the `lookback` to be greater than one. For example, researchers could find participants that will begin a new diet in 4 weeks. Their health conditions may change between the announcement and the official beginning of the treatment; `lookback` would be set to four.

2. **Calculate Propensity Scores:** Propensity scores are calculated for all data left after the reduction step.

3. **Find Matches:** Individuals are matched based on their propensity scores and entry period through a matching algorithm developed specifically for REM (see Matching algorithm). When a match is created, the control observation is assigned the intervention start date of the treatment observation.

Rolling entry matching is one of several matching methods used to select a comparison group for treatments that occur on a rolling basis, including balance risk set matching (Li et al., 2011), stepwise matching (Yi, 2014), and sequential cohort matching (Seeger et al. (2005); Schneeweiss et al. (2011); Mack et al. (2013)). In addition, inverse probability propensity score weighting methods, such as marginal structural models (Robins et al., 2000), have also been suggested to deal with time-varying covariates. However, despite its importance across a number of different settings, there are few implementations of longitudinal propensity score methods for R. At the time of writing, the only packages that natively support longitudinal propensity score methods are (1) the **CBPS** package, which implements covariate balancing propensity score for longitudinal settings to be used in conjunction with marginal structural models (Imai and Ratkovic, 2015); (2) the **ipw** package, which allows users to estimate marginal structural models; and (3) the **rollmatch** package, which implements rolling entry matching. Of these three, only rollmatch provides an integrated matching approach, as both **CBPS** and **ipw** rely on propensity score weighting.

We now introduce **rollmatch**, an R package for performing rolling entry matching. In particular, we will provide an overview of the main functions in **rollmatch**, a walk-through of the rolling entry matching algorithm, and commentary on relevant parameter choices such as caliper selection.

The rollmatch package

The **rollmatch** package is comprised of three functions.

- `reduce_data()`: Step 1 of REM - Reduces the input panel dataset
- `score_data()`: Step 2 of REM - Calculates propensity scores for the reduced data. This function is not required if users want to develop their own propensity score models
- `rollmatch()`: Step 3 of REM - Performs the matching algorithm and produces output

rollmatch example:

```
library(rollmatch)
data(package = "rollmatch", "rem_synthdata_small")
reduced_data <- reduce_data(data = rem_synthdata_small, treat = "treat",
                            tm = "quarter", entry = "entry_q",
                            id = "indiv_id", lookback = 1)
fm <- as.formula(treat ~ qtr_pmt + yr_pmt + age)
vars <- all.vars(fm)
scored_data <- score_data(reduced_data = reduced_data,
                          model_type = "logistic", match_on = "logit",
                          fm = fm, treat = "treat",
                          tm = "quarter", entry = "entry_q", id = "indiv_id")
output <- rollmatch(scored_data, data=rem_synthdata_small, treat = "treat",
                    tm = "quarter", entry = "entry_q", id = "indiv_id",
                    vars = vars, lookback = 1, alpha = .2,
                    standard_deviation = "average", num_matches = 3,
                    replacement = TRUE)
```

Rolling entry matching: a walkthrough

This section describes the operations performed in **rollmatch** through an illustrative example. Though some of these operations are hidden from the user, understanding the matching algorithm will help troubleshoot potential errors and better inform the selection of parameter values. In addition to discussing steps taken to trim potential matches and calculate propensity scores, special attention is paid to the specifics of the matching algorithm.

Step 1: Trim the treatment data

We begin with a panel dataset that includes individuals who received an intervention at different time periods, as well as other individuals that are being considered for selection into the comparison group. For each individual, we have background variables (e.g., demographics, health conditions, spending habits, etc.) at each time step, an indicator variable for if the individual was treated, a variable specifying the time period of the observation, and a time period variable for when the participant entered the intervention. We let `Treat = 1` indicate an individual who had an intervention and `Treat = 0` indicate someone who did not. Finally, we will let `lookback = 1`.

ID	Treat	Time	Entry	Background Variables			
X	1	1	2	...	...	...	...
X	1	2	2	...	...	...	...
X	1	3	2	...	...	...	...
Y	1	1	2	...	...	...	...
...	...	...	...	...	...	...	...

Table 1: Example dataset of treated observations

In this example, individual X has 3 quarters of data and is part of the treatment group. Since participant X entered the treatment in time period 2 and `lookback = 1`, her data from time period 1 will be used for matching with control observations. As REM allows for matching on dynamic variables that can change over time, matching individual *X* on observations prior to the intervention provides a clean comparison in which we do not need to worry about the influence of the intervention on the dynamic covariates.

Recall that `lookback` can be written as `entry-time`. Rows that do not match `entry-time = 1` are then dropped.

ID	Treat	Time	Entry	Background Variables			
X	1	1	2	...	...	...	...
Y	1	1	2	...	...	...	...
...	...	...	...	...	...	...	...

Table 2: Dropping treated observations based on `lookback = 1`

Step 2: Trim Control data

Let Table 3 represent the control data.

ID	Treat	Time	Background Variables			
A	0	1	...	...	...	...
A	0	2	...	...	...	...
A	0	3	...	...	...	...
B	0	1	...	...	...	...
...	...	...	...	...	...	...

Table 3: Original control observations

Since rolling entry matching requires that the entry period of any potential comparison observations be equal to the entry period of a treatment observation, we drop all comparison observations that do not share a time period with at least one treatment record. Our example treatment observation data only has individuals that enter the intervention at time periods 2 and 3. Therefore, we only look at control observations whose time is equal to 1 or 2.

ID	Treat	Time	Background Variables			
A	0	1	...	...	...	...
A	0	2	...	...	...	...
B	0	1	...	...	...	...
...	...	...	...	...	...	...

Table 4: Control data after dropping observations

Step 3: Calculate propensity scores and absolute differences for all possible matches

Users are allowed to calculate their own propensity scores to use with the **rollmatch** matching algorithm, or they can use the scoring function provided in the package. If using `score_data()`, users can specify either "logistic" or "probit" regression and the formula for the model (i.e., selecting the covariates to be used). Once a propensity score has been generated for all observations, we look at the absolute difference in scores for all possible matches between control and treatment observations. Recall that in order to be a match, the time period of a control observation must match the time period of a treatment observation. We have provided Table 5 in full so that we can go into detail about the matching algorithm.

Time	Treat ID	Treat Score	Control ID	Control Score	**Difference**
1	X	0.95	A	0.16	**0.79**
1	X	0.95	B	0.42	**0.53**
1	X	0.95	C	0.61	**0.34**
1	X	0.95	D	0.32	**0.63**
1	X	0.95	E	0.15	**0.80**
1	Y	0.03	A	0.16	**0.13**
1	Y	0.03	B	0.42	**0.39**
1	Y	0.03	C	0.61	**0.58**
1	Y	0.03	D	0.32	**0.29**
1	Y	0.03	E	0.15	**0.12**
2	Z	0.65	A	0.63	**0.02**
2	Z	0.65	B	0.26	**0.39**
2	Z	0.65	C	0.05	**0.60**
2	Z	0.65	D	0.57	**0.08**
2	Z	0.65	E	0.43	**0.22**
2	Q	0.11	A	0.63	**0.52**
2	Q	0.11	B	0.26	**0.15**
2	Q	0.11	C	0.05	**0.06**
2	Q	0.11	D	0.57	**0.46**
2	Q	0.11	E	0.43	**0.32**

Table 5: Calculated absolute differences for all matches from Table 2 and Table 4.

Step 4: Trim the Comparison Pool

Caliper

For the data in Table 5, treatment id X has been compared to control ids A, B, C, D, and E. The lowest difference value for these five comparisons is .34, which while being the best match available, may still be too different to provide a high quality match and may bias estimates of the outcome if included (Lunt, 2013). To limit the potential matches, an alpha value between 0 and 1 can be specified. The alpha value is a scaling factor that effects which propensity scores are considered. A value closer to 1 allows for a wider range of propensity scores to be considered, while a value close to 0 provides stricter requirements for matching. The alpha value is multiplied by the pooled standard deviation of the propensity scores; this final value is called the caliper and is used as a cutoff.

Consequently, if an alpha is specified, there is no guarantee that each treatment ID will receive a match. As caliper selection can play a large role in selecting potential matches, we have provided Appendix A: Theorem and Appendix B: Selecting the appropriate pooled standard deviation discussing caliper selection.

Number of Matches

When running `rollmatch()`, the user can specify the maximum number of control matches that should be assigned, when possible, to each treatment observation. If the user sets this to one, and no additional steps are taken, every single treatment observation will be assigned one control observation, regardless of the quality of their best-match (assuming there are enough control observations). However, if the user specifies an alpha value and a caliper is used, there may be some treatment observations that do not receive a match. As the value of alpha decreases, the likelihood that some treatment observations do not have a match will rise. If any treatment observations are not matched, their ids are listed in the output as `ids_not_matched`.

Currently, the user can only guarantee that each treatment observation be assigned at least one match (i.e. by not specifying an alpha). In a future version of **rollmatch**, the user will be able to specify the number of matches to attempt to create (`num_matches`), as well as a minimum number of matches to create. This would ensure that each treatment observation is matched with some number of control individuals, regardless of the alpha selected. It would also allow for other treatment observations to be matched to more observations if enough control individuals are within the caliper.

For simplicity, we will not trim the comparison pool from Table 5 for our example.

Step 5: Assign matches

After the comparison pool has been created and trimmed, treatment and control observations are matched. Rosenbaum and Rubin (1985) used the following matching rules:

1. Randomly order treatment observations
2. For the first treatment subject, based on the comparison pool find all comparison matches for the treated observation whose difference is less than the caliper. If no match exists, match treated observation to control observation with smallest difference
3. From this group, select a match based on the Mahalanobis distance for the background variables
4. Remove the treated and matched observation and repeat steps 2-4 for the next treated observation

There have already been several R packages released that make use of this original algorithm while making modifications to the algorithm to fit the specific goal of the package. Packages such as **MatchIt**, **Matching**, and **optmatch** all offer various matching algorithms for propensity scores.

Rolling entry matching takes a different approach by matching non-participants based on the entry period for which their data is most similar to their matched participant. Whereas other methods like sequential cohort matching (Seeger et al., 2005) start from specific cohorts to begin matching (allowing early cohorts to get the matches that work the best for them without consideration of later cohorts), rolling entry matching considers all periods when matching non-participants to participants. The algorithm for rollmatch must be different because control participants are treated as if they could enter the study at any time. This creates a lot more potential matches per observation. Furthermore, a control observation could best match multiple treatment observations across multiple quarters of entry and there must be logic to handle this scenario.

Matching algorithm

Each treated observation is initially assigned to its best-matching control observation based on the smallest absolute difference between their propensity scores. Recall that the comparison pool only consists of treatment/control pairs that have already been matched on their entry period. As long as the control is not the best-match for another treated observation of a different entry quarter, then the two are matched and the algorithm continues. Recall that any given control individual could have several data entries (one for each quarter they have data available). It is possible that a control observation could be matched to two different treatment observations who entered a study in different time periods. Using Table 5 we have the following matches for iteration one of the algorithm:

Time	Treat ID	Treat Score	Control ID	Control Score	Difference
1	X	0.95	C	0.61	0.34
1	Y	0.03	E	0.15	0.12
2	Q	0.11	C	0.05	0.06
1	Z	0.65	A	0.63	0.02

Table 6: Possible matches for iteration one

Notice that control individual C has been matched to X in time period 1 and matched to Q in time period 2. Since the propensity score difference between Q and C is smaller than the difference between X and C, Q will be matched to C for time period 2, and X will not be matched to any control this round. If X and Q enrolled in the same quarter, then control C would be matched to both treatments if the `replacement` parameter was set to `TRUE`, indicating matching with replacement is desired. `replacement`

Time	Treat ID	Treat Score	Control ID	Control Score	Difference
1	X	.95	C	.61	.34
1	Y	.03	E	.15	.12
1	W	.70	C	.61	.09
2	Q	.11	C	.05	.06

Table 7: Alternative possible matches for iteration one

allows for multiple treatments to be assigned the same control observation if the treatments enrolled in the same quarter. Consider the alternative set of matches in Table 7.

If `replacement` is TRUE and a control individual is matched to multiple treatment IDs for multiple quarters, we take the average difference for all treatment IDs in each quarter to make the final decision. In Table 7, control C is the best match for X and W in time period 1, and the best match for Q in time period 2. The average difference for X and W is .215 and the average difference for Q is simply .06. In this case, Control C will only be matched with treatment Q.

After each iteration of assignment, any matched treatment and control observations are removed from the pool of potential matches and the process is repeated. Once all treatment observations have been assigned the desired number of matches, or there are no more possible matches remaining, the process is complete.

An explaination of caliper selection

Rosenbaum and Rubin used the results of Cochran and Rubin (1973) to conclude that under certain conditions, specific caliper widths could remove a certain percentage of the bias of confounding variables (Rosenbaum and Rubin, 1985). Let σ_1^2 and σ_2^2 be the variances of the logit of the propensity scores (referred to as just variance going forward) for the treated and control groups, and let:

$$\sigma = \sqrt{[(\sigma_1^2 + \sigma_2^2)/2]}. \tag{1}$$

Finally, let our caliper equal $\alpha * \sigma$. According to Rosenbaum and Rubin, at different levels of α, we can remove different levels of bias. Austin (2010) conducted Monte Carlo simulations to verify these findings. We have outlined the reduction in bias in Table 8 . Note that in this case, the variance for the treatment and control groups must be equal.

Alpha	Rosenbaum and Rubin	Austin
.2	99%	At least 99.3%
.6	89%	95.2%-99.6%

Table 8: Expected bias reduction at various α levels

The likelihood that the variance of the two groups being equal is unknown, and although Rosenbaum and Rubin (1985) provided estimates for bias reduction when they are equal, the guidance on selecting a caliper is minimal. We have left the selection of the caliper width up to the user, but we will go into further detail about the two parameters effecting the caliper that are included in **rollmatch**.

The `alpha` parameter must be 0 or greater. At 0, the trimming function is ignored. For all values above 0, the dataset of potential control matches is trimmed based on if the difference between scores is less than $\alpha * \sigma$.

The second decision the user can make is on how sigma is calculated. Both Rosenbaum and Rubin, and Austin use the pooled standard deviation, which we have defined as σ above (Rosenbaum and Rubin (1983); Austin (2010)). Consider the alternative formula for pooled standard deviation for i groups:

$$\sigma = \sqrt{\frac{(n_1-1)\sigma_1^2 + (n_2-1)\sigma_2^2 + \ldots + (n_i-1)\sigma_i^2}{(n_1+n_2+\ldots+n_i)-i}} \tag{2}$$

Let σ_{f1} be the average pooled standard deviation that our sources have been using so far, and let σ_{f2} be equal to the weighted pooled standard deviation for that we just introduced (for $i = 2$). These two calculations are only equal only under specific conditions (see Appendix A: Theorem). If a dataset has a much larger treatment or control group, or the variances for the two groups propensity scores are vastly different, the weighted pooled standard deviation may do a better job at selecting a cutoff.

Conclusion

We have presented **rollmatch** as an R package for performing rolling entry matching. When observational studies are conducted on a rolling entry basis or when control entry periods do not exist, **rollmatch** is an effective package for finding matches between treated and untreated subjects. The amount of bias introduced by confounding variables can often be reduced by using propensity score matching. However, rolling entry matching furthers this ability by matching treated individuals to control individuals as if they were enrolled at the same time. The parameters and options included in **rollmatch** create a robust and user friendly package. We hope to continue expanding this package as further development of rolling entry matching is completed.

Acknowledgements

rollmatch was made possible by the research conducting on rolling entry matching by Allison Witman, Yiyan Liu, Mahin Manley, and Chris Beadles. The concepts, matching algorithm, and techniques demonstrated in this paper should be credited to them. We would also like to thank Georgiy Bobashev, a Fellow at RTI International. This paper would not have been written without him.

Appendix A: Theorem

Theorem 1. *σ_{f1} is equal to σ_{f2} if and only if $n_1 = n_2$ or $\sigma_1 = \sigma_2$*

Proof. Assume σ_{f1} and σ_{f2} are equal. We will show that this is true only when $\mathrm{n}_1 = \mathrm{n}_2$ or $\sigma_1 = \sigma_2$.

$$\begin{aligned} \sigma_{f1} &= \sigma_{f2} \\ \sqrt{\frac{\sigma_1^2+\sigma_2^2}{2}} &= \sqrt{\frac{(n_1-1)\sigma_1^2+(n_2-1)\sigma_2^2}{n_1+n_2-2}} \end{aligned}$$

Variance's are positive by nature and the number of samples in each group must be greater than 0. We can remove the square root.

$$\begin{aligned} \frac{\sigma_1^2+\sigma_2^2}{2} &= \frac{(n_1-1)\sigma_1^2+(n_2-1)\sigma_2^2}{n_1+n_2-2} \\ (\sigma_1^2+\sigma_2^2)(n_1+n_2-2) &= 2[(n_1-1)*\sigma_1^2+(n_2-1)*\sigma_2^2] \\ \sigma_1^2*n_1+\sigma_1^2*n_2+\sigma_2^2*n_1+\sigma_2^2*n_2 &= 2\sigma_1^2*n_1+2\sigma_2^2*n_2 \\ \sigma_1^2*n_2+\sigma_2^2*n_1 &= \sigma_1^2*n_1+\sigma_2^2*n_2 \end{aligned}$$

This can be written two ways.

$$\begin{aligned} \sigma_1^2*(n_2-n_1) &= \sigma_2^2*(n_2-n_1) \\ n_2(\sigma_1^2-\sigma_2^2) &= n_1(\sigma_1^2-\sigma_2^2) \end{aligned}$$

Let's examine these two equations. If $n_1 \neq n_2$, we can divide $(n_2 - n_1)$ from both sides of the first equation, and we end up with $\sigma_1^2 = \sigma_2^2$. This implies $\sigma_1 = \sigma_2$ because variances cannot be negative. Similarly, if $\sigma_1 \neq \sigma_2$, we can divide $(\sigma_1^2 - \sigma_2^2)$ from both sides of the second equation and we find that $n_1 = n_2$. The original equation can only hold if at least one equality holds: $\mathrm{n}_1 = \mathrm{n}_2$ or $\sigma_1 = \sigma_2$.

□

Appendix B: Selecting the appropriate pooled standard deviation

Selection between σ_{f1} and σ_{f2} is important when the variances of the treatment and control group are not equal. Setting $\alpha = .2$ may not reduce 99% of the bias due to confounding variables if this is true.

Let us examine how different our results are when using different options. We will use the following parameters:

```
formula <- as.formula(treat ~ qtr_pmt + yr_pmt + age)
tm = 'quarter'
entry = 'entry_q'
id = 'indiv_id'
lookback = 1
match_on = 'logit'
model_type = 'logistic'
```

For the smaller synthetic dataset, the variance (of the logit of the propensity score) of our treated group is .891. While the variance of the untreated group is 4.690. In this case, σ_{f1} is equal to 1.658 and σ_{f2} equals 2.141. The original comparison pool had 15,000 treatment and control comparison. Table 9 shows how the alpha value and choice of sigma limit the number of potential matches.

Table 9: Pooled standard deviation comparisons

Alpha	Sigma	Comparions Available
.2	σ_{f1}	414
.2	σ_{f2}	516
.6	σ_{f1}	1044
.6	σ_{f2}	1192
1.0	σ_{f1}	1350
1.0	σ_{f2}	1451

We did not do any simulations of our own to determine how much bias could be reduced when variances are not equal and when the two σ calculations are implemented. Some studies such as Wang et al. (2013) have used σ_{f2} in their calculations. However, when they used only a single treatment group they still assumed equal variances.

To summarize why this is important, if variances among the groups are not equal, the amount of bias reduced at certain levels of alpha will not be the same as what is suggested by Table 8.