Observational Studies 2 (2016) 4-23 Submitted 7/15; Published 1/16

Large, Sparse Optimal Matching with R package rcbalance

Samuel D. Pimentel **spi@wharton.upenn.edu**
Department of Statistics
Wharton School, University of Pennsylvania
Philadelphia, PA 19104-6304.

Abstract

A new R package for matching in observational studies, `rcbalance`, is presented. `rcbalance` is designed to exploit sparsity among potential treated-control pairings and can conduct matches on a very large scale at low computational cost. Unlike existing packages, it also supports refined covariate balance constraints, which use prioritized lists of nominal covariates to induce high degrees of balance on the covariates and their interactions, even when it is difficult to find individual pairs that are similar on many covariates. Matching with `rcbalance` is demonstrated using data from an observational study of right heart catheterization.

Keywords: Pair matching; observational studies; fine balance; sparse matching; near-exact matching; optimal subset matching.

1 Introduction

1.1 Matching in observational studies

In an observational study, the population of units receiving a treatment of interest may differ systematically before treatment from the population of units receiving a control condition. These systematic differences may be manifest in observed variables, or they may be due to unobserved variables. Treatment effect estimates will be biased unless these confounding variables are properly accounted for (Rosenbaum, 1989).

Matching provides an effective way to adjust for observed treatment-control confounding. By creating sets of treated and control units with similar values of observed covariates and performing analysis within these sets, substantial bias may be removed when estimating a treatment effect. For discussions of matching in various contexts see Rosenbaum and Rubin (1985), Stuart (2010), Lu et al. (2011), and Baiocchi et al. (2012), and for examples of matching in medical research see Silber et al. (2013), Silber et al. (2014), and Neuman et al. (2014). Matching cannot account directly for unobserved sources of bias; for methods of sensitivity analysis designed to account for unobserved confounding see Rosenbaum (2009) and Rosenbaum (2015).

1.2 Package rcbalance

This article describes the R package `rcbalance`, which conducts large, sparse optimal matching. Given a study population of treated and control units, a list of potential control partners for each treated unit, a covariate distance for each potential pairwise treated-control population, and (optionally) a set of balance constraints, the package's main function produces a collection of matched sets of treated and control units that achieves the optimal (i.e. smallest) total covariate distance subject to the balance constraints.

Other packages for matching in R include `Matching` (Sekhon, 2011), `MatchIt` (Ho et al., 2011), `mipmatch` (Zubizarreta, 2012), `nbpMatching` (Beck et al., 2015), and `optmatch` (Hansen and Klopfer, 2006). Many of these do not support optimal matching as described in Section 2.1. Among those that do, the package most closely related to `rcbalance` is `optmatch`. Both packages compute optimal matches by formulating them as network flow problems — in fact, `rcbalance` actually leverages an optimization routine provided by `optmatch` to do so — and both are designed to support sparse matching, in which only a relative small subset of potential treated-control links may be considered. Sparse matching has considerable computational advantages for large problems, and many datasets have structure that makes it a natural choice.

In contrast to existing packages, however, `rcbalance` allows users to specify refined covariate balance constraints (hence the name of the package). Most matching techniques rely on finding close matches in terms of pairwise covariate distances or propensity scores in order to induce overall covariate balance between the treated and control groups. This is similar to the type of balance achieved by randomization and is called stochastic balancing. Stochastic balancing guarantees close balance only in large samples; for an example where it fails badly in practice, see Zubizarreta et al. (2011). In contrast, matching with fine balance or refined covariate balance constraints can produce high degrees of balance directly in finite sample situations where stochastic balancing fails. Refined covariate balance constraints are particularly flexible and useful since they allow the user to prioritize balance on certain covariates over others.

1.3 Outline

Section 2 provides explanations of sparse matching and refined covariate balance and explains why these concepts are useful in designing observational studies. Section 3 describes the steps of constructing a match using `rcbalance`, using an example from a study of right heart catheterization. Section 4 discusses additional features of `rcbalance`, including near-exact matching, matching with multiple controls per treated unit, and exclusion of treated units. Section 5 presents a generalization of refined covariate balance appropriate for situations in which some differences between the treated group and the matched controls are desired, and describes how such constraints can be implemented using `rcbalance`.

2 Key concepts

2.1 Optimal pair matching

In optimal pair matching, each treated unit in the study is paired to a similar control. Similarity between treated and control units is measured by a covariate distance of some kind, often a form of the Mahalanobis distance. Pairs are formed so that the sum of within-pair covariate distances across all pairs is made as small as possible. Forming such a match requires solving a linear optimization problem (hence the name "optimal" matching).

Optimal matching guarantees that the resulting match has the lowest average within-pair dissimilarity possible. Non-overlapping nearest-neighbor or greedy matching methods do not give us such a guarantee. For discussion of the advantages of optimal matching over greedy matching, see Rosenbaum (1989).

Algorithms for optimal matching generally formulate matching as an optimization problem in canonical form, in order to leverage existing routines developed by computer scientists and operations researchers. Many matching problems can be formulated as network flow optimization problems. Network flow optimization involves sending discrete units of "flow" from certain nodes of a directed graph to other nodes, respecting flow capacities on the graph's edges and paying a flow cost for each edge used. The discrete optimization problem is to find the cheapest overall collection of paths by which all units of flow may be delivered from their sources to their destinations. For more discussion of network flow problems, see Bertsekas (1991) and section 1.12 of Schrijver (2003). Matching can be put into this framework by forming a network with a node for each treated unit, a node for each control, and edges connecting treated units to controls with costs given by pairwise covariate distances, as shown in Figure 1. Additional features may be included in the graph to enforce balance constraints of various kinds; for examples of this, see the online supplement to Yang et al. (2012), and Pimentel et al. (2015a).

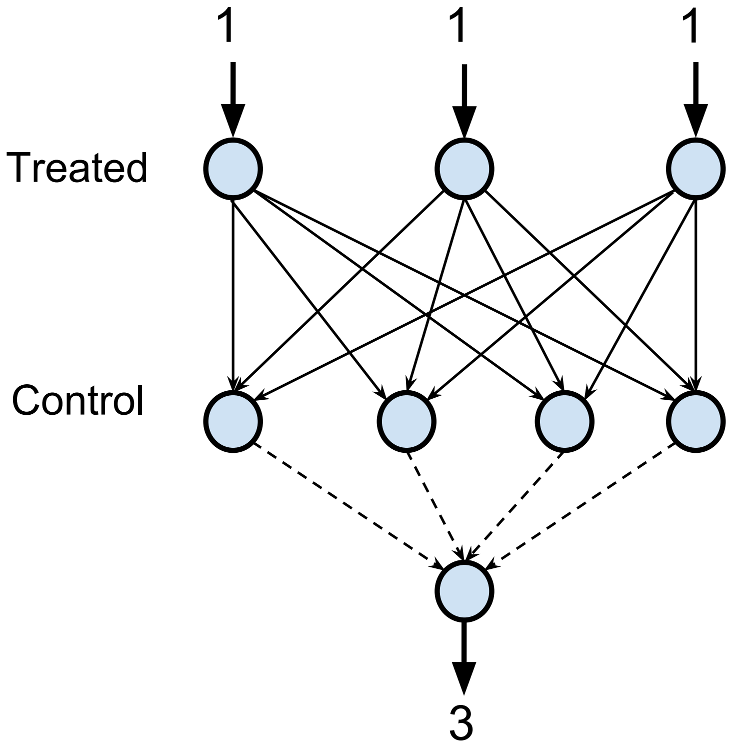
Software implementations of network flow optimization algorithms are widely available. In particular, Dimitri Bertsekas' RELAX-IV algorithm (Bertsekas et al., 1994), which is implemented efficiently in FORTRAN, is used by the `optmatch` package to solve network flow problems. The `rcbalance` package conducts matching using this same solver. In particular, to perform a match in `rcbalance`, the user must load `optmatch` and agree to its software license, after which matches are computed using `optmatch`'s interface to the FORTRAN solver.

2.2 Sparse optimal matching

As mentioned in Section 2.1, optimal matching problems usually permit any treated unit to be matched to any control. However, we might instead constrain the set of possible control partners for a given treated unit, specifying a short list of allowed matches from among a large control pool. In the network flow formulation of matching shown in Figure 1, one would implement this strategy by removing many of the edges that connect treated nodes to control nodes, allowing only a small set of edges to depart from each treated node. Matching with small, constrained sets of potential controls for each treated unit is called sparse matching, while the usual strategy of allowing all potential treated-control pairings to be considered is called dense matching. For a more technical explanation see Pimentel et al. (2015a).

There are a number of reasons why sparse matching might make sense for a practical matching problem. For example, in many matching problems there are natural subgroups or "blocks" within which units are expected to be less heterogeneous, and often researchers wish to match units within such blocks to those within the same block, or to "match exactly" on the variable defining the blocks. In clinical studies researchers might wish to match patients within hospitals; in education research, it might be desirable to match

Figure 1: Matching as a network flow problem. The row of treated units at the top of the figure each have a supply of one unit of flow, and the sink (at the bottom of the figure) has a demand for three units of flow. All edges have capacity 1, meaning at most 1 unit of flow is allowed to cross each edge. The dashed edges have zero cost and the solid edges have costs given by pairwise distances between treated and control units.



classrooms within a school. In such cases sparsity can be induced by exact matching, or by forbidding all matches outside of a unit's own natural block.

Another way of inducing sparsity is to forbid matches between individuals whose propensity scores differed by more than a small fixed amount. This is often known as caliper matching. It can be used in addition to or in place of matching exactly on natural blocks.

In principle, the matches formed by sparse matching algorithms could also be produced by dense matching: one could construct a covariate distance that simply assigned very large penalties to the pairings forbidden in the sparse match, effectively forcing the distance-minimization algorithm to choose matches from among the smaller edge sets. However, sparse matching offers a substantial computational advantage over dense matching that makes it practically useful in cases where dense matching is intractable. While dense matching algorithms may fail to produce answers in reasonable time once the total number of treated and control units N climbs into the tens of thousands, sufficiently sparse matching algorithms can still run efficiently for values of N in the hundreds of thousands; for an example of such a large, sparse match see Pimentel et al. (2015a). The `rcbalance` package is specifically designed to exploit this difference (as are the `exactMatch` and `caliper` functions in the `optmatch` package), and makes it easy to introduce sparsity and reap the associated computational rewards for large datasets.

2.3 Fine and near fine balance

The explicit goal of optimal matching is to form treated-control pairs with similar values on covariates. However, in conducting an analysis the main concern is often that the matched control group (i.e. the set of all controls paired to a treated unit) and the treated group have the same covariate distributions overall. This latter condition is called fine balance, and it is less restrictive than achieving pairwise closeness; for example, in a study where treated patients are matched to controls, even if males are matched to females in certain pairs then fine balance on sex is still achieved if the total numbers of males and females in the matched control group are equal to the totals in the treated group. While finding close pairwise matches may produce similar covariate distributions in the treated and matched control groups overall, there is no guarantee of this, especially when many nominal covariates are used for matching (Zubizarreta et al., 2011). To ensure that the treated and matched control distributions of a key covariate are similar, matching may be conducted with a fine balance constraint on an important variable. This guarantees that match produced satisfies fine balance with respect to the key variable and is optimal among all such matches. In the patient match example, this would mean finding the best pairwise match for which the treated group and matched control group had the same numbers of males and females. For a discussion of fine balance and an example of a study design that uses it see Rosenbaum et al. (2007).

In some cases it may not be possible to achieve exact fine balance. For example, if there are more males in the treated group than there are in the entire set of controls in the study, then we can never select a matched control group with the appropriate number of males for sex to be balanced exactly. In such situations we can use a generalization of fine balance called near fine balance. Under a near fine balance constraint on a covariate, the match produced is optimal among matches that minimize the discrepancy between the covariate distribution between treated units and matched controls. So in a patient match where there were more treated males than control males, matching with near fine balance on sex would produce the best pairwise match that included all the males from the control group, bringing the count of male patients as close as possible to the count in the control group. For a more thorough explanation of near fine balance, see Yang et al. (2012).

2.4 Refined covariate balance

Refined covariate balance is an extension of fine balance and near fine balance for settings with large numbers of nominal covariates. To achieve fine balance on multiple nominal covariates, one might impose a near fine balance constraint on an interaction of the covariates (creating a new category for each unique combination of values of the covariates). However, as the number of variables in the interaction increases the number of categories explodes exponentially and the resulting variables can be difficult to balance well. In addition, an interaction treats each component variable as equally important. In practice, researchers often care a great deal about balancing certain important variables and place less priority on others.

Suppose we have a list of K nominal covariates, arranged in descending order of importance in our study. Suppose also that the covariates are nested, so that the categories of the second covariate are all subcategories of the first covariate, and the covariates become progressively finer across the list. A match satisfies refined covariate balance if:

1. The match satisfies near fine balance for the first-priority covariate.

2. The treated and matched-control distributions of the second covariate are as close as possible among matches satisfying (1).

3. The treated and matched-control distributions of the third covariate are as close as possible among matches satisfying (1) and (2).

$\vdots$

(K) The treated and matched-control distributions of the Kth covariate are as close as possible among matches satisfying (1), (2), . . ., ($K - 1$).

Intuitively, refined covariate balance asks for balance in priority order, requiring near fine balance on the most important (and coarsest) covariate and asking for the closest remaining balance on the remainder, in decreasing order of importance.

In many real datasets the special nested list of covariates may not arise naturally, but it is easy to create such a list. One can divide the nominal covariates in one's data into K groups in decreasing order of importance and then create a nested hierarchy of interactions among these variables. For example, suppose we care about balancing $K = 2$ groups of covariates, a high priority group consisting of patient sex and a patient diabetes indicator (yes/no), and a low priority group consisting of a patient asthma indicator (yes/no). Then we can form a hierarchy of two balance variables, where the first level is an interaction of sex and diabetes status (with four categories: males with diabetes, males without diabetes, females with diabetes, and females without diabetes), and the second level is a further interaction of the first variable with the patient asthma indicator (with categories such as males with diabetes and asthma, males with diabetes but not asthma, etc.). This approach generalizes easily to larger values of K and larger sets of covariates at each level.

For a formal definition of refined covariate balance constraints, a matching study that uses them, and an example of a balance hierarchy created by interactions among nominal variables, see Pimentel et al. (2015a).

3 An Example Using `rcbalance`

3.1 Data

Within the first 24 hours after hospitalization, many cardiology patients receive a procedure known as right heart catheterization (RHC) in which a doctor inserts a catheter into the patient's heart to allow measurement of important diagnostic variables. While RHC may lead to better outcomes when the information it provides guides subsequent treatment, the procedure itself has its own risks. Connors and co-authors conducted a matched comparison among ICU patients in 5 hospitals to determine the impact of the RHC procedure on clinical outcomes (Connors et al., 1996). We will conduct a matched analysis using `rcbalance` to answer the same question from their data. The dataset has 5,735 patients, of whom 2,184 received the RHC procedure. Table 1 gives an overview of the variables in the data.

Table 1: Description of main variables in the data. The ”Code” column gives the variable names as they will be referenced in the R code, and the ”Variable” column gives the form in which they will be described in other tables and in the text.

		Mean		Proportion Missing	
Variable	Code	Treated	Control	Treated	Control
Right heart catheterization	**swang1**				
Yes	No RHC	0.00	1.00	0.00	0.00
No	RHC	1.00	0.00	0.00	0.00
Primary disease category	**cat1**				
Acute renal failure	ARF	0.42	0.45	0.00	0.00
Chronic obstructive pulmonary disease	COPD	0.03	0.11	0.00	0.00
Congestive heart failure	CHF	0.10	0.07	0.00	0.00
Cirrhosis	Cirrhosis	0.02	0.05	0.00	0.00
Coma	Coma	0.04	0.10	0.00	0.00
Colon Cancer	Colon Cancer	0.00	0.00	0.00	0.00
Lung Cancer	Lung Cancer	0.00	0.01	0.00	0.00
Multiple organ system failure w/ malignancy	MOSF w/Malignancy	0.07	0.07	0.00	0.00
Multiple organ system failure w/ sepsis	MOSF w/Sepsis	0.32	0.15	0.00	0.00
Primary disease category	**cat2**				
Acute renal failure	ARF	0.00	0.00	0.77	0.81
Chronic obstructive pulmonary disease	COPD	0.00	0.00	0.77	0.81
Congestive heart failure	CHF	0.00	0.00	0.77	0.81
Cirrhosis	Cirrhosis	0.02	0.04	0.77	0.81
Coma	Coma	0.04	0.10	0.77	0.81
Colon Cancer	Colon Cancer	0.00	0.00	0.77	0.81
Lung Cancer	Lung Cancer	0.00	0.02	0.77	0.81
Multiple organ system failure w/ malignancy	MOSF w/Malignancy	0.11	0.25	0.77	0.81
Multiple organ system failure w/ sepsis	MOSF w/Sepsis	0.82	0.59	0.77	0.81
Race	**race**				
White	white	0.78	0.78	0.00	0.00
Black	black	0.15	0.16	0.00	0.00
Asian	asian	0.00	0.00	0.00	0.00
Other	other	0.07	0.06	0.00	0.00
Insurance class	**ninsclass**				
No insurance	No insurance	0.06	0.05	0.00	0.00
Private	Private	0.33	0.27	0.00	0.00
Medicare	Medicare	0.23	0.27	0.00	0.00
Medicaid	Medicaid	0.09	0.13	0.00	0.00
Private & Medicare	Private & Medicare	0.22	0.21	0.00	0.00
Medicare & Medicaid	Medicare & Medicaid	0.06	0.07	0.00	0.00
Cancer status	**ca**				
No cancer	No	0.79	0.75	0.00	0.00
Cancer	Yes	0.15	0.18	0.00	0.00
Metastatic cancer	Metastatic	0.06	0.07	0.00	0.00
Sex	**sex**				
Female	Female	0.41	0.46	0.00	0.00
Male	Male	0.59	0.54	0.00	0.00
Age	age	60.75	61.76	0.00	0.00
Cardiovascular disease	cardiohx	0.20	0.16	0.00	0.00
Congestive heart failure	chfhx	0.19	0.17	0.00	0.00
Dementia	dementhx	0.07	0.12	0.00	0.00
Psychiatric history	psychhx	0.05	0.08	0.00	0.00
Chronic pulmonary disease	chrpulhx	0.14	0.22	0.00	0.00
Chronic renal disease	renalhx	0.05	0.04	0.00	0.00
Liver disease	liverhx	0.06	0.07	0.00	0.00
Upper GI bleeding	gibledhx	0.02	0.04	0.00	0.00
Solid Tumor/Metastatic Disease/Leukemia	malighx	0.20	0.25	0.00	0.00
Immunosuppression	immunhx	0.29	0.26	0.00	0.00
Transfer from another hospital	transhx	0.15	0.09	0.00	0.00
Myocardial infarction	amihx	0.04	0.03	0.00	0.00
Activities of Daily Living scale	adld3p	1.02	1.24	0.82	0.70
Duke Activity Score index	das2d3pc	20.70	20.37	0.00	0.00
APACHE score	aps1	60.74	50.93	0.00	0.00
Estimated 2-month survival probability	surv2md1	0.57	0.61	0.00	0.00
Glasgow Coma score	scoma1	18.97	22.25	0.00	0.00

3.2 Sparsity via natural blocks

The first step in constructing a sparse, optimal match is to identify sparse structure in our problem. In this example we examine covariates and their interactions and seek to identify a blocking covariate.

A good blocking covariate should be important, in the sense that the analysis must adjust well for it to be credible. Since matching within blocks of a covariate imposes exact similarity between treated and control groups on that covariate, it is desirable to achieve this similarity on a covariate that matters. A covariate is important if it is thought to have a strong influence on both treatment assignment and outcomes and/or to be highly correlated with unobserved confounders influencing treatment and outcomes.

In addition, in large problems a good blocking covariate must vastly reduce the number of total candidate treatment-control pairings in order to make the match computationally feasible. This occurs when no individual category or block contains too many individuals from both treated and control groups. How can one tell whether blocks are small enough to permit a computationally tractable match? The runtime of the matching algorithm is driven mainly by the number of treatment-control pairings, so a good rule of thumb is to consider whether dense matches with similar numbers of pairings would be feasible. For example, a sparse match with 100,000 treatment-control pairings is likely feasible for most computers, since a dense match with 100 treated units and 1000 controls (for 100,000 total pairings) is generally feasible. The function `count.pairings`, which takes as arguments a treatment indicator and a potential blocking variable, will produce the number of treatment-control pairings within the proposed blocks.

In the RHC study, the patient's primary disease category is a good choice for a blocking variable. The authors of the original study deem this variable highly important to balance and focus their analysis on case comparisons within disease categories, so primary disease category satisfies the first criterion above. A call to `count.pairings` shows that blocking on primary disease category permits 1.96 million treatment-control pairings, which is similar to the number of treatment-control pairings in a dense match of 1000 treated units and 2000 controls. The imagined dense match would not cause any computational difficulties, so we can trust that the second criterion is satisfied as well.

3.3 Distance Structure

Once a sparsity approach has been chosen, pairwise distances must be computed for the allowable treated-control matches based on the observed covariates thought to be related to treatment and outcome. An excellent default option is the robust Mahalanobis distance described in section 8.3 of Rosenbaum (2009), which modifies the usual Mahalanobis distance by replacing each covariate with its ranks or average ranks (for tied observations), in order to reduce the influence of outlying observations, and by adjusting the resulting covariance matrix to have a constant diagonal in order to reduce the influence of heavily-tied covariates such as binary indicators. For the RHC data, we compute a robust Mahalanobis distance using secondary disease category, age, sex, and several types of overall health score (Duke Activity Score Index, APACHE score, 2-month survival probability estimated from a support model, Glasgow Coma score, and Activies of Daily Living scale).

Table 2: Counts for each primary disease category in the RHC data, broken down by treatment status (RHC = treated, No RHC = control). Since primary disease category is used as a blocking variable, each row summarizes the treated and control counts within a distinct block. Units will only be allowed to match to others within the same block. Notice that there are more treated patients with multiple organ system failure with sepsis than controls; to match within this block, we will need to exclude at least (700-527) treated units from the match. For a discussion of this phenomenon and its implications for the match, see Section 4.3.

	Treatment status	
Primary Disease Category	No RHC	RHC
Acute renal failure	1581	909
Chronic obstructive pulmonary disease	399	58
Congestive heart failure	247	209
Cirrhosis	175	49
Coma	341	95
Colon cancer	6	1
Lung cancer	34	5
Multiple organ system failure with malignancy	241	158
Multiple organ system failure with sepsis	527	700
Missing	0	0

We also add a caliper on the propensity score (via the arguments `calip.option` and `caliper`) which forbids matches between individuals whose propensity scores differ by more than a fifth of a standard deviation of the propensity score in the full sample. Using a caliper helps ensure paired individuals are similar in their propensity to be treated, but it also has computational benefits since it makes the match sparser. In the RHC example, adding a caliper to the within-blocks match reduces the number of treatment-control pairings by an order of magnitude, from 1.96 million to roughly 234,000.

Some covariates in the RHC data have missing values. Generally speaking, the `rcbalance` package follows the recommendations in section 9.4 of Rosenbaum (2009) on dealing with missing values. The goal is to condition on the fact of having observed that a value is missing, rather than making assumptions about the mechanism for missingness. As such missing values in categorical covariates are treated as a separate category when these covariates are used in balancing constraints and in computing the Mahalanobis distance. For missing values in continuous covariates used for the Mahalanobis distance, a new binary variable is added to indicate which observations are missing and then the missing values in the original vector are imputed using the mean of observed values.

In conventional dense matching problems, covariate distances can be represented by matrices, with each row corresponding to a treated unit and each column to a control and entries i, j) corresponding to the distance between treated unit i and control j. Distances built using the `rcbalance` package are instead represented as a list of vectors: each entry in the list corresponds to a treated unit, and for a given treated unit the vector stored in the corresponding index of the list has an entry for each potential control match with the corresponding pairwise distance. For very large, very sparse problems the list-of-vectors structure is helpful because it stores the relevant distance information far more efficiently than an ordinary `matrix` object in R can.

An alternative strategy for efficiently representing sparse distance structures is to use modified sparse matrix objects, such as the `InfinitySparseMatrix` objects created by certain commands in `optmatch`. `rcbalance` can also accept these objects as distance inputs, (as well as ordinary matrices with values of `Inf` representing forbidden treatment-control pairings).

To create a list-of-vectors containing distance information, the `build.dist.struct` command can be used:

```
> library(optmatch)
> library(rcbalance)

#read in dataframe rhc
> load(url("http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/rhc.sav"))

#define variables for Mahalanobis distance
> maha.vars <- c("cat2","age","sex","das2d3pc","aps1","surv2md1","scoma1",
       "adld3p")

#make distance structure
> my.dist <- build.dist.struct(z=rhc$swang1 == "RHC", X=rhc[maha.vars],
       exact = rhc$cat1, calip.option = "propensity", caliper = 0.2)
```

The final call to `build.dist.struct` gives the treatment variable `z`, the covariates `X` for the Mahalanobis distance, the blocking variable `exact`, and options indicating to include a caliper of 0.2 standard deviations on the propensity score.

3.4 Balance constraints

To define refined covariate balance constraints, categorical covariates can be divided into several tiers. The first tier should be composed of variables that are highly important and must be balanced closely for a credible analysis. The second tier should contain variables of slightly less importance and so on. Although refined covariate balance constraints can only be used with categorical covariates, important continuous covariates can be coarsened and included in these tiers (for example, one could create an indicator for patients above the age of 85 if age was deemed to be important). In the RHC example, we include secondary disease category and sex in the first tier. In the second tier we include binary variables formed by coarsening the 5 health scores included in the Mahalanobis distance (in each case, we set the variable equal to 1 if the risk score exceeds the median in the full sample). In the the third tier we include 13 comorbid conditions. Finally, in the fourth tier we include race and insurance status. Note that the most important variable, primary disease category, is not included since the blocked structure of the match already guarantees that it will be perfectly balanced. The resulting refined balance constraint has four levels: most important and coarsest, an interaction of all the variables in tier 1; second, an interaction of all the variables in tier 1 and tier 2; third, an interaction of all the variables in tiers 1 through 3; and finally, an interaction of all the variables in all four tiers. These constraints are included in the `fb.list` argument to the `rcbalance` command used to run the match as shown:

```
#coarsen continuous health scores
> rhc$highADL <- rhc$scoma1 > median(rhc$adld3p)
> rhc$highComa <- rhc$scoma1 > mean(rhc$scoma1)
> rhc$highApache <- rhc$aps1 > mean(rhc$aps1)
> rhc$highSurv <- rhc$surv2md1 > mean(rhc$surv2md1)
> rhc$highDAS <- rhc$das2d3pc > mean(rhc$das2d3pc)

#define fine balance levels
> l1 <- c("cat2","sex")
> l2 <- c(l1, "highADL", "highComa","highApache","highSurv","highDAS")
> l3 <- c(l2,"ca","cardiohx","chfhx","dementhx","psychhx","chrpulhx","renalhx",
       "liverhx","gibledhx","malighx","immunhx","transhx","amihx")
> l4 <- c(l3,"race","ninsclas")

#compute match
> match.out <- rcbalance(my.dist, fb.list = list(l1,l2,l3,l4),
       treated.info = rhc[rhc$swang1 == "RHC",],
       control.info = rhc[rhc$swang1 != "RHC",], exclude.treated = TRUE)
```

The `exclude.treated=TRUE` option used in this call to `rcbalance` is necessary since there is at least one natural block that contains more treated units than controls (see Table 2) and the match will not be feasible unless some treated units are excluded; see Section 4.3. The `match.out` object produced by the `rcbalance` command is a list with two entries. The first is a matrix of one column called "matches" that has row names indexing treated units and entries indexing paired controls. This object can be used to quickly extract the matched data from the full treated and control data frames passed into `rcbalance`. The second entry is a list called "fb.tables" which contains one entry for every level in the fine balance hierarchy. Each entry is a contingency table showing the treated and control counts for each of the different categories of the interaction at that level. Table 3 shows the first entry of fb.tables (associated with the coarsest, highest-priority level in the balance hierarchy) for this match.

3.5 Basic dense match

While `rcbalance` is designed to facilitate sparse matching with refined covariate balance, it can also perform more traditional forms of matching as special cases. In the RHC example, for instance, one might simply wish to compute Mahalanobis distances between all treated and control units based on all variables deemed important and perform optimal dense matching, allowing any treated unit to be matched to any control. This type of matching can be performed by the `optmatch` package in R, but it is also easy to do using `rcbalance`.

To do this we use the `build.dist.struct` command to compute Mahalanobis distances, but we do not specify the `exact` argument and we set the `calip.option` argument to "none" rather than "propensity." This computes distances between all treated and control units, not solely within natural blocks and calipers. Once we have computed the distance structure,

Table 3: Fine balance contingency table for highest-priority balance level in the match of Section 3.4, an interaction of secondary disease category and sex. Here exact balance has been achieved for all categories. The corresponding tables for subsequent, lower-priority balance levels do not have exact agreement between the two columns, but refined covariate balance guarantees that no other match can achieve closer balance on their sets of categories as a whole (where balance is measured by total variation distance).

Interaction Category	Controls	Treated
Cirrhosis.Female	4	4
Cirrhosis.Male	7	7
Colon Cancer.Female	1	1
Coma.Female	8	8
Coma.Male	12	12
Lung Cancer.Female	1	1
Lung Cancer.Male	1	1
MOSF w/Malignancy.Female	22	22
MOSF w/Malignancy.Male	36	36
MOSF w/Sepsis.Female	165	165
MOSF w/Sepsis.Male	203	203
NA.Female	637	637
NA.Male	843	843

it can be passed into the `rcbalance` directly with no additional arguments needed, since no fine balance constraints are used and all treated units can be retained in the match.

```
#define variables for Mahalanobis distance (include all relevant variables)
> maha.vars2 <- c("cat1","cat2","age""sex","adld3p","das2d3pc","aps1","surv2md1",
       "scoma1","ca","cardiohx","chfhx","dementhx","psychhx","chrpulhx",
       "renalhx","liverhx","gibledhx","malighx","immunhx","transhx","amihx",
       "race","ninsclas")

#make distance structure
> my.dist2 <- build.dist.struct(z=rhc$swang1 == "RHC", X=rhc[maha.vars2],
       calip.option = "none")

#compute match
> match.out2 <- rcbalance(my.dist2)
```

The call to `rcbalance` to compute this match was much more computationally intensive than the sparse matching call described in the previous section; it required 25-30 times as much computation time.

3.6 Balance assessment

The standardized differences on covariates for the unmatched data and for both matches are shown in Table 4. Both matches offer improvement over the unmatched comparison. While the dense match is better at balancing certain covariates in the lowest two sections of the table (which correspond to lower-priority sets of covariates), the sparse match achieves almost perfect balance on the first two tiers of variables, offering substantial improvement over the dense match in balancing several primary and secondary disease categories. While three covariates in the dense match have absolute standardized differences of 0.20 or greater, none of the covariates in the sparse match do. Evidently, conducting matching sparsely within natural blocks need not harm overall balance in matching but can improve it significantly, with refined covariate balance constraints ensuring that the most important variables receive the closest balance. Furthermore, the sparse match ran much faster than the dense match.

In this case the sparse match produced satisfactory balance immediately. If some standardized differences are not deemed low enough after an initial sparse match, however, altering the refined covariate balance hierarchy to place greater emphasis on the inadequately-balanced variables can often induce better balance. This type of sequential refitting of the match does not introduce inferential problems as variable selection and other sequential refitting techniques do in regression analysis, since matching is conducted without reference to outcome measurements.

Table 4: Absolute standardized differences for important covariates in raw unmatched data, after the dense match in Section 3.5, and after the sparse match in Section 3.4. The four sections of the table sort the variables into five rough categories of decreasing importance. The first section contains the primary disease categories (used for natural blocks in the sparse match) and the other four sections correspond closely to to the four levels of refined covariate balance described in Section 3.4. Covariate levels listed in Table 1 that are not present here were empty in both treated and control groups.

Covariate	Unmatched	Dense Match	Sparse Match
Primary disease category			
Acute renal failure	-0.06	-0.12	0.00
Chronic obstructive pulmonary disease	-0.34	-0.13	0.00
Congestive heart failure	0.10	0.07	0.00
Cirrhosis	-0.14	-0.05	0.00
Coma	-0.21	-0.13	0.00
Colon Cancer	-0.04	-0.01	0.00
Lung Cancer	-0.09	-0.04	0.00
Multiple organ system failure w/ malignancy	0.02	-0.01	0.00
Multiple organ system failure w/ sepsis	0.41	0.30	0.00
Secondary disease category			
Cirrhosis	-0.10	-0.08	0.00
Coma	-0.25	-0.12	0.00
Colon Cancer	0.01	-0.01	0.00
Lung Cancer	-0.14	0.01	0.00
Multiple organ system failure w/ malignancy	-0.36	-0.19	0.00
Multiple organ system failure w/ sepsis	0.52	0.26	0.00
Sex			
Female	-0.09	-0.02	0.00
Male	0.09	0.02	0.00
Age	-0.06	-0.04	-0.02
Activities of Daily Living scale	-0.13	0.01	-0.11
Duke Activity Score index	0.06	-0.01	0.04
APACHE score	0.50	0.37	0.05
Estimated 2-month survival probability	-0.20	-0.15	-0.09
Glasgow Coma score	-0.11	-0.07	0.02
Cancer status			
No cancer	0.10	0.02	0.08
Cancer	-0.07	-0.02	-0.09
Metastatic cancer	-0.07	0.00	-0.00
Cardiovascular disease	0.12	0.09	0.05
Congestive heart failure	0.07	0.11	0.03
Dementia	-0.16	-0.06	-0.13
Psychiatric history	-0.14	-0.03	-0.10
Chronic pulmonary disease	-0.19	-0.03	-0.02
Chronic renal disease	0.03	0.07	-0.03
Liver disease	-0.05	0.00	0.01
Upper GI bleeding	-0.07	0.00	0.00
Solid Tumor/Metastatic Disease/Leukemia	-0.10	-0.02	-0.09
Immunosuppression	0.08	0.12	0.02
Transfer from another hospital	0.17	0.18	0.18
Myocardial infarction	0.07	0.13	0.10
Race			
White	0.02	-0.05	0.01
Black	-0.03	0.02	-0.02
Other	0.02	0.06	0.01
Insurance class			
No insurance	0.04	0.04	0.04
Private	0.14	0.03	0.12
Medicare	-0.08	-0.04	-0.08
Medicaid	-0.13	-0.03	-0.10
Private & Medicare	0.03	-0.01	0.03
Medicare & Medicaid	-0.06	0.03	-0.04

4 Additional features

4.1 Near-exact matching

In the RHC example, matches are computed within natural blocks formed by the primary disease categories. In other words, we match patients exactly on primary disease category, and exclude treated patients who could not be matched under this requirement. However, in some settings researchers might wish to conduct near-exact matching, where matched treated and control subjects are required to agree on a covariate wherever possible, but treated units are not be excluded in cases where exact matching is impossible. For a more thorough description of near-exact matching and an example, see Zubizarreta et al. (2011).

Near-exact matching on a variable can be conducted using `rcbalance` by passing a variable name (or, if near-exact matching is desired on an interaction of several variables, a character vector of variable names) to the `near.exact` argument in the `rcbalance` function. Near-exact matching takes precedence over refined covariate balance if refined covariate balance constraints are also present. For example, if we were to redo the RHC match with near-exact matching on sex and refined covariate balance on secondary disease category, the matched groups would have the closest possible fine balance among secondary disease categories, subject to the requirement that patients of the same sex be matched to one another wherever possible. In general then, variables used for near-exact matching should be at least as important as those in the coarsest and most important layer of the refined covariate balance hierarchy.

4.2 Fixed ratio matching

If the control group is very large relative to the treated group, it may be desirable to match multiple controls to each treated unit (Rosenbaum, 2009). To conduct $1 : k$ matching in R (with k a nonnegative integer), one may use the `rcbalance` command and specify the desired ratio via the `k` argument.

When analyzing outcomes from such a match, controls are given weights of $1/k$. When assessing balance, this same weighting is used. For example, suppose we conduct 1:k matching with refined covariate balance on sex. Then if there are m males in the treated group, the match will seek to find exactly km males in the control group so that the weighted sum of male indicators within the control group will equal the sum in the treated group. For a more formal description of balance in fixed-ratio settings, see (Pimentel et al., 2015c). The costs and benefits of fixed-ratio matching, relative to pair matching, are described in Hansen (2004) and Pimentel et al. (2015a) offers some practical guidance about how to choose k.

4.3 Optimal exclusion of treated units

In some cases it may not be possible to match all treated units to a single control. In dense matching this occurs only when there are more treated units present in the data than there are controls. In sparse matching it may happen in many other cases as well. When matching within blocks for instance, it happens whenever any individual block contains more treated units than controls as occurs in the RHC example discussed above (see Table

2). Generally speaking,the cases in which it is not possible to match all treated units are characterized by Hall's theorem see (see Diestel, 2010, p. 38), and they are most easily detected by attempting to form a pair match. When `rcbalance` is run under default settings (`exclude.treated=FALSE`) for such a matching problem, it will return an error indicating that the match is not feasible.

One reasonable response to infeasibility is to reduce the sparsity of the network so that all treated units can be used; one might do this by matching within larger, coarser blocks or by widening a caliper. Another possibility is to exclude some treated units from the match. This can be done based on heuristics such as excluding blocks. If the problem is due to a lack of common support between the treated and control covariate distributions, the methods in Crump et al. (2009) and Traskin and Small (2011) may also be helpful, although they do not guarantee that the match will become feasible.

A different approach to excluding treated units comes from the idea of optimal subset matching (Rosenbaum, 2012). Optimal subset matching makes simultaneous decisions about how many treated units to match, which specific treated units to match, and which controls to match them to. The tradeoff between including more treated units and improving the average pairwise distance in the match by excluding less easily matchable treated units is managed by a tuning parameter $\widetilde{\delta}$; loosely speaking, treated units are excluded if doing so can decrease the sum of pairwise match distances by at least $\widetilde{\delta}$. The `rcbalance` package implements a special case of optimal subset matching in which the parameter $\widetilde{\delta}$ is set to be very large, larger than any of the pairwise distances or fine balance penalties in the match. This tells the algorithm that we wish to include as many treated units as possible and permits exclusion of treated units only when the match is infeasible.

This approach to excluding treated units is desirable for several reasons. First, unlike the other methods for excluding treated units, it is both fully automated and gives a guarantee that a feasible match will be produced, removing the need for researchers to pick through a potentially complex sparse configuration of allowed treated-control edges to diagnose an infeasibility manually. Second, since the problem of which individuals to exclude is solved concurrently with the match as part of an optimization problem, it always includes the maximal number of treated units possible among feasible matches and (conditional on this) excludes the poorest-matching units in terms of fine balance and pairwise distances. This is the "optimality" referred to in the name of the general method.

When using the `rcbalance` command, automated optimal exclusion of treated units will be used if the `exclude.treated` argument is set to `TRUE` (default is `FALSE`). `rcbalance` will exclude only the minimal number of treated units in order to achieve feasibility, so if a match is feasible the `exclude.treated` argument makes no difference in the results.

5 Target Distributions

Everything discussed so far assumes that the researcher's goal in matching is to make the covariate distribution of the selected control group as similar as possible to the covariate distribution of the treated group. However in some cases it may be desirable to select controls with a covariate distribution that differs in some way from the treated distribution. For example, (Pimentel et al., 2015b) describes how unmeasured biases in observational studies may be attenuated if treated and control distributions differ on certain "innocuous" covariates. Another situation where differences in treated and control distributions might be desirable is when a number of smaller matches are being performed and the researchers desire to make each control group similar to a single reference distribution. This might arise in entire number matching, where controls are selected using separate matches within entire number strata, but the goal is to produce a single overall weighted control distribution that is similar to the overall treated distribution, rather than making each individual-stratum control group similar to its stratum-specific counterpart (Pimentel et al., 2015c).

We describe matching under such goals as matching for balance with respect to a desired control distribution (as opposed to the default, which is matching for balance with respect to the treated distribution). Refined covariate balance provides a natural way to formalize matching with respect to a distribution and produce an optimal match under that constraint. Technical details of how the algorithm of Pimentel et al. (2015a) can be modified to produce balance with respect to another distribution are described in the online appendix to (Pimentel et al., 2015b). This appendix also offers practical suggestions about how to construct a new target distribution in one case where that might be desirable.

Using `rcbalance` to match for balance with respect to a desired control distribution distinct from the treated distribution is straightforward. Suppose we have a problem where the treated group has size T. To specify a desired control distribution, create a sample of T observations from the desired distribution and store it in a matrix or a data frame. Passing this object to the argument `target.group` in the `rcbalance` instructs the algorithm to use the empirical covariate distribution in the sample as the target for balance. Note that the `target.group` matrix must contain the same column count and names as the treated.info and `control.info` arguments.

Acknowledgments

Many thanks to Dylan Small, Paul Rosenbaum, and Zijian Guo for helpful comments. Supported by Fellowship FA9550-11-C-0028 from the Department of Defense, Army Research Office, National Defense Science and Engineering Graduate (NDSEG) Fellowship Program, 32 CFR 168a4.

References

Baiocchi, M., Small, D. S., Yang, L., Polsky, D., and Groeneveld, P. W. (2012). Near/-far matching: a study design approach to instrumental variables. *Health Services and Outcomes Research Methodology*, 12(4):237–253.

Beck, C., Lu, B., and Greevy, R. (2015). *nbpMatching: Functions for Optimal Non-Bipartite Matching*. R package version 1.4.5. Available from: `http://CRAN.R-project.org/package=nbpMatching`.

Bertsekas, D. P. (1991). *Linear network optimization: algorithms and codes*. MIT Press.

Bertsekas, D. P., Tseng, P., et al. (1994). *RELAX-IV: A faster version of the RELAX code for solving minimum cost flow problems*. Citeseer.

Connors, A. F., Speroff, T., Dawson, N. V., Thomas, C., Harrell, F. E., Wagner, D., Desbiens, N., Goldman, L., Wu, A. W., Califf, R. M., et al. (1996). The effectiveness of right heart catheterization in the initial care of critically iii patients. *Jama*, 276(11):889–897.

Crump, R. K., Hotz, V. J., Imbens, G. W., and Mitnik, O. A. (2009). Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, page asn055.

Diestel, R. (2010). *Graph Theory*. Springer.

Hansen, B. B. (2004). Full matching in an observational study of coaching for the sat. *Journal of the American Statistical Association*, 99(467):609–618.

Hansen, B. B. and Klopfer, S. O. (2006). Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627.

Ho, D. E., Imai, K., King, G., and Stuart, E. A. (2011). MatchIt: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software*, 42(8):1–28. Available from: `http://www.jstatsoft.org/v42/i08/`.

Lu, B., Greevy, R., Xu, X., and Beck, C. (2011). Optimal nonbipartite matching and its statistical applications. *The American Statistician*, 65(1).

Neuman, M. D., Rosenbaum, P. R., Ludwig, J. M., Zubizarreta, J. R., and Silber, J. H. (2014). Anesthesia technique, mortality, and length of stay after hip fracture surgery. *JAMA*, 311(24):2508–2517.

Pimentel, S. D., Kelz, R. R., Silber, J. H., and Rosenbaum, P. R. (2015a). Large, sparse optimal matching with refined covariate balance in an observational study of the health outcomes produced by new surgeons. *Journal of the American Statistical Association*, 110:115–127.

Pimentel, S. D., Small, D. S., and Rosenbaum, P. R. (2015b). Constructed second control groups and attenuation of unmeasured bias. Journal of the American Statistical Association, in press.

Pimentel, S. D., Yoon, F., and Keele, L. (2015c). Variable ratio matching with fine balance in a study of the peer health exchange. Statistics in Medicine, in press.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Available from: `http://www.R-project.org/`.

Rosenbaum, P. R. (1989). Optimal matching for observational studies. *Journal of the American Statistical Association*, 84(408):1024–1032.

Rosenbaum, P. R. (2009). *Design of Observational Studies*. Springer Science & Business Media.

Rosenbaum, P. R. (2012). Optimal matching of an optimally chosen subset in observational studies. *Journal of Computational and Graphical Statistics*, 21(1):57–71.

Rosenbaum, P. R. (2015). Two r packages for sensitivity analysis in observational studies. *Observational Studies*, 1(1).

Rosenbaum, P. R., Ross, R. N., and Silber, J. H. (2007). Minimum distance matched sampling with fine balance in an observational study of treatment for ovarian cancer. *Journal of the American Statistical Association*, 102(477):75–83.

Rosenbaum, P. R. and Rubin, D. B. (1985). Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38.

Schrijver, A. (2003). *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media.

Sekhon, J. S. (2011). Multivariate and propensity score matching software with automated balance optimization: The Matching package for R. *Journal of Statistical Software*, 42(7):1–52. Available from: `http://www.jstatsoft.org/v42/i07/`.

Silber, J. H., Rosenbaum, P. R., Clark, A. S., Giantonio, B. J., Ross, R. N., Teng, Y., Wang, M., Niknam, B. A., Ludwig, J. M., Wang, W., et al. (2013). Characteristics associated with differences in survival among black and white women with breast cancer. *JAMA*, 310(4):389–397.

Silber, J. H., Rosenbaum, P. R., Ross, R. N., Ludwig, J. M., Wang, W., Niknam, B. A., Mukherjee, N., Saynisch, P. A., Even-Shoshan, O., Kelz, R. R., et al. (2014). Template matching for auditing hospital cost and quality. *Health services research*, 49(5):1446–1474.

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

Traskin, M. and Small, D. S. (2011). Defining the study population for an observational study to ensure sufficient overlap: a tree approach. *Statistics in Biosciences*, 3(1):94–118.

Yang, D., Small, D. S., Silber, J. H., and Rosenbaum, P. R. (2012). Optimal matching with minimal deviation from fine balance in a study of obesity and surgical outcomes. *Biometrics*, 68(2):628–636.

Zubizarreta, J. R. (2012). Using mixed integer programming for matching in an observational study of kidney failure after surgery. *Journal of the American Statistical Association*, 107(500):1360–1371.

Zubizarreta, J. R., Reinke, C. E., Kelz, R. R., Silber, J. H., and Rosenbaum, P. R. (2011). Matching for several sparse nominal variables in a case-control study of readmission following surgery. *The American Statistician*, 65(4).