

Imposing a bottleneck constraint to improve the performance of statistical
matching

by

Sanjeewani Weerasingha

B.S., Rajarata University of Sri Lanka, 2010

M.S., Sam Houston State University, 2016

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

Consider an observational study where a researcher aims to assess the effect of a treatment on a response. Estimation of this effect may be complicated by the presence of confounders—variables that are correlated both with the treatment and the response. Failure to account for confounders may lead to large biases in the estimation of treatment effects. Statistical matching, in which each treated unit is paired with a control unit with similar covariate values, is a common way to account these confounders.

In its most basic form, statistical matching pairs each treated unit with a control unit with “similar” values on confounding covariates. Estimates of treatment effects are then obtained by taking the average differences between these pairs. Common measures of the similarity of confounding covariates between treated and control units include the Euclidean distance, the Mahalanobis distance, and the absolute difference in propensity scores. Most matching methods aim to minimize the total dissimilarity between treated and matched control units. However, this may open the door for a matching with significant imbalances on certain covariates within a small proportion of units. Hence, a researcher may also desire to limit the maximum imbalance possible within a matched pair. Additionally, when matching is performed without replacement—that is, when each control unit can be used in a match at most one time—many existing algorithms for finding an optimal matching scale poorly to large datasets.

In this dissertation, we formulate statistical matching as a graph theory problem, where nodes are units, edges are drawn between potential matches, and each edge has a weight measuring the dissimilarity between the corresponding match. We argue that the computational complexity of a problem can be substantially reduced by sparsening the original matching graph—that is, by removing edges until the number of edges is proportional to the

number of vertices. Further, we outline how algorithms can be constructed for efficiently solving matching problems on these constrained sparse graphs.

Further, we show that matching with a constraint on the maximum imbalance on a covariate is equivalent to finding a match within a bottleneck subgraph, in which edges in the graph are drawn if and only if their weight is less than some pre-specified threshold. Thus, the addition of this bottleneck constraint serves two purposes. First, it prevents imbalance on covariates within matches, thereby reducing potential bias in treatment effect estimates. Second, because the matching subgraph is sparse, optimally matching within this subgraph can substantially decrease computational complexity. We apply our method to a study on the efficacy of right heart catheterization. Results indicate that implementing bottleneck constraints can efficiently obtain a statistical matching. The matching successfully eliminate poor matches from the sample and may yield strong overall balance on confounding covariates.

Finally, we propose a new algorithm for the minimum cost-optimal pair matching problem by using negative cycles and negative alternating paths of even length (NAPELs). This approach allows for solving the minimum cost matching problem without having to introduce additional structure to the matching graph. We prove that our algorithm will find a minimum cost matching. Additionally, we give simulation results that confirm the optimality of our algorithm, and suggest methods for improving its runtime.

Imposing a bottleneck constraint to improve the performance of statistical
matching

by

Sanjeevani Weerasingha

B.S., Rajarata University of Sri Lanka, 2010

M.S., Sam Houston State University, 2016

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics
Collage of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
Michael Higgins

Copyright

© Sanjeevani Weerasingha 2022.

Abstract

Consider an observational study where a researcher aims to assess the effect of a treatment on a response. Estimation of this effect may be complicated by the presence of confounders—variables that are correlated both with the treatment and the response. Failure to account for confounders may lead to large biases in the estimation of treatment effects. Statistical matching, in which each treated unit is paired with a control unit with similar covariate values, is a common way to account these confounders.

In its most basic form, statistical matching pairs each treated unit with a control unit with “similar” values on confounding covariates. Estimates of treatment effects are then obtained by taking the average differences between these pairs. Common measures of the similarity of confounding covariates between treated and control units include the Euclidean distance, the Mahalanobis distance, and the absolute difference in propensity scores. Most matching methods aim to minimize the total dissimilarity between treated and matched control units. However, this may open the door for a matching with significant imbalances on certain covariates within a small proportion of units. Hence, a researcher may also desire to limit the maximum imbalance possible within a matched pair. Additionally, when matching is performed without replacement—that is, when each control unit can be used in a match at most one time—many existing algorithms for finding an optimal matching scale poorly to large datasets.

In this dissertation, we formulate statistical matching as a graph theory problem, where nodes are units, edges are drawn between potential matches, and each edge has a weight measuring the dissimilarity between the corresponding match. We argue that the computational complexity of a problem can be substantially reduced by sparsening the original matching graph—that is, by removing edges until the number of edges is proportional to the

number of vertices. Further, we outline how algorithms can be constructed for efficiently solving matching problems on these constrained sparse graphs.

Further, we show that matching with a constraint on the maximum imbalance on a covariate is equivalent to finding a match within a bottleneck subgraph, in which edges in the graph are drawn if and only if their weight is less than some pre-specified threshold. Thus, the addition of this bottleneck constraint serves two purposes. First, it prevents imbalance on covariates within matches, thereby reducing potential bias in treatment effect estimates. Second, because the matching subgraph is sparse, optimally matching within this subgraph can substantially decrease computational complexity. We apply our method to a study on the efficacy of right heart catheterization. Results indicate that implementing bottleneck constraints can efficiently obtain a statistical matching. The matching successfully eliminate poor matches from the sample and may yield strong overall balance on confounding covariates.

Finally, we propose a new algorithm for the minimum cost-optimal pair matching problem by using negative cycles and negative alternating paths of even length (NAPELs). This approach allows for solving the minimum cost matching problem without having to introduce additional structure to the matching graph. We prove that our algorithm will find a minimum cost matching. Additionally, we give simulation results that confirm the optimality of our algorithm, and suggest methods for improving its runtime.

Table of Contents

List of Figures	xi
List of Tables	xiii
Acknowledgements	xiv
Dedication	xv
1 Introduction	1
1.1 Problem Statement	1
1.2 Neyman-Rubin Causal Model with Assumptions	3
1.2.1 Parameter of Interest:	5
1.2.2 The Selection On observables/Conditional Independence/Ignorability Assumption	6
1.2.3 Assumption of Common Support	6
1.3 Dissimilarity	8
1.4 Matching Methods	9
1.5 Matching and graph theory	11
1.6 Contribution	11
2 Demystifying Statistical Matching Algorithms for Big Data	13
2.1 Introduction	13
2.2 Problem Setup for Statistical Matching	15
2.3 Greedy Matching	17
2.4 Statistical Matching with Replacement	19

2.5	Optimal Statistical Matching Without Replacement	20
2.5.1	Maximum Cardinality Matching	21
2.5.2	Minimum Cost Matching	26
2.6	Scaling down data in statistical matching	30
2.6.1	Complete, dense, and sparse graphs	30
2.6.2	Adjacency list and adjacency matrix	30
2.7	Matching in a Sparse Graph	32
2.7.1	An overview of existing methods	32
2.7.2	A discussion about existing software tools	34
2.8	Conclusion	37
3	Preventing poorly-matched pairs using bottleneck subgraph matching	38
3.1	Introduction	38
3.2	Motivation	40
3.2.1	Prevent poor individual matches	40
3.2.2	Reducing number of potential matches	45
3.2.3	Researcher-oriented covariate balance/ Inflating and deflating the caliper	45
3.3	Related work	45
3.4	Notation and preliminaries	47
3.4.1	Largest Caliper Dissimilarity	48
3.4.2	Model of Response: Neyman-Rubin Causal Model	48
3.5	Bottleneck subgraph matching with largest caliper dissimilarity	49
3.5.1	BSM as a linear integer program	49
3.5.2	Graph theory interpretation of BSM	51
3.5.3	Software tools for solving BSM	52
3.6	Illustration with data example	52
3.6.1	Results for LCD	53
3.6.2	Inflating the LCD to increase sample size	58

3.7	Conclusion	60
4	Negative Alternating Paths of Even Length for the Minimum Weight-Maximum Cardinality Matching Problem	61
4.1	Introduction	61
4.2	Maximum matching (MaxCard)	62
4.3	Minimum weight (MinCost)	63
4.3.1	Common approach	63
4.3.2	The proposed method	64
4.4	Preliminaries and notations for the proposed method	65
4.5	Algorithm with NAPELs to solve Minimum weight-Maximum cardinality pair matching in a Bottleneck subgraph	69
4.6	Simulation study	70
4.7	Conclusion	70
5	Conclusion and Future Work	72
5.1	Conclusion	72
5.2	Future Work	74
	Bibliography	75
A	Cardinality matching followed by minimum distance pairing with Bottleneck con- straint	82
B	Simulation study for optimal pair matching problem using Ford Fulkerson and cycle cancelling algorithms	85
C	Detail description for RHC Data	88

List of Figures

1.1	The causal relationship between smoking, heart disease, and exercise habits is the confounding variable that influences both smoking and heart disease. .	2
2.1	An example of transforming an observational study into a bipartite graph with distance as dissimilarities	16
2.2	(a). A bipartite graph with five treated and control units. (b). The network flow graph for (a).	23
2.3	(a). The flow network G and initial flow f with (capacity, flow). (b) The residual graph for (a) with augmenting path p in blue and residual capacity (δ); Consider the reverse-path $C_2 - T_1$ and selecting path $s - T_2 - C_2 - T_1 - C_1 - t$. (c). The flow in G that results from augmenting along path p by its residual capacity. (d). The residual network induced by the flow in (c); no path can be found $s - t$ with all edges those with $\delta = 1$	26
2.4	(a). The original flow network G with initial flow f with dissimilarities w . (b). The residual graph for (a) with augmenting path p color in blue; consider the reverse-paths $c_1 - T_1$ and $c_2 - T_2$, selecting path $c_1 - T_1 - C_2 - T_2 - C_1$ with $-w_{11} + w_{12} - w_{22} + w_{21} < 0$ negative cost. (c). The flow in G that results from augmenting along path p	30
2.5	Figure shows the sparse graph of bipartite graph shown in Figure 2.2 (a) . .	31
2.6	Adjacency matrix and weight matrix	31
3.1	Bipartite graph without MWPD	43
3.2	Bipartite graphs MWPD=0.2, and MWPD=0.1	44
3.3	Number of treatment units in the matched sample for different caliper values	54

3.4	Standardized mean difference before and after matching	56
3.5	Standardized mean difference before and after matching	57
3.6	Box plot for APACHE score (aps1) variable before and after matching	57
3.7	Box plot for Age variable before and after matching	58
3.8	Covariate balance for three variables for different caliper values	59
3.9	Covariate balance for three variables with different user-defined MRPD values	60
4.1	(a). Bipartite graph with feasible maximum cardinality pair matching. (b). The pair matching is infeasible for all treatment units	64
4.2	(a). Bipartite digraph with the initial matching. (b). The residual bipartite graph for graph (a). In Figure (b), if $-w_{11} + w_{12} - w_{22} + w_{21} < 0$, $C_1 - T_1 -$ $C_2 - T_2 - C_1$ cycle is a negative cycle. In Figure (b), if $-w_{11} + w_{13} < 0$, $C_1 - T_1 - C_3$ path is a negative alternative path with even-length	66
4.3	(a). Bipartite graph with two matching sets M and M' . (b). The symmetric difference of two matching sets $M \triangle M'$	66
4.4	(a). Alternating cycle start from a matched control unit and ended up with the same control unit. (b). Alternating even path start from a matched control unit and ended up with an unmatched control unit	67

List of Tables

1.1	Causal inference as a missing data problem	5
3.1	Dissimilarity values between treated and control units.	41
3.2	Dissimilarity measures between treated and control units with maximum dis- similarity threshold value of 1.5.	41
3.3	Propensity score values for three treatment and control units.	42
3.4	Absolute difference of dissimilarity between treatment and control units. . .	42
3.5	Optimal four solutions without any maximum dissimilarity threshold value .	43
3.6	Absolute difference of dissimilarity measures between treated and control units for $MWPD \leq 0.2$	44
3.7	Absolute difference of dissimilarity measures between treated and control units for $MWPD \leq 0.1$	45
3.8	Covariate balance(SMD) for two matching samples with user specified MWPD	54
3.9	Two sample t-test results for each covariate before matching (Raw Data, n=2184)	55
3.10	Paired t-test results for each covariate after matching with MWPD (n=1696)	55
3.11	Covariate balance(SMD) for different caliper with number of matched and unmatched treated units	59
4.1	Simulated results for optimal pair matching and the proposed new algorithm with NAPELs(Step 4 to Step 8)	70
B.1	Simulated results for optimal pair matching(PM) and minimum weight-maximum cardinality method(MWMC).	86

Acknowledgments

I would like to express my wholehearted gratitude to Dr. Michael Higgins, my major professor, for all his knowledge, guidance, and many helpful suggestions. Completing this study could not have been possible without his encouragement and motivation throughout my Ph.D. studies. I especially thank Dr. Christopher I. Vahl, Dr. Pietro Poggi-Corradini, and Dr. Perla Reyes for serving on my committee and their valuable guidance throughout my study.

I would also like to thank the entire faculty and department staff and all the professors for their excellent courses and valuable knowledge. I will forever be thankful to Dr. James Neill and Dr. Gary Gadbury, who taught me and helped me during challenging times in my studies. I also thank all current and former students, of whom there are too many to mention, who influenced my statistical education.

I would like to thank my father, Jayantha Nimal Weerasinghe, and my mother, Bandula Malkanthi, for scarifying their lives to provide unconditional love and care for me. I also thank my sister and two brothers, who always supported me to succeed in my life. I am lucky to have you all in my life.

Thanks are due foremost to my husband, Dr. Steephanson Anthonymuthu, for being with me all the time on my side and directing me patiently in this journey. I would not have made it this far without you. Also, I would like to thank my lovely daughter, Sarah Sanujini Steephanson, for the patience she had when I was busy with my study. I also thank my friends (too many to list here) for providing the support and friendship I needed.

Dedication

I sincerely dedicate this work to my father, my mother, my husband, my sister and two brothers, and my daughter.

Chapter 1

Introduction

1.1 Problem Statement

Causation, the effect of a treatment on a response, is not correlation. Many statistical applications essentially focus on causation rather than correlation ([Imbens and Rubin, 2015](#)). Causal inference is a statistical analysis procedure that considers the assumptions, study designs, and estimation strategies that allow researchers to draw causal conclusions based on data.

Randomized experiments and observational studies aim to estimate the treatment effect on a response. For example, in a study to investigate the impact of smoking on heart disease, smoking will be considered the treatment, and heart disease is the response. Researchers prefer randomized experiments instead of observational studies because the random treatment allocation to experimental units in an experiment alleviates bias due to confounding variables (variables that correlate with treatment and response variables).

However, randomized experiments are not always possible; consider the example of smoking on heart disease. Random allocation for this example means the researcher must randomly assign somebody to smoke or not to smoke, which is impossible. Conducting an observational study is reasonable for such a situation, where the treatments are not randomly allocated to the experimental units. Because observational studies do not benefit

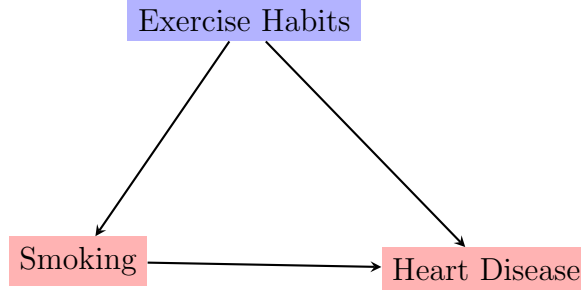


Figure 1.1: The causal relationship between smoking, heart disease, and exercise habits is the confounding variable that influences both smoking and heart disease.

from the random allocation of treatment to experimental units, the effect estimation has the risk of containing bias due to confounding variables.

Figure 1.1 shows a confounding variable where the treatment is smoking and the response is heart disease. The causal effect estimation requires isolating the causal relationship between smoking and heart disease from the confounding variable exercise habits. A commonly used regression model can isolate the causal relationship if and only if all confounding variables are in the model, and the relationships between confounders follow a linear relationship. However, the above conditions are hard to achieve in many real-world scenario.

By balancing the covariate distributions between treatment and control groups, matching can reduce the effect of confounding variables in treatment effect estimation. Statistical matching pairs statistically significantly similar control and treatment units. Hence, matching can create similar multivariate distributions across treatment and control groups for confounders.

In other words, matching is a statistical technique to reconstruct randomness in an observational study as achievable as with observed covariates. The treatment effect may then be estimated by the average treatment effect (ATE) of the matched sample.

Because a well-matched sample can minimize the bias in treatment effect estimation, these approaches are common in fields of economics, political science, and statistical quality control. This study provides a new strategy to apply in any matching method that aims to reduce the imbalance of observed covariates. The proposed new method is minimum weight-maximum cardinality pair matching with bottleneck constraint. The proposed method aims

to find the maximum matching in the bottleneck subgraph with minimum total dissimilarity among treatment and control units.

The minimum weight-maximum cardinality pair matching with bottleneck constraints follows two steps. The method will first remove the set of pairs with dissimilarity greater than the maximum dissimilarity threshold value. And second, any optimal pair matching method may then be applied to optimize the minimum weight in the bottleneck subgraph.

1.2 Neyman-Rubin Causal Model with Assumptions

Consider a study where every unit is given either a treatment condition or a control condition, and a set of treatment conditions.

$$T \in \{T_1, T_2, \dots, T_k\}$$

Suppose a single unit differs if and only if the treatment condition has changed. Then we call the change in y caused by the change in treatment conditions.

Let

$y_i(T_j)$ = state of object i under treatment j and

a causal effect denoted by,

$$\delta_i^{jk} = y_i(T_j) - y_i(T_k).$$

The above expression says that a change in dependent variable will occur only when there is a change in treatment condition.

In an observational study, typically there are two treatment conditions treated and control.

$$T \in \{T, C\}.$$

Then the treatment effect for unit i denoted by δ_i can be defined as,

$$\delta_i = y_i(T) - y_i(C)$$

All these effects are defined in terms of individuals. Therefore, it is important to know about how an individual is different if it goes from treatment to control or vice versa. However, the fundamental problem of this calculation is that it is impossible to assign these treatment and control conditions to one unit at the same time to obtain the treatment effect specific to each unit.

To address the fundamental problem in a study where it is impossible to find the unit-wise treatment effect, the Rubin Causal model is defined as a missing data problem in which we see only half of the data ([Holland and Rubin, 1987](#)).

Let the treatment indicator be

$$T \in \{1, 0\},$$

and

$y_i(T) = y_{i1}$ = the response of unit i if it received the treatment condition,

$y_i(C) = y_{i0}$ = the response of unit i if it received the control condition.

We assume that the response of unit i follows the Neyman-Rubin Causal Model (NRCM) of response ([Holland and Rubin, 1987](#); [Rubin, 1974](#); [Splawa-Neyman et al., 1990](#)):

$$Y_i = y_{i1}T_i + y_{i0}(1 - T_i) \tag{1.1}$$

where T_i is a treatment indicator variable,

$$T_i = \begin{cases} 1, & \text{unit } i \text{ is treated,} \\ 0, & \text{otherwise.} \end{cases}$$

and y_{i1}, y_{i0} are the potential outcomes.

According to this model, we may assume no spillover effects among these units. This assumption is called Stable Unit Treatment Value Assumption(SUTVA) ([Rosenbaum et al., 2010](#)).

Definition 1. *The Stable Unit Treatment Value Assumption or SUTVA:*

The response for a unit only depends on whether or not that unit receives treatment, and

does not depend on the treatment assignment of any other unit.

Unit	Full data		observed data	
	Treatment	Control	Treatment	Control
i	y_{i1}	y_{i0}	y_{i1}	-
j	y_{j1}	y_{j0}	-	y_{j0}

Table 1.1: Causal inference as a missing data problem

According to Table 1.1, researchers prefer complete data. However, if unit i receives the treatment (y_{i1} – *observe*), then the same unit i can not receive the control condition (y_{i0} – *missing*) and vice versa.

Consider two units indexed by i, j where unit i receives the treatment and unit j receives the control. Then the estimation of the average treatment effect can be calculated as follows:

$$\delta = \frac{1}{2}[(y_{i1} + y_{j1}) - (y_{i0} + y_{j0})]$$

Because, at most, a single potential outcome is available for any unit, the evaluation of the treatment effect relies on multiple units.

When multiple units are used to learn about treatment effects, some units are subjected to the treatment condition, and some units subject to the control condition.

1.2.1 Parameter of Interest:

Instead of focusing on individual treatment effects, it is possible to focus on average treatment effects across populations. The most prominent two parameters are average treatment effect (ATE) and average treatment effect for the treated (ATT).

ATE measures the effect of treatment on the entire population:

$$\delta_{ATE} = E[y_{i1} - y_{i0}].$$

ATT measures the effect of treatment on treated sub population.

$$\delta_{ATT} = E[y_{i1} - y_{i0} | T_i = 1].$$

With the assumption that all covariates are evenly distributed between treated and control in expectation, ATE and ATT are equal for randomized experiments. Because treatment assignment is unevenly distributed in an observational study, ATE may bias the treatment effect estimation. Thus, ATT is used for observational studies to estimate the treatment effect.

Estimates of average treatment effects are valid if and only if the observations can meet a set of fundamental assumptions as well as the SUTVA.

1.2.2 The Selection On observables/Conditional Independence/Ignorability Assumption

This assumption says the conditional on observable covariates X , where X is a set of confounding covariates, the assignment of units to experimental groups is independent of potential outcomes.

$$(y_{i1}, y_{i0}) \perp T | X.$$

Here, the ignorability assumption is a more general condition compared to the random assignment of treatment conditions in a randomized experiment.

1.2.3 Assumption of Common Support

Causal inference in observational studies involves making comparisons between groups that are dissimilar. Common support is an assumption of covariate overlap between treated and control units across the population of interest to make the comparison. Therefore, the causal inference strategies in observational studies typically require an assumption of common support.

For binary treatment assignment, T , and vector of confounding covariates, X , it is assumed that

$$0 < Pr(T = 1|X) < 1.$$

To satisfy the common support assumption, researchers have used various strategies to identify observations in neighborhoods of the covariate space.

The combination of the ignorability assumption and the common support assumption will result in the strong ignorability assumption.

When estimating causal effects, an approach to be consider is blocking confounding variables by obtaining treated and control groups with similar covariate distributions. This similarity can be achieved by choosing a well-matched sample of the original treated and control groups.

Estimation of the ATE can also be performed using linear regression, but the linear regression approach must satisfy many assumptions to make the inference. Three sets of assumptions should be satisfied in a causal model: i) assumption about the model, ii) assumption about the variables, and iii) statistical assumptions [Yetley \(1969\)](#). The randomness in treatment and control units can easily accommodate those assumptions, but the random assignments can not justify strong assumptions like independence, common distributions, and linearity of response ([Freedman, 2009](#)).

Moreover, checking covariate balance and the degree of overlap in the causal analysis are significant factors in an observational study. The covariate balance gives an idea about whether the treated and control groups overlap sufficiently to make the treatment effect comparison. The departure from comparability occurs if there are regions in the covariate space where there are some treated units but no controls units, or vice versa. This situation is called a lack of overlap. If covariate distributions of treatment and control groups do not overlap, the treatment effect comparison will automatically limit to the overlap region.

There is a departure from comparability, imbalance, lack of balance, or unbalance if the distributions of relevant pre-treatment variables significantly differ for the treated and control groups. Moreover, if there is a difference among averages, skewness, or standard

deviations, the imbalance exists. The simple comparisons of group averages are not a good estimation when treatment and control groups are unbalanced.

1.3 Dissimilarity

Matching aims to pair each treatment unit with the most similar control unit based on pretreatment covariates. The best way is to match each treatment unit to a control unit with exactly the same covariate values. Such matching is called Exact matching and can be considered the gold standard for matching. However, Exact matching is not always applicable to an observational study because often the treatment units lack a perfect match in the control unit. In this situation, we will need to choose a metric to evaluate the distance between units that can be used to match them.

There are many ways to calculate distances. Some common ways to measure the distance between unit i and j for matching are Euclidean distance, Mahalanobis distance, and propensity score calculation.

Euclidean distance is defined as,

$$d_{ij} = \sqrt{\frac{\sum_{k=1}^K (x_{ik} - x_{jk})^2}{s_k}}.$$

where x_{ik} is the covariate value for unit i for k^{th} covariate, and s_k is the standard deviation of the k^{th} covariate.

Mahalanobis distance is defined as,

$$md_{ij} = \{(x_i - x_j)' s^{-1} (x_i - x_j)\}^{\frac{1}{2}}$$

where s is the sample covariate matrix of X . Propensity score distance is defined as,

$$ps_{ij} = |e_i - e_j|$$

where e_k is the propensity score for individual k . The propensity score for individual k can be defined as the probability of receiving the treatment given the observed covariates:

$$e_k(X_k) = p(T_k = 1|X_k)$$

The main characteristic of the propensity score is that the propensity scores are balancing scores. The balancing property says that treated ($T = 1$) and control ($T = 0$) subjects with the same propensity score $e(X)$ have the same distribution of the covariate X . Hence the treatment T and observed covariate X are conditionally independent given the propensity score. Although it is difficult to match multiple covariates at once, it is easy to match one covariate, the propensity score $e(X)$. Matching $e(X)$ will tend to be balanced across all the covariates. However, in an observational study, the propensity score is unknown and needs to be estimated using a regression model. For instance, the estimate $\hat{e}(X)$ of the propensity score $e(X)$ can be determined by a model, such as a logit model, relating treatment assignment T and observed covariates X . Hence this falls under the first limitation. Randomization in an experimental design can provide a basis for believing that observed and unobserved covariates are balanced, but matching on propensity score only can provide balance on observed covariate not on unobserved covariates. This falls under the second limitation of propensity score matching.

Mahalanobis metric matching has some useful properties. First, it is scaled invariant. The second property is that the Mahalanobis metric incorporates correlations between the covariates. As a limitation, the Mahalanobis metric matching does not work well when covariate vector X is high dimensional because the Mahalanobis metric matching method tries to account for all interactions between the covariates.

1.4 Matching Methods

Matching can be done either with or without replacement. Several treatment units can be matched to the same control unit in matching with replacement. However, only one

treatment unit can be matched into each control unit in matching without replacement. Therefore, matching with replacement proved to produce high-quality matches (Sävje, 2019).

Researchers may choose a matching technique depending on certain factors, such as the size of the data set and the time required to get the optimal matching. The chosen method will sometimes give the best but not the optimal solution. Considering any matching solution other than the optimal solution many necessitate assessing the accuracy of the solution.

Optimal pair matching finds the matched sample with the smallest average distance weights. Thus, the optimal matching technique does a better job of minimizing the total distance within matched pair.

Greedy matching is a commonly used matching technique in which the solution is sub-optimal. The greedy matching technique always chooses which element of a set seems to be the best with the lowest weight at the moment. Moreover, greedy matching will consider the closest control match for each treated unit one at a time without reconsidering it at a later point. Greedy matching sometimes may not achieve the minimum total distance in matched pairs. Thus, optimal matching is always better compared to greedy matching.

Another important multivariate matching method is the genetic matching method (Diamond and Sekhon, 2013). The Genetic matching method does not depend on knowing or estimating the propensity score. However, Genetic matching is greatly improved when a known or estimated propensity score is incorporated. The Genetic matching method uses a search algorithm to check and enhance covariate balance iteratively, and it is a generalization of propensity score and Mahalanobis metric matching.

Finally, the optimal cardinality matching method solves an integer programming problem to find the maximum matched sample with some constraints to obtain covariate balance (Zubizarreta et al., 2014). Furthermore, the optimal cardinality matching method may match data with missing covariates values if the procedure has a similar pattern of missing covariates in the matched treatment and control groups. The optimal cardinality matching method uses original covariates instead of a summary to match, and the method can directly balance covariate distributions.

1.5 Matching and graph theory

Matching may consider a mathematical graph theory problem where vertices are treatment or control units, and edges between vertices indicate that the corresponding units may pair in the matched sample. A graph $G = (V, E)$ is connected if for every pair of distinct vertices $u, v \in V(G)$, there is a path from u to v in G .

A matching M on G is an edge set or subgraph induced by the edge set M . More precisely, matching is the graph created by edges of the matching and related incident vertices.

Maximal matching is unique and not a subset of any other matching. Including a new edge cannot increase the size of the maximal matching because the included edge will share a common vertex with at least one edge already in the matching.

A matching M in graph G is a maximum matching if no other matching in the graph has more edges than M . However, not every maximal matching is a maximum matching.

If matching $M \in G$ and every vertex in G is incident with an edge in M , this is called perfect or complete matching. Not every graph will have a perfect match.

A subgraph induced by a matching M is a bipartite graph. A bipartite graph is in bijection form where all the edges start from one subset of a vertex set and end in the other disjoint subset of the vertex set.

1.6 Contribution

This study proposes to perform matching using the bottleneck subgraph. The solution holds the bottleneck constraint on optimal matching by performing a matching algorithm within a subgraph. The objective of common matching algorithm is to minimize the sum of imbalances. For instance, existing matching methods can make average balance on observed covariates such as weight height variables. This study proposes a way to prevent poor matches from being paired while minimizing the sum of imbalances. In other words, this study focuses on avoiding severe imbalances within individual matching.

One way to prevent poor matching in a matched sample is to use a caliper and remove all

the matched pairs that have within-pair imbalances higher than the caliper value. Another way is to impose the maximum imbalance as a constraint in the optimal matching framework. The proposed method may impose the maximum imbalance as a constraint in the optimal matching framework.

Optimal matching without incorporating the bottleneck constraints still results in numerous matches with modest imbalances and some matches with large imbalances. However, the optimal matching with imposed maximum dissimilarity threshold value via bottleneck constraint will ensure the prevention of bad imbalances within individual matching.

This dissertation addresses three tasks in statistical pair matching. First, it discusses available techniques and demystifies matching algorithms in Chapter 2. Second, Chapter 3, performs matching using the bottleneck subgraph. Finally, this study presents a new algorithm to solve minimum weight-maximum cardinality pair matching problem in Chapter 4.

Chapter 2

Demystifying Statistical Matching Algorithms for Big Data

2.1 Introduction

Consider an observational study where each unit is given exactly one of two treatment conditions: treatment or control. When confounding variables—those that are correlated with both treatment and response—are present, failure to account for this confounding may lead to significant bias in treatment effect estimates ([Rosenbaum et al., 2010](#)). For instance, in a study assessing the effect of smoking and heart disease, confounders include having poor diet and exercise habits as both variables are correlated with an increased incidence in heart disease and a higher likelihood of smoking.

Statistical matching is a technique designed to isolate the effect of treatment in the presence of confounders. In statistical matching, treated units are matched with control units with similar values for confounding covariates. Treatment effect estimates can then be obtained by taking, for example, the average of the differences in response between the treated and matched control units. Statistical matching plays an essential role in conducting research work in many subject areas, such as medicine ([Austin, 2009](#)), economics ([Galiani et al., 2005](#)), and political science ([Sävje et al., 2021](#)), since experiments are not always

practical or ethical to conduct.

With advances in computing, the volume of observational data has increased dramatically. For example, Electronic Health Records(EHR) collect valuable clinical information that researchers can use to guide patient care. EHRs include information on patient demographics, progress notes, problem lists, medications, vital signs, past medical history, etc([Gliklich et al., 2019](#)).

With this surge in available data, there is a significant need for matching methods that can be applied under big data settings.

We aim to provide a detailed description of the available techniques and tools for statistical matching, thereby adding some clarity to the black box of statistical matching and possibly enlightening the path towards future advances. We have particular focus on issues of the scalability of matching algorithms—the ability to successfully apply matching algorithms as the number of units under study becomes large.

The chapter is organized as follows. Section [2.2](#) review the problem setup for statistical matching with notations. Matching problems are well-studied optimization problems in the operation research area. Hence, in Section [2.3](#), we discuss greedy matching with advantages and disadvantages and matching with replacement in Section [2.4](#). Then, in Section [2.5](#), we discuss in full detail the problem of optimal statistical matching without replacement, which is the problem we will be discussing rest of our study. The discussion will continue to explain related materials from the optimization area for statistical matching. Section [2.5.1](#) will discuss how to model the maximum cardinality matching problem as an optimization problem and solve it using an integer linear programming approach. Also, there will be an explanation of the Ford Fulkerson algorithm to find maximum cardinality in Section. Moreover, Section [2.5.2](#) discusses minimum cost matching with the cycle canceling algorithm. Section [2.6](#) will explore why scaling down data in statistical matching is important, and Section [2.7](#) discusses the existing methods to solve minimum cost maximum cardinality matching on a sparse graph.

2.2 Problem Setup for Statistical Matching

Consider an observational study on N units, numbered 1 through N . For each unit i , we observe a response y_i , a treatment status $t_i \in \{0, 1\}$ —where $t_i = 1$ denotes that i is given treatment and $t_i = 0$ denotes that i is given control—and a p -dimensional vector of confounding covariates $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$. Let N_t denote the number of treated units, numbered 1 through N_t , and let N_c denote the number of control units, numbered 1 through N_c . For ease of exposition, we assume $N_t < N_c$.

Between each treated unit i and control unit j , a dissimilarity measure w_{ij} may be computed on the confounding covariates $\mathbf{x}$, where smaller values of w_{ij} indicate that i and j have more similar values of confounding covariates. Common choices of w_{ij} include the standardized Euclidean and Mahalanobis distances and the absolute difference in estimated propensity scores (Imbens and Rubin, 2015). Intuitively, matching aims to find, for each treated unit i , one or more control units j that have “small dissimilarity.”

As Rosenbaum (1989) astutely noted, in considering the problem of statistical matching, there is a large body of literature—historically in the field of operations research—on similar types of matching problems from which to draw inspiration. In deliberately vague terms, these problems often start by assuming a mathematical graph and aim to select connected pairs of units within the graph in an “optimal” way. Hence, to make statistical matching problems more precise and to help draw connections between statistical matching and matching problems in the operations research literature, we describe these problems in terms graph theory. For simplicity, we focus on the 1:1 matching case, where each treated unit is allowed to be matched to, at most, one control unit (Sävje et al., 2021), and extend our approach to more complicated matching schemes (e.g. 1: k matching, full matching, generalized full matching, cardinality matching) when appropriate (Hansen, 2007).

For statistical matching, units under study are represented as a graph $G = (V, E)$; each node i in the node set V represents a unit under study (hence, $|V| = N$), and edges ij in the edge set E are drawn between two nodes if their corresponding units are allowed to be matched with each other. Each edge ij has a non-negative cost w_{ij} equal to the dissimilarity

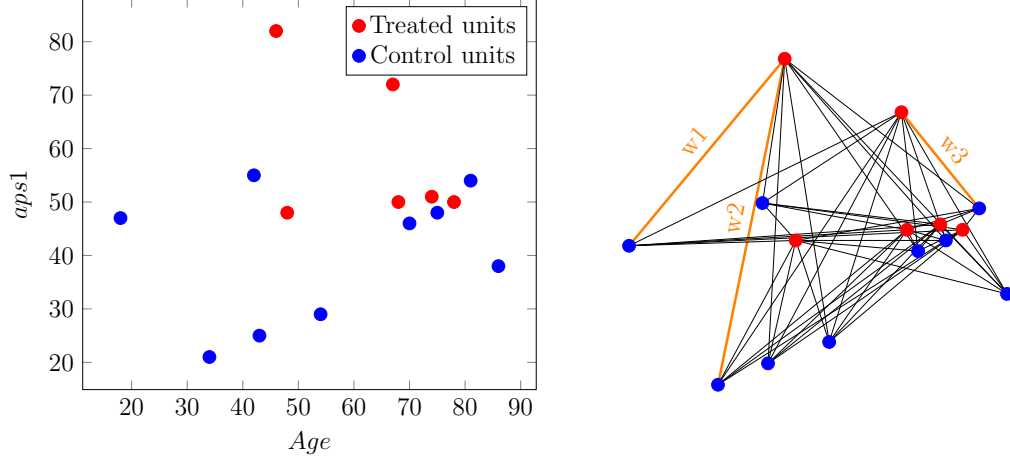


Figure 2.1: An example of transforming an observational study into a bipartite graph with distance as dissimilarities

between the corresponding units i and j . The resulting graph is a *bipartite* graph; the node set V can be partitioned into two groups V_T and $V_{C'}$ —those nodes that correspond to treated and control units respectively—and edges are only allowed to connect a node from V_T to one in $V_{C'}$ (e.g. you cannot match a treated unit to another treated unit nor a control unit to another control unit). When initializing a matching problem, it is common to make minimal assumptions on which units can be matched to each other, thereby allowing the matching algorithm to completely determine which matches are appropriate. In terms of the graph G , this corresponds to the assumption that edges $ij \in E$ exist between each pair of units $i \in V_T, j \in V_C$ —that is, G is a *complete* bipartite graph. For ease of exposition, we may refer to nodes as units and edge costs as dissimilarities throughout this paper.

A 1:1 statistical matching is a subset of edges $M \subset E$ such that each treated node $i \in V_T$ is *incident* to, at most, one edge $ij \in M$ —that is, each node $i \in V_T$ is the endpoint of at most one edge in M . If $ij \in M$, then the control unit j is matched to the treated unit i before performing analyses. Since the power of a study is most affected by the number of observations included in the study, often we aim to select a matching M with a large cardinality $|M|$; often, we require the cardinality to be maximized.

For a given dataset, there may be a large number of candidate matchings $\mathbb{M}$ from which to choose. Hence, many statistical matching algorithms aim to select a matching $M^\dagger \in \mathbb{M}$ that

is optimal with respect to some *objective function*. The commonly used objective function in statistical matching is to minimize total dissimilarity between treatment and control pairs in the matched sample; matching algorithms aim to minimize the total dissimilarity

$$M^\dagger = \arg \min_{M \in \mathbb{M}} \sum_{ij \in M} w_{ij}. \quad (2.1)$$

Other objectives used in matching include: minimizing the maximum dissimilarity within a match (Sävje et al., 2021); minimizing the maximum p -value for tests for the null hypothesis that covariate distributions between treated and matched control groups are equivalent (Diamond and Sekhon, 2013); and maximizing the number of matched pairs subject to constraints on the difference in sample moments between treated and matched control groups (Zubizarreta, 2012; Zubizarreta et al., 2014).

Matching can be performed *with replacement*—multiple treated units are allowed to be matched to the same control units—or *without replacement*. The statistical problem of finding a matching without replacement and the operations research problem of finding a bipartite matching are equivalent. Hence, significant progress on the statistical matching problem can be made by importing well-studied ideas from the optimization literature.

Before we discuss optimal methods for performing matching without replacement, we take a couple brief detours. First, we describe greedy matching, which is computationally efficient but can suffer from arbitrarily poor performance when matching without replacement. Then, we discuss in full detail the problem of matching with replacement. Statistical matching with replacement is a well-understood problem—greedy algorithms can often obtain an optimal matching straightforwardly and efficiently—but it may be inappropriate to use under certain settings.

2.3 Greedy Matching

Greedy algorithms provide a simple and intuitive solution for statistical matching problems. Greedy matching algorithms match each treated unit with the eligible control unit that is

most similar (with respect to the dissimilarity measure w). Simple implementations of greedy algorithms can terminate quickly. Specifically, for each treated unit, the problem of finding the most similar control unit requires $O(N)$ time, and this problem is solved a maximum of $O(N)$ times, leading to a worst-case total runtime of $O(N^2)$ (Cormen et al., 2022), outside of the cost of computing the dissimilarities w . Thus, greedy matching is computationally inexpensive enough for most studies using observational data.

However, when matching without replacement, greedy matching may have significant drawbacks. When the selected dissimilarity measure does not satisfy the triangle inequality (*i.e.* for any three units i, j, k , $w_{ij} + w_{jk} \leq w_{ik}$), the total dissimilarity of a 1:1 greedy matching can be infinitely bigger than that for an optimal matching (Rosenbaum, 1989). Even when the dissimilarity measure satisfies the triangle inequality, the difference in total dissimilarity between 1:1 greedy matching and optimal matching may worsen as data sizes get large—to be exact, the difference may be as large as $O(n^{\log_2(3/2)}) \approx O(n^{0.58})$ (Agarwal and Sharathkumar, 2014; Reingold and Tarjan, 1981). Additionally, a greedy matching may have a smaller cardinality than an optimal matching (Rosenbaum, 1989), and the matching quality may depend on the order in which treatment units are selected for matching (Dehejia and Wahba, 2002).

A 1:1 greedy matching algorithm proceeds as follows.

1. **(Initialize)** Set the greedy matching $M = \emptyset$.
2. **(Select treated node)** Select unit $i \in V_T$ (for example, at random).
3. **(Find control match)** Of all eligible control units j , find the unit $j^\dagger \in V_C$ that is the most similar to i :

$$j^\dagger = \arg \min_{j: ij \in E} w_{ij}. \quad (2.2)$$

Match i to $j^\dagger$: Set $M \xleftarrow{set} M \cup ij^\dagger$. If no matches are possible, skip to Step 4.

4. **(Remove matches)** Set $V_T \xleftarrow{set} V_T \setminus \{i\}$. If matching without replacement, and if a control match $j^\dagger$ was found in Step 3, set $V_C \xleftarrow{set} V_C \setminus \{j^\dagger\}$ and $E \xleftarrow{set} E \setminus \{ij^\dagger : i \in V_C\}$.

5. **(Terminate)** If $V_T = \emptyset$, stop. The matching M is a greedy matching. Otherwise, return to Step 2.

A greedy 1: k matching is performed by choosing the k most similar units to the treated unit i in Step 3 of the algorithm. The performance of greedy matching without replacement highly depends on the order in which treated units are selected for matching in Step 2 (Austin, 2014). Improved methods for choosing treated nodes—for example, finding the edge $ij \in E$ with the largest dissimilarity w_{ij} , and choosing the treated unit i incident to this edge—often come with an increased computational cost.

2.4 Statistical Matching with Replacement

Matching with replacement permits different treated units to be matched to the same control unit. The biggest advantage of matching with replacement is computational cost. Greedy matching is almost always used to perform matching with replacement as it is optimal for a number of commonly used objective functions—including the total dissimilarity and the maximum dissimilarity—under this setting.

There may be additional instances where, in practice, matching with replacement outperforms without replacement. For example, matching with replacement may perform better in practice when the distribution of confounding covariates between treated and control groups have little overlap (Dehejia and Wahba, 2002). It can also be used to estimate the average treatment effect for the treated (ATT) when the number of treated units is greater than the number of control units—matching without replacement would necessarily leave some treated units unmatched, thereby changing the estimand. Monte Carlo simulations have suggested that matching with replacement can provide reliable treatment effect estimates if control units are reused for matches infrequently, and suggest that covariate distributions between treated and control groups are more similar for 1: k matching with replacement than without replacement, $k > 1$ (Bottigliengo et al., 2021).

However, there may also be some drawbacks with matching with replacement. First and foremost, there is no way to easily control how many times one control unit is used in a

match. For a given study, it may be possible that many treatment units are matched to a single control unit. In this case, the response of the control unit will disproportionately influence the estimate of the treatment effect, thereby inflating the standard error of the matching estimator. Moreover, simulation results suggest that 1:1 matching without replacement usually yields a smaller difference in sample means between treated and matched control groups than with replacement (Bottigliengo et al., 2021). Matching with replacement is rarely used in certain areas of study—for example, in the biomedical sciences—where matching without replacement appears to be more effective (Austin and Small, 2014).

2.5 Optimal Statistical Matching Without Replacement

In 1:1 matching without replacement, each control unit is included in at most one pair in the matched sample. Hence, once a control unit is selected for matching, that control unit is no longer eligible for consideration as a potential match for subsequent treatment units. This substantially increases the difficulty of finding an optimal match, for example, with respect to the total dissimilarity objective. Thankfully, these types of statistical matching problems are well studied, though most of this work originates from the field of operations research—Rosenbaum (1989) first identified the connection between statistical matching and this optimization literature.

The most common statistical matching optimization problem is to find a matching $M^\dagger$ that minimizes the total dissimilarity given that it contains as many matched pairs as possible—or more precisely, under the constraint that the cardinality of $M^\dagger$ is maximized. In the statistical matching literature, these matchings are simply called *optimal matchings* (Rosenbaum, 1989). This optimization problem is traditionally broken into two separate subproblems:

1. **(Maximum cardinality matching)** Find the maximum cardinality $m^\dagger$ across all feasible matchings.

2. **(Minimum cost matching)** Find the matching that has the smallest total dissimilarity under the constraint that the matching contains $m^\dagger$ matched pairs.

Note, if $|V_T| \geq |V_C|$ and each treated unit is eligible to be matched to every control unit—*i.e.* if G is a complete bipartite graph—then the maximum cardinality $m^\dagger = |V_T|$. We now describe these subproblems in detail.

2.5.1 Maximum Cardinality Matching

Recall that a 1:1 matching without replacement M on a graph $G = ((V_T, V_C), E)$ is a subset $M \subset E$ such that each treated node $i \in V_T$ and control node $j \in V_C$ is incident with at most one edge in M . The maximum cardinality matching problem (MaxCard) is to find a matching in G such that the cardinality of the matching $|M|$ is as large as possible.

MaxCard may be formulated as an *integer linear programming* problem (ILP). This formulation associates each edge $ij \in E$ with a binary variable z_{ij} , where $z_{ij} = 1$ if the match $ij \in M$ and $z_{ij} = 0$ if $ij \notin M$. MaxCard aims to find, across all possible vectors of variables $\mathbf{z} = (z_{ij})_{ij \in E}$, an optimal vector $\mathbf{z}^\dagger$ that satisfies

$$\mathbf{z}^\dagger = \arg \max_{\mathbf{z}} \sum_{ij \in E} z_{ij}$$

under the constraints that

$$\begin{aligned} \sum_{i \in V_T} z_{ij} &\leq 1 \quad \forall j \in V_C, \\ \sum_{j \in V_C} z_{ij} &\leq 1 \quad \forall i \in V_T, \\ z_{ij} &\in \{0, 1\} \quad \forall ij \in E. \end{aligned} \tag{2.3}$$

This problem is an *integer* programming problem as the variables $\mathbf{z}$ are integer valued, and is *linear* because both the objective function and the constraints are linear combinations of the $\mathbf{z}$ variables.

Some available matching methods work directly on this objective. One notable example, *cardinality matching* (Zubizarreta et al., 2014), solves this ILP with an additional constraint ensuring that, for example, the differences in sample means of confounding covariates between treated and matching control groups are within some pre-specified tolerance threshold. After finding a matching $M^\dagger$ that maximizes the cardinality, this method then finds a optimal matching (Rosenbaum, 1989) on all units incident to an edge in $M^\dagger$ before estimating treatment effects.

Solving Integer Linear Programming Problems

There are, broadly speaking, two kinds of approaches for solving these types of ILP matching problems. The first method is to work directly on the integer program. A common approach is to relax the integer constraint on the variables $\mathbf{z}$ to allow z_{ij} to take values within the entire interval $[0, 1]$. This relaxation results in a standard linear programming (LP) problem. After this relaxation, additional constraints—for example, blossom inequalities (Edmonds and Johnson, 2003)—can be iteratively added to the LP to force a solution with 0–1-valued variables.

As a particularly interesting case, when all coefficients on the variables z_{ij} are integer valued, the *integrality theorem* (Dasgupta et al., 2008) ensures that there exists a solution to the LP such that $z_{ij} \in \{0, 1\}$. In this case, a standard a linear program (LP) solver—for example, the simplex method (Dantzig, 1990; Nelder and Mead, 1965)—can exactly solve the original ILP. In practice, this is a quite common setting; edge costs are often multiplied by a large power of 10 and rounded to the nearest integer before the optimization problem is initialized. While this necessarily yields an approximation to the original statistical matching problem, such a matching tends to be acceptable in practice.

The second approach is to iteratively manipulate characteristics of the matching graph G —for example, cycles, minimum cuts, or shortest paths (Kovács, 2015)—until an optimal solution is found. For instance, a traditional approach for solving MaxCard is to first transform this problem into a network flow problem. Then, algorithms designed to find maximum

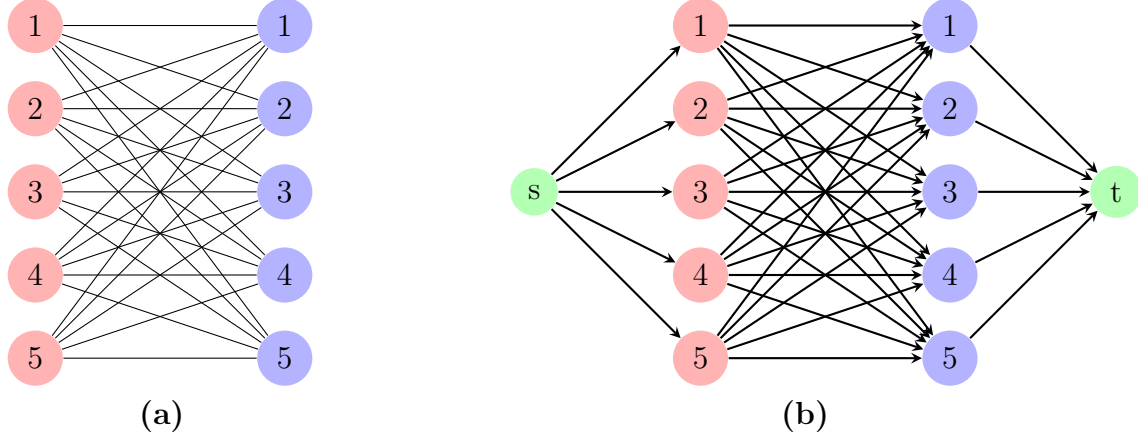


Figure 2.2: (a). A bipartite graph with five treated and control units. (b). The network flow graph for (a).

flows can be applied to solve the original MaxCard problem. Algorithms for solving maximum flow problems are often computationally efficient, and thus, may be scalable to large observational studies. We now describe a solution using this approach—the Ford-Fulkerson algorithm—in detail (Ford and Fulkerson, 1957).

Ford-Fulkerson for Solving MaxCard

A traditional approach to solving MaxCard begins by reducing this problem to a maximum flow problem. To do this, we first transform G to a digraph $G' = (V', E')$ —that is, each edge in E' is now directed. Specifically, we allow edges in E' to travel from a node in V_T to a node in V_C , but not the other direction: for $i \in V_T$, $j \in V_C$, and $ij \in E$, we have $\vec{ij} \in E'$, but $\vec{ji} \notin E'$. We then add a *source* node s and a *sink* node t to G , and we connect these nodes to G' by adding edges traveling from the source node to each node in V_T and edges traveling from each node in V_C to the sink node: for $i \in V_T$, $\vec{si} \in E'$, and for $j \in V_C$, $\vec{jt} \in E'$. Finally, we assign each edge in $e \in E'$ a *capacity* c_e equal to 1. Figure 2.2 details this transformation.

In general, a *flow* on the digraph G' from the source s to the sink t is a real valued function f on each edge $e \in E'$ satisfying the following conditions:

1. For any edge $e \in E'$: $0 \leq f(e) \leq c_e$. If $f(e) = c_e$, we say that the flow is *saturated* on that edge.

2. For any vertex $j \in V' \setminus \{s, t\}$, the total flow into the vertex j is same as the total flow out of the vertex. That is,

$$\sum_{i: \vec{ij} \in E'} f(\vec{ij}) = \sum_{k: \vec{jk} \in E'} f(\vec{jk}). \quad (2.4)$$

The value $|f| = \sum_{i: \vec{si} \in E'} f(\vec{si})$ is the total flow out from the source s , and hence, from (2.4), the total flow entering into t is $|f|$. Under this setup, each flow f on G' induces a matching $M_f \subset E$ obtained by selecting the edges that the flow saturates:

$$M_f = \left\{ ij \in E : i \in V_T, j \in V_C, f(\vec{ij}) = 1 \right\}. \quad (2.5)$$

The maximum flow problem is to find a flow $f^\dagger$ that maximizes the total flow into t . If all capacities are integers—as is the case with MaxCard—it is possible to find such an $f^\dagger$ with integer values for all edges: $f(e) \in \mathbb{N} \cup 0 \forall e \in E'$ (Dasgupta et al., 2008). Upon finding such a maximum flow $f^\dagger$, a maximum cardinality matching $M^\dagger$ is a matching induced by this flow: $M^\dagger = M_{f^\dagger}$.

The Ford-Fulkerson algorithm (FFA) is commonly used to solve maximum flow problems (Ford and Fulkerson, 1956). Intuitively, FFA works by starting from an initial flow f and iteratively finding paths of edges in G' from s to t that will lead to increases in the total flow of f .

FFA is most easily described through the introduction of residual graphs. Given the maximum flow digraph $G' = (V, E')$ and a flow f , the *residual* graph $H = (V, E_f)$ is a digraph on the nodes V . For each edge $\vec{ij} \in E'$, there is a “forwards” $\vec{ij}$ and “backwards” $\vec{ji}$ version of this edge in the residual graph E_f :

$$E_f = \left\{ \vec{ij} \cup \vec{ji} : \vec{ij} \in E' \right\} \quad (2.6)$$

For MaxCard in particular, forward edges $\vec{ij}$ have a *residual capacity* of $\delta(\vec{ij}) = 1 - f(\vec{ij})$, which denotes the unused capacity on edge $\vec{ij}$. Backward edges $\vec{ji}$ have capacity $\delta(\vec{ji}) = f(\vec{ij})$,

which denotes how much the flow on edge $\vec{ij}$ can be suppressed. That is,

$$\delta(\vec{ij}) = \begin{cases} 1 - f(\vec{ij}), & \vec{ij} \in E', \\ f(\vec{ij}), & \vec{ji} \in E'. \end{cases} \quad (2.7)$$

Once the residual graph H is constructed, FFA finds paths $P = \{s\vec{i}_1, \overrightarrow{i_1 i_2}, \dots, \overrightarrow{i_{\ell-1} i_\ell}, i_\ell \vec{t}\}$ from s to t within this residual graph such that the capacity $\delta(\vec{ij}) > 0$ for each edge $\vec{ij} \in P$. These paths are called *augmenting paths*. The current flow f can then be improved by adding flow to the forward edges and decreasing flow to the backwards edges along this path.

Rigorously, FFA for MaxCard is performed as follows:

1. **(Initialize flow)** Set $f(\vec{ij}) = 0$ for all edges $\vec{ij} \in E'$.
2. **(Update residual graph)** Update the residual graph $H = (V, E_f)$ with capacities given in (2.7).

3. **(Find augmenting path or terminate)** Find an augmenting path

$$P = \{s\vec{i}_1, \overrightarrow{i_1 i_2}, \dots, \overrightarrow{i_{\ell-1} i_\ell}, i_\ell \vec{t}\} \text{ from } s \text{ to } t \text{ such that } \delta(\vec{ij}) = 1 \text{ for all edges } \vec{ij} \in P.$$

If no such path exists, stop.

4. **(Augment the flow)** Update the flow f along all edges $\vec{ij} \in P$:

$$\begin{aligned} f(\vec{ij}) &\leftarrow f(\vec{ij}) + 1, & \vec{ij} \in P, \vec{ij} \in E', \\ f(\vec{ij}) &\leftarrow f(\vec{ij}) - 1, & \vec{ji} \in P, \vec{ji} \in E'. \end{aligned} \quad (2.8)$$

Return to Step 2.

Each iteration of FFA increases the flow of f by 1. For general maximum flow problems, finding an augmenting path takes $O(E')$ time. Moreover, if all capacities in the maximum flow problem are integer-valued, then the flow f at termination in Step 3 is a maximum flow, and reaching this flow requires, at most, $|f^\dagger|$ iterations. In particular, for MaxCard, the maximum cardinality $m^\dagger \leq |V_T| < N$, and so, total runtime of FFA is bounded by

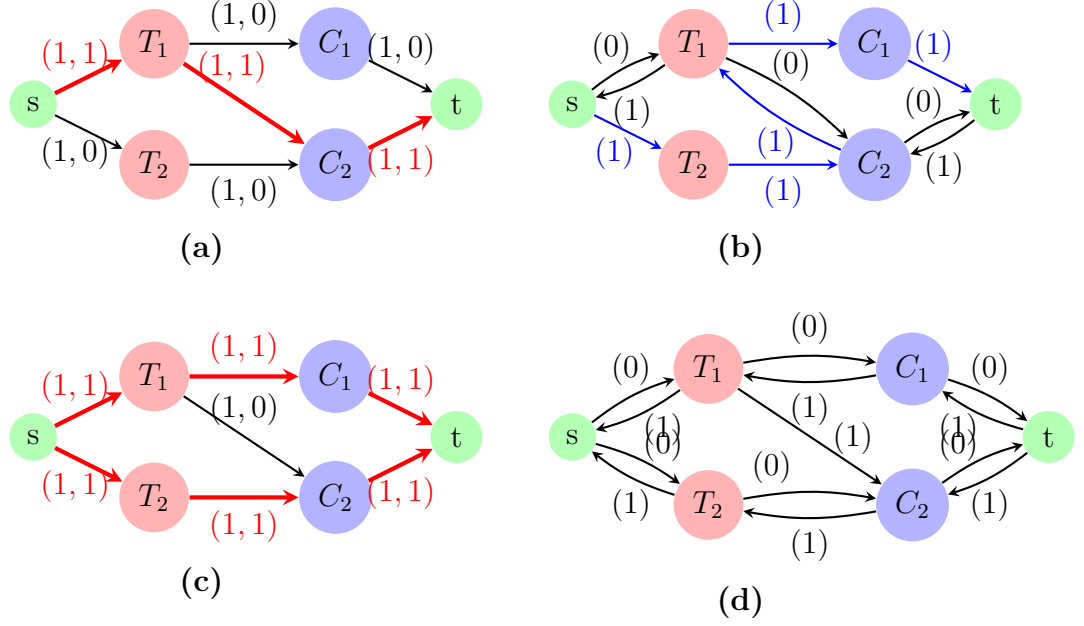


Figure 2.3: (a). The flow network G and initial flow f with (capacity, flow). (b) The residual graph for (a) with augmenting path p in blue and residual capacity (δ); Consider the reverse-path $C_2 - T_1$ and selecting path $s - T_2 - C_2 - T_1 - C_1 - t$. (c). The flow in G that results from augmenting along path p by its residual capacity. (d). The residual network induced by the flow in (c); no path can be found $s - t$ with all edges those with $\delta = 1$.

$O(NE) \leq O(N^3)$. Moreover, when the graph is *sparse*—that is, when the number of edges is proportional to the number of nodes—this runtime is reduced to $O(N^2)$. In practice, FFA tends to be computationally efficient enough for most statistical matching applications.

2.5.2 Minimum Cost Matching

Recall that, in the matching graph $G = (V, E)$, each edge $ij \in E$ has a cost $w_{ij} \geq 0$. The general form of a minimum cost matching problem (MinCost) is to find a matching $M^\dagger$ that minimizes the total cost (2.1) under a constraint that $M^\dagger$ has sufficiently large cardinality. Constraints on the cardinality of the matching prevent a trivial optimal solution of $M^\dagger$ containing no matched pairs.

As with MaxCard, MinCost can be formulated as an ILP, where we aim to find an optimal

solution $z^\dagger$ satisfying

$$z^\dagger = \arg \min_z \sum_{ij \in E} w_{ij} z_{ij}$$

under the constraints that

$$\begin{aligned} \sum_{i \in V_T} z_{ij} &\leq 1 \quad \forall j \in V_C, \\ \sum_{j \in V_C} z_{ij} &\leq 1 \quad \forall i \in V_T, \\ \sum_{i \in V_T} \sum_{j \in V_C} z_{ij} &\geq m, \\ z_{ij} &\in \{0, 1\} \quad \forall ij \in E. \end{aligned} \tag{2.9}$$

The constraint $\sum_{i \in V_T} \sum_{j \in V_C} z_{ij} \geq m$ ensures that the optimal matching $M^\dagger$ satisfies $|M^\dagger| \geq m$ (and, in fact, $|M^\dagger| = m$, as any extra edges in $M^\dagger$ can be removed without an increase in the total cost). In practice, optimal matching problems will set the cardinality to $m^\dagger$, the maximum cardinality possible for a match.

Cycle Canceling for Solving MinCost

Apart from the linear programming approach, there are a variety of approaches for solving MinCost. We discuss one of these approaches—cycle canceling ([Klein, 1967](#))—while noting that other approaches, including cost-scaling, relaxation, and simplex approaches, may also yield relatively efficient solutions for MinCost.

As with MaxCard, cycle canceling approaches for solving MinCost begin by transforming the problem into an optimal flow problem. The digraph $G' = (V, E')$ described in [Section 2.5.1](#) is constructed. For completeness, costs w are defined on all edges $\vec{ij} \in E'$ by setting $w_{si} = 1$ for all $i \in V_T$ and $w_{jt} = 1$ for all $j \in V_C$. FFA approaches can then be used to find an initial flow f_0 satisfying $|f_0| = m$. Finally, the residual graph $H = (V, E_{f_0})$ is

constructed, and costs w^H are assigned to each edge $\vec{ij} \in E_{f_0}$ as follows:

$$w_{ij}^H = \begin{cases} w_{ij}, & \vec{ij} \in E' \text{ and } f_0(\vec{ij}) = 1, \\ w_{ij}, & \vec{ji} \in E' \text{ and } f_0(\vec{ji}) = 0, \\ -w_{ij}, & \text{otherwise.} \end{cases} \quad (2.10)$$

That is, costs are positive for forward edges that are used the flow from s to t and for backwards edges not used in this flow; costs are negative otherwise.

Searching negative cycles and canceling them with a cycle canceling algorithm will then find the minimum cost for the matching. A *cycle* C in the residual graph H is a path that begins and ends at the same node $C = \{\overrightarrow{i_1 i_2}, \overrightarrow{i_2 i_3}, \dots, \overrightarrow{i_{\ell-1} i_\ell}, \overrightarrow{i_\ell i_1}\}$. A *negative cycle* is a cycle C^- in which the sum of the costs along edges in the cycle is negative: $\sum_{\vec{ij} \in C^-} w_{ij} < 0$. It can be shown that the matching M induced by a flow f is a minimum cost matching if and only if there are no negative cycles within the corresponding residual graph (Klein, 1967). There are a variety of methods for finding negative cycles, including the Bellman-Ford algorithm and minimum-mean cycle approaches.

For MinCost specifically, each cycle within the residual graph will have the same number of forward edges traveling from a treated unit to a control units as backward edges traveling from a control unit to a treated unit. Once a negative cycle C^- is found, the flow is updated by pushing flow forward through the backward edges in C^- and preventing flow from traveling through the forward edges in C^- . The matching induced by the updated flow will have the same cardinality as the matching with the original flow, but will have a smaller total cost. The process of updating the flow from the negative cycle is called *cycle canceling*.

Rigorously, cycle canceling for MinCost is performed as follows:

1. **(Initialize flow)** Find initial flow f on $G' = (V, E')$ with total flow $|f| = m$. Define costs w on all edges E' as previously described.
2. **(Update residual graph)** Update the residual graph $H = (V, E_f)$ with costs given in (2.10).

3. **(Find negative cycle or terminate)** Find a cycle $C^- = \{\overrightarrow{i_1 i_2}, \overrightarrow{i_2 i_3}, \dots, \overrightarrow{i_{\ell-1} i_\ell}, \overrightarrow{i_\ell i_1}\}$ satisfying $\sum_{\vec{ij} \in C^-} w_{ij} < 0$.

If no such cycle exists, stop.

4. **(Update the flow)** Update the flow f along all edges $\vec{ij} \in C$ as follows:

$$\begin{aligned} f(\vec{ij}) &\leftarrow f(\vec{ij}) + 1, & \vec{ij} \in C^-, \vec{ij} \in E', \\ f(\vec{ij}) &\leftarrow f(\vec{ij}) - 1, & \vec{ji} \in C^-, \vec{ji} \in E'. \end{aligned} \tag{2.11}$$

Return to Step 2.

As mentioned before, each iteration of the cycle canceling algorithm will find a flow with the same total flow but a smaller total cost. Standard approaches for finding negative cycles require $O(N|E'|)$ time. However, unlike with MaxCard, there may not be a restrictive upper bound for the number of iterations required to find an optimal solution. If all costs are integer-valued, cycle canceling algorithms can terminate in $O(N|E'| \sum_{\vec{ij} \in E'} w_{ij})$ iterations as each iteration will reduce the total cost by at least 1. Additionally, some algorithms have been developed for MinCost that are guaranteed to terminate in polynomial time with respect to N , even if costs are not integer-valued. The most well-known of these algorithms, minimum mean-cycle cancelling (Goldberg and Tarjan, 1989), requires at most $O(N|E'|^2)$ iterations, leading to a total runtime of $O(N^2|E'|^3) \leq O(N^8)$. This is substantially more computationally complex than FFA. Again, ensuring sparsity in the matching graph can dramatically reduce the runtime—down to $O(N^5)$ for sparse graphs.

More recent approaches for solving MinCost may yield improvements to this total run time. However, despite these developments, current state-of-the-art algorithms for solving MinCost still require significantly more computation than those for solving MaxCard. Consequently, solving MinCost tends to be the computational bottleneck for statistical matching algorithms.

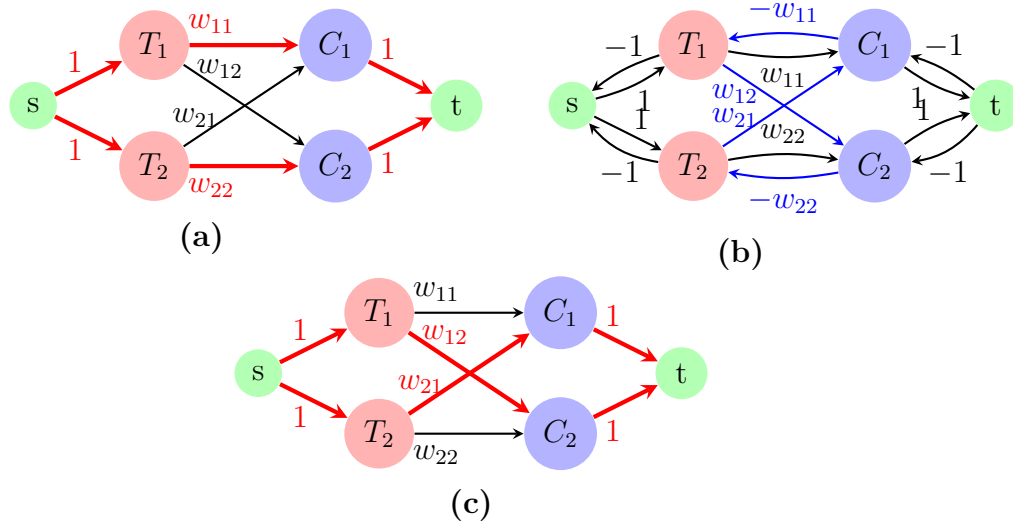


Figure 2.4: (a). The original flow network G with initial flow f with dissimilarities w . (b). The residual graph for (a) with augmenting path p color in blue; consider the reverse-paths $c_1 - T_1$ and $c_2 - T_2$, selecting path $c_1 - T_1 - C_2 - T_2 - C_1$ with $-w_{11} + w_{12} - w_{22} + w_{21} < 0$ negative cost. (c). The flow in G that results from augmenting along path p .

2.6 Scaling down data in statistical matching

2.6.1 Complete, dense, and sparse graphs

A complete bipartite graph is a particular type of bipartite graph where every vertex of one set is connected to every vertex of the other set. A dense graph has an edge count close to the number of vertices squared. Thus a complete bipartite graph is a dense graph with density 1, as in Figure 2.2(a). If no edges connect two sets of vertices, the density is 0. Hence, the graph density is a spectrum from entirely unconnected, that is, no edges, to completely connected. In a sparse graph, the number of edges is roughly proportional to the the number of vertices. Many algorithms on graphs run in time proportional to the number of edges. Thus, graph algorithms will run a lot faster on a sparse graph.

2.6.2 Adjacency list and adjacency matrix

The most common data structures to represent graphs are adjacency matrices and adjacency lists. An adjacency matrix for the graph $G = [V = (V_T, V_C), E]$ is a $N \times N$ matrix A as

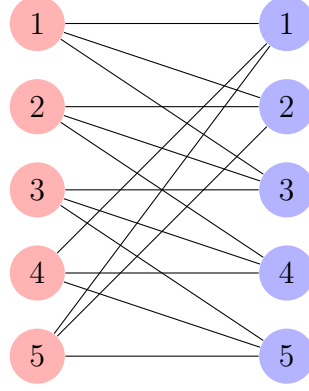


Figure 2.5: Figure shows the sparse graph of bipartite graph shown in Figure 2.2 (a)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) Adjacency matrix for 2.2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \omega_{11} & \omega_{12} & \omega_{13} & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \omega_{22} & \omega_{23} & \omega_{24} & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \omega_{33} & \omega_{34} & \omega_{35} \\ \infty & \infty & \infty & \infty & \infty & \omega_{41} & \infty & \infty & \omega_{44} & \omega_{45} \\ \infty & \infty & \infty & \infty & \infty & \omega_{51} & \omega_{52} & \infty & \infty & \omega_{55} \\ \omega_{11} & \omega_{12} & \omega_{13} & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \omega_{22} & \omega_{23} & \omega_{24} & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \omega_{33} & \omega_{34} & \omega_{35} & \infty & \infty & \infty & \infty & \infty \\ \omega_{41} & \infty & \infty & \omega_{44} & \omega_{45} & \infty & \infty & \infty & \infty & \infty \\ \omega_{51} & \omega_{52} & \infty & \infty & \omega_{55} & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

(b) Weight matrix for figure 2.2

Figure 2.6: Adjacency matrix and weight matrix

shown in Figure 2.6 (a), such that:

$$A_{ij} = \begin{cases} 1 & ij \in E \\ 0 & \text{otherwise} \end{cases}$$

If G is an undirected graph, then $A_{ij} = A_{ji}$, whereas if G is a digraph, then A is generally asymmetric. If A represents a sparse graph, that is, a graph with the ratio $\frac{|E|}{N}$ is low, then most of the elements are zero, as shown in Figure 2.6 (a). In an adjacency list representation, each vertex has an associated list of adjacent vertices.

In terms of the accessing time, an adjacency matrix is more efficient to check a given edge exists. However, in an adjacency list, it is exactly the opposite. All the list elements

need to be checked to find any relationship with the list structure in a graph. Hence, an adjacency matrix is much more efficient when finding the connections in a graph in terms of the accessing time.

The main advantage of adjacency lists over the matrix is space requirement. Since the adjacency matrix is a $N \times N$ matrix, it requires $O(N^2)$ space complexity. However, adjacency lists only store the vertex connected in a list or a vector. Thus, the adjacency list uses much lower memory than the matrices for graphs. Therefore, the adjacency list is much more efficient for storing the graph, especially sparse graphs, with fewer edges than nodes.

2.7 Matching in a Sparse Graph

2.7.1 An overview of existing methods

Contemporary technological advancements generate large data sets in biological, social, and other scientific fields. When the data set gets large and large, the graph has more edges to store and access. Even though large data sets offer many options, some bad options can be eliminated initially. For example, if we have a data set with 5000 control units and 5000 treated units, the first treatment unit has 5000 control units to be considered. There should be some control units with high covariate imbalance that are impossible to match with the first treatment unit. In statistical matching, eliminating these impossible matches from the graph (deleting edges with high dissimilarity) will lead to a sparse graph. As discussed in section 2.4.1, graph algorithms will run faster on a sparse graph than on a dense graph.

Notice that reducing information from the original data set may lead to biased results if the elimination is not done carefully. Thus, the practical problem is how to define the boundary value for the elimination. Next, we will discuss existing methods to reduce the size of the graph.

Sparse graphs have been discussed in some contexts of matching literature. In particular, if researchers want to compare treated and control units within natural blocks, this approach helps solve small-sized optimization problems ([Pimentel et al., 2015](#)). Further, natural blocks

are beneficial for creating a sparse graph with a shortlist of treated and control units within the same block.

The subclassification or exact matching of some significant categorical type covariates can be used to obtain a sparse graph. Even though the exact matching is used to control confounders, this approach reduces the number of edges from a dense graph. Hence, each treated unit's shortlist consists of controls from homogeneous unit set or from the same strata.

Another interesting restriction to making a sparse graph is fine balance. Since the exact matching strictly includes identical control units for treatment units, the list of potential controls for a given treatment unit is concise. The fine balance principle does not preserve such a condition and tolerates a mismatch in one pair providing it can counterbalance in another pair ([Rosenbaum et al., 2007](#)). A more general framework concerning matching a treated male to a control female in one block will match a treated female to a control male in another Block. Thus, the final treated and control groups have the same number of males and females. This exact counterbalancing is called fine balance. If the researcher still has difficulty obtaining fine balance, near fine balance is the minimal deviation from fine balance achievable from the available data ([Yang et al., 2012](#)). Near fine balance constrains the marginal distributions of a nominal variable to be as close as possible to Fine balance. Refined covariate balance is an extension of fine or near fine balance ([Pimentel et al., 2015](#)). In refined covariate balance, treatment and control units would obtain pairs such that the nominal covariates are balanced as closely as possible.

Another possible restriction is to consider matching with caliper. The propensity score is helpful to compare treated and control groups with similar distributions of measured covariates. In pair matching with a pre-specified caliper, forms pairs of units such that the difference in propensity scores between matched units is at most of the caliper width ([Cochran and Rubin, 1973](#)). Thus the caliper matching can lead to a sparse graph. However, the choice of caliper should reflect the variance-bias trade-off. Using narrower calipers will result in matching more similar subjects and sparser graphs. However, it may also reduce the number of matched units, thereby increasing the variance of the estimated treatment

effect. Using wider calipers should have the opposite effect.

The outstanding work of imposing minimax constraint on a minimum total distance matching proposed a new method to minimize the total within-pair distance subject to various constraints, including the constraint that the maximum pair difference is as small as possible ([Rosenbaum, 2017](#)). However, this method also eliminates edges in the network to shortlist the potential control units for each treatment unit. Thus, the performance of the minimum cost flow algorithm takes less time to construct in the resulted sparse graph. A new approach related to the minimax constraint has been discussed as a fast algorithm to find an interval containing the optimal caliper to find the optimally matched sample without generating infeasibility ([Yu, 2021](#)). The new approach considers removing the maximum number of edges that can be removed to make a feasible matched sample, and it will lead to making a sparser graph.

2.7.2 A discussion about existing software tools

The concept of using network flow theory to obtain optimally matched samples gained much algorithmic attention in statistical matching literature ([Rosenbaum, 1989](#)). For instant, minimum cost flow problems in network theory can be used to obtain statistical matching. Another feature is that the minimum cost flow algorithm is efficient for large matching problems with dense, sparse, and intermediate networks. However, Rosenbaum highlighted that the linear programming methods could directly increase the run time and are inefficient in using space. Furthermore, Rosenbaum has suggested optimal construction for several statistical matching problems like matching with multiple controls, matching with the variable number of controls, and balance matching.

There are numerous implementations frameworks available in R software for statistical matching. `MatchIt` package is one famous implementation with several matching methods and various options. These option helps to accommodate facilities for different matching goals, such as matching with multiple controls, matching with a variable number of controls, and matching with balance ([Ho et al., 2007](#)). In addition, `MatchIt` acts as a flexible wrapper

to function in various packages, each of which has a different syntax, but the users do not need to learn.

Notice that our principle in this study depends on optimal 1:1 matching, and the discussion about optimal pair matching for sparse graph is the main focus. The simplest version of the assignment problem, the *personal-assignment problem*, was solved by Kuhn in 1955 with the proposed Hungarian algorithm (Kuhn, 1955). The optimal pair matching problem is a version of the *personal-assignment problem* in which the target is to pair each treated unit with a different control unit to minimize the total dissimilarity within matched pairs. A new faster algorithm to solve the *personal-assignment* problem was proposed by Bertseks in 1981 and was more efficient than Kuhn’s algorithm (Bertsekas, 1981). The implementation of the Bertsekas algorithm is available in Hansen’s `optmatch` R package (Hansen, 2007).

When finding the optimal pair matching, the time-consuming part is to solve the Minimum Cost Flow (MCF) network problem. Hence, the discussion about solvers for MCF problem is important for this study. There are well-known efficient MCF solvers available in the operation research area, such as CS2, RelaxIV, LEMON, etc.

The RelaxIV algorithm is a faster version of the Relax algorithm and combines the Relax algorithm with initialization based on the auction/sequential shortest path algorithm (Bertsekas et al., 1994). This initialization helps speed up the solution of problems with long augmenting paths.

CS2 implements the cost-scaling push-relabel algorithm (Goldberg, 1997). The cost-scaling codes are substantially faster on the largest sparse network due to their better asymptotic behavior regarding the number of nodes. This variant of the cost-scaling algorithm performs path augmentations instead of local push operations, and relabeling is heavily used to find augmenting paths.

The LEMON (Library for Efficient Modeling and Optimization in Networks) solver has four efficient implementations for solving the MCF network flow problem (Kovács, 2015), (LEM). The four separate algorithms in the LEMON library are Cycle Cancelling, Network Simplex, Cost Scaling, and Capacity Scaling. Each algorithm has its own trade-offs and performance advantages. Network Simplex is usually the fastest algorithm on relatively

small and dense graphs, while Cost Scaling performs better on large and sparse graphs. The `optmatch` package has two solver options; *rlemon* and *rrelxiv*. The *relemon* and *rrelxiv* are R versions of the LEMON and RELAX-IV solvers.

The `optmatch` package is the most famous R package used for optimal pair matching in the statistical matching literature. However, to execute the `pairmatch` function in `optmatch`, we need to give information about a complete graph (or distance matrix), as in Figure 2.3 (b). Hence, we need to input details of a complete graph to get the output from the `pairmatch` function in R, which is an inefficient way, especially for a sparse graph. Thus, if we consider the computational complexity, these unnecessary details can delay the solution and use memory inefficiently.

The optimal pair matching technique can only minimize the sum of distances between treated units and matched control units. To have more control over the covariate balance, Jose R. Zubizarreta has proposed to enter some covariate imbalance terms to the objective function of the Mixed Integer Programming problem. This added constraints can ensure the covariate balance in the matched sample (Zubizarreta, 2012).

A study about cardinality matching followed by minimum distance pairing was proposed to magnify the above study with Mixed Integer Program problem (Zubizarreta et al., 2014). The implementation of maximum cardinality matching, `cardmatch`, is available to the statistical community with the `designmatch` R package. The solver options for the `cardmatch` function are `cplex` and `gurobi` (Zubizarreta et al., 2018). Hence, the largest matched sample that satisfies the covariate balance requirements can be found using the `cardmatch` function. But to achieve the minimum total dissimilarity requirement in the matched sample, `cardmatch` will rematch the sample using the `pairmatch` function (Visconti and Zubizarreta, 2018).

The next most important package is `rcbalance` package for optimal pair matching. Let us remark here that the `rcbalance` can use both adjacency list and adjacency matrix formats to input graph information. Hence, `rcbalance` can efficiently deal with large, sparse graphs with respect to time and space requirements to find optimal matching. Another remarkable property of this recently implemented `rcbalance` package is finding optimal matches under

a set of refined covariate balance constraints (Pimentel et al., 2015). These constraints are nominal variables that intend to marginally balance as closely as possible in the resulting treated and matched control set.

2.8 Conclusion

Several methods exist in optimization theory to improve and support statistical matching, but at present, adding some clarity to the black box of statistical matching and possibly enlightening the path towards future advances is essential. Also, the adjacency list format in statistical matching is more important than the commonly used dissimilarity matrix format in big data setting. Existing algorithms do not sufficiently facilitate ways to solve minimum weight-maximum cardinality matching on a sparse graph with an adjacency list format. The *rcbalance* is competitive compared to *optmatch*, which we accept as a good source for optimal pair matching in big data sets because of the adjacency list format.

Chapter 3

Preventing poorly-matched pairs using bottleneck subgraph matching

3.1 Introduction

Consider an observational study where each unit is given exactly one of two different conditions: treatment or control. Confounding variables—those that impact both the response of a unit and the likelihood of that unit receiving the treatment condition—can complicate estimates and inferences made on the effect of treatment on response. Failing to account for these variables may exaggerate the relationship between treatment and response.

Matching procedures are used to restrict and reorganize the original sample data for statistical analysis to isolate the effect of treatment in the presence of confounders. In statistical matching, each treated unit is paired with a similar control unit with respect to their pretreatment variables. Often, the similarity of two units is determined via a dissimilarity measure—for example, the Mahalanobis distance between the pretreatment covariates of two units—where more-similar pairs of units have smaller dissimilarity values. Treatment effects are then estimated by, for example, taking the difference in response between the treated unit and its matched control unit.

Most matching methods aim to minimize the total dissimilarity between treated and

matched control units in the matched sample. However, this may open the door for matching with significant imbalances on certain covariates within a small proportion of pairs (Rosenbaum, 2017). Hence, a researcher may also desire to restrict the allowable maximum within-pair dissimilarity for each covariate.

In this chapter, we focus on matchings that restrict the maximum within-pair dissimilarity (MWPD) on each covariate. The approach may first define a subset of potential matches for each treated unit within MWPD, called a *bottleneck subgraph*. Any matching method that is performed within this bottleneck subgraph may ensure that each match has within this tolerance. Moreover, the proposed method has other benefits, such as producing a well-matched sample, being effective in large data sets, and allowing the researchers to select prespecified threshold values for each covariate.

Our proposed method works by appending a new set of constraints to matching estimators to restrict allowable MWPD for each covariate. The proposed constraints guarantee to discard all pairings that exceed a researcher-specified dissimilarity threshold for any given covariate. We then solve the matching problem by transforming it into a graph theory problem: units under study are nodes in the graph, an edge between two nodes denotes that the two units are permitted to be paired together, and each edge has a weight denoting the dissimilarity between two units. We show that matching with the MWPD constraint is equivalent to finding a match within a bottleneck subgraph, in which edges in the graph are drawn if and only if their weight is less than some prespecified threshold. Then we apply our method to a study on the efficacy of right heart catheterization. Results indicate that implementing bottleneck constraints can obtain a statistical matching efficiently. The matching successfully eliminates poor matches from the sample and may yield a strong overall balance on confounding covariates.

This chapter is organized as follows. Section 3.2 may discuss the benefits of the proposed method, specifically the importance of the maximum dissimilarity threshold value in optimal pair matching with motivating two examples. Section 3.5 discussed how to impose the bottleneck constraint on some optimal matching problems in the context of statistical matching. After introducing the maximum dissimilarity value as a constraint first, the discussion may

poof imposing bottleneck constraint is equivalent to bottleneck subgraph matching. Some related work for this study is discussed in Section 3.3, and Section 3.4 discuss how to impose bottleneck constraints with notation and preliminaries. Section 3.6 illustrates how to find the optimal pair matching constrained by the maximum dissimilarity threshold value with a data example of Study to Understand Prognoses and Preferences for Outcomes and Risks of Treatments (SUPPORT) data. Finally, Section 3.7 discusses the findings in conclusion.

3.2 Motivation

An *optimal pair matching* (Rosenbaum, 1989) pairs each treated unit to one control unit *without replacement*—control units cannot be reused across pairs—and selects the matching that minimizes the *total dissimilarity*—the sum of the dissimilarities across all pairs. This type of matching is the most common matching method in many subject areas, such as medicine (Austin, 2009), economics (Galiani et al., 2005), and political science (Sävje et al., 2021). We now motivate our approach of imposing constraints on the maximum dissimilarity allowed within a match.

3.2.1 Prevent poor individual matches

Optimal pair matching with MWPD makes it possible to prevent poor individual matches in the matched sample. We now give two motivating examples comparing the results of optimal pair matching with MWPD without MWPD.

Example 1:

Suppose there are four treatment and four control units with dissimilarities as shown in Table 3.1, where ϵ is a positive number close to 0. The value of 0 indicates those units are perfect matches, and the infinity indicates such pairings are impossible. There are two possible matchings without replacement for which all four treated units are paired to control units: Let, $M_1 = \{(C_1, T_1), (C_2, T_2), (C_3, T_3), (C_4, T_4)\}$ and $M_2 = \{(C_1, T_2), (C_2, T_3), (C_3, T_4), (C_4, T_1)\}$.

Table 3.1: Dissimilarity values between treated and control units.

		Control Units			
		C_1	C_2	C_3	C_4
Treated Units	T_1	1	∞	∞	$4-\varepsilon$
	T_2	0	1	∞	∞
	T_3	∞	0	1	∞
	T_4	∞	∞	0	1

Table 3.2: Dissimilarity measures between treated and control units with maximum dissimilarity threshold value of 1.5.

		Control Units			
		C_1	C_2	C_3	C_4
Treated Units	T_1	1	∞	∞	∞
	T_2	0	1	∞	∞
	T_3	∞	0	1	∞
	T_4	∞	∞	0	1

The total dissimilarity for M_1 is 4 and the total dissimilarity for M_2 is $4 - \varepsilon$.

Optimal pair matching yields M_2 with $4 - \varepsilon$ total dissimilarity. However, the dissimilarity between T_1 and C_4 in M_2 is nearly 4 times as large as the dissimilarity between any two pairs in M_1 . That is, although M_2 is optimal, M_1 may be preferable to prevent a poor match from being formed.

To be explicit, suppose that a researcher specifies that only dissimilarities of 1.5 or less within a match are tolerable. Table 3.2 then shows the dissimilarity matrix for the modified problem with this MWPD. The deletions of all edges beyond the MWPD allows the researcher to obtain an optimal pair matching within the MWPD. According to Table 3.2, M_1 is the only optimal pair matching that exists. Hence, the optimal pair matching with MWPD therefore prevents choosing M_2 , preventing a poorly-matched pair in the matched sample.

Table 3.3: Propensity score values for three treatment and control units.

Treatment Unit	Propensity Score	Control Unit	Propensity Score
T_1	0.4	C_1	0.3
T_2	0.3	C_2	0.2
T_3	0.2	C_3	0.1

Table 3.4: Absolute difference of dissimilarity between treatment and control units.

		Control Units		
		C_1	C_2	C_3
Treatment Units	T_1	0.1	0.2	0.3
	T_2	0	0.1	0.2
	T_3	0.1	0	0.1

Example 2:

This illustration uses three treatment units and three control units with propensity score dissimilarity to discuss the concept of preventing poor matching in the matched sample; see Table 3.3.

Table 3.4 shows the absolute difference in dissimilarity for each pair of treatment and control units. The value of zero indicates those units are perfect matches, and the infinity indicates such pairings are impossible. According to Figure 3.1, there are $3! = 6$ possible matches, and four of them are optimal solutions with the same minimum total dissimilarity of 0.3. Table 3.5 shows all four optimal solutions without applying MWPD: S_1, S_2, S_3 , and S_4 . The S_1 option has a maximum within-pair distance of 0.1, the S_2 and S_3 matching options have the maximum within-pair distance of 0.2, and the S_4 has a maximum within-pair distance of 0.3. As all optimal matching options have the same total dissimilarity of 0.3, typical optimal pair matching algorithms may arbitrarily choose one. The user then has no authority to restrict within-pair maximum dissimilarity in the matched sample.

In contrast, the MWPD can be used to restrict the within-pair maximum dissimilarity in the matched sample. For example, suppose that the researcher decides to allow 0.2 as the maximum dissimilarity threshold value for the propensity score. Then Figure 3.2(a)

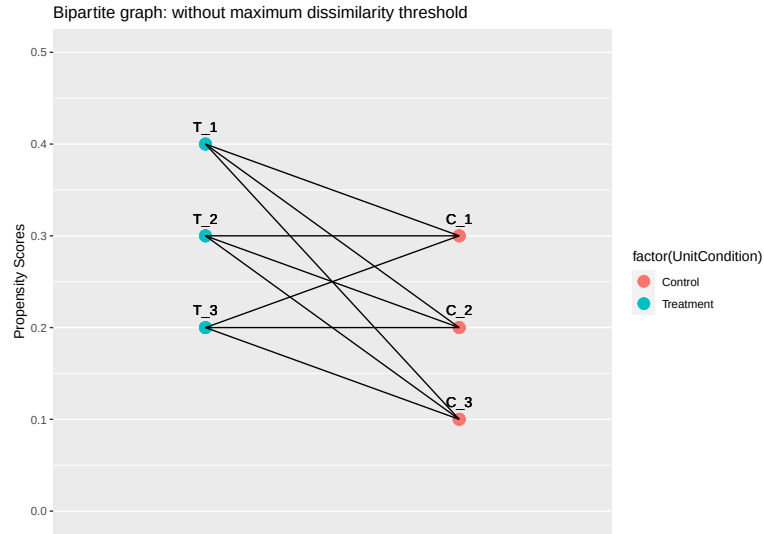


Figure 3.1: Bipartite graph without MWPD

Table 3.5: Optimal four solutions without any maximum dissimilarity threshold value

		Matching set	Total dissimilarity
Optimal Solutions	S_1	$(T_1, C_1), (T_2, C_2), (T_3, C_3)$	$0.1+0.1+0.1=0.3$
	S_2	$(T_1, C_1), (T_2, C_3), (T_3, C_2)$	$0.1+0.2+0=0.3$
	S_3	$(T_1, C_2), (T_2, C_3), (T_3, C_1)$	$0.2+0+0.1=0.3$
	S_4	$(T_1, C_3), (T_2, C_1), (T_3, C_2)$	$0.3+0+0=0.3$

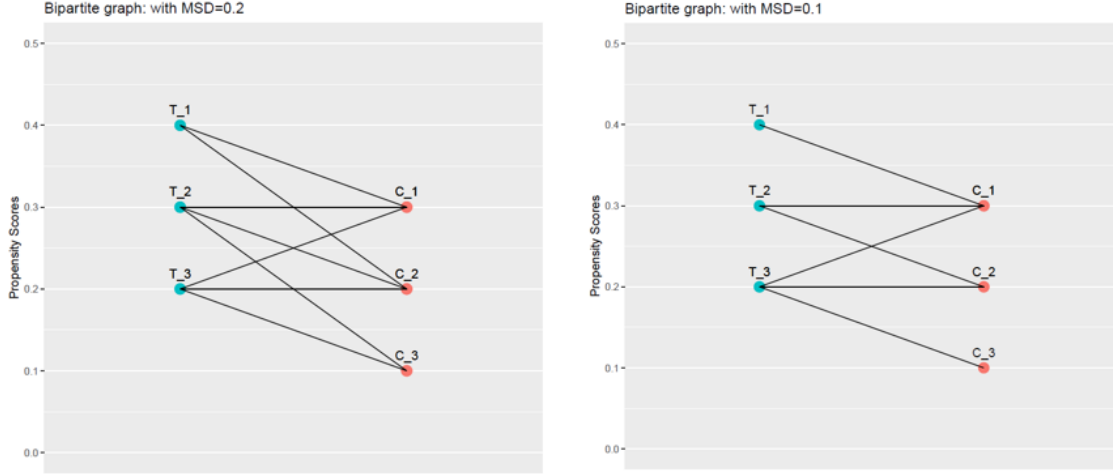


Figure 3.2: Bipartite graphs MWPD=0.2, and MWPD=0.1

Table 3.6: Absolute difference of dissimilarity measures between treated and control units for $MWPD \leq 0.2$

		Control Units		
		C_1	C_2	C_3
Treatment Units	T_1	0.1	0.2	∞
	T_2	0	0.1	0.2
	T_3	0.1	0	0.1

illustrates the bottleneck subgraph for the $MWPD = 0.2$.

When deleting all the edges with dissimilarity greater than 0.2, the resulting subgraph remains with four possible matches. Thus, the optimal pair matching with MWPD is considered a more convenient sparse graph than a dense one. The three optimal solutions, S_1, S_2 , and S_3 , with $MWPD = 0.2$, are given in Table 3.6.

Now, assume a MWPD as 0.1 for propensity scores. In this case, S_2, S_3 , and S_4 optimal solutions are no longer valid, and S_1 is the only optimal solution for this problem as given in Figure 3.2(b) and Table 3.7. Hence, when tightening MWPD, there is a reduction in the density of the graph and the complexity of the problem.

Table 3.7: Absolute difference of dissimilarity measures between treated and control units for $MWPD \leq 0.1$

		Control Units		
		C_1	C_2	C_3
Treatment Units	T_1	0.1	∞	∞
	T_2	0	0.1	∞
	T_3	0.1	0	0.1

3.2.2 Reducing number of potential matches

Imposing maximum dissimilarity threshold constraints provides a pathway to restrict the solution space for the original matching problem. This restriction can dramatically reduce the computational cost required to perform optimal matching, which is critical for large observational studies.

3.2.3 Researcher-oriented covariate balance/ Inflating and deflating the caliper

A convenient property is that it allows researchers to specify how large of a difference is tolerable on each covariate within a match. The matching then is performed with this specification. If such a match does not permit each treated unit to be matched to a control unit, all tolerances may be scaled up in a similar manner to [Rosenbaum \(2017\)](#) until such a matching exists. Furthermore, if the selected matched sample does match each treated unit to a control unit, researchers may shrink the tolerances to tighten the pairs in the matched sample. That is, the proposed method allows a straight-forward way to inflate or deflate tolerances gives researchers more flexibility in their analysis process.

3.3 Related work

Preventing poor matches may be as essential as producing *covariate balance*—ensuring equal pretreatment covariate distributions between treated and control groups—in observational

data. Researchers historically focus on covariate balancing by incorporating constraints, helping eliminate some poorly matched pairs from the final matched sample. Often, these constraints also work to limit the allowable dissimilarity between matched pairs.

The research question may require perfect balance on some covariates, attainable by exact matching, and close balance on other covariates, achieved through the fine balance (Zubizarreta et al., 2011). Exact matching matches each treated unit to a possible control unit with exactly the same values on pre-treated covariates. Fine balance refers to exactly balancing a nominal variable by constraining the marginal distribution of a nominal covariate without trying to match individuals on this variable (Rosenbaum et al., 2007). Exact matching is the most powerful matching method of removing the effects of confounders without creating poorly matched pairs, but is ineffective with continuous covariates. Especially in the multivariate setting, finding two units with the same combination of all covariates is infrequent. Coarsened exact matching proposes a solution to this problem (Iacus et al., 2011). Coarsening transforms a continuous covariate into a discrete one by partitioning it into strata. The matching then takes place on these strata to achieve covariate balance in the matched sample. While exact matching and coarsened exact matching are intended to balance pretreatment covariates, they can still somewhat restrict intolerable pairing.

For example, a sample with a fine-balanced gender covariate may still pair a male with a female in the matched sample. If a fine balance is not feasible, researchers try to achieve near-fine balance; the marginal distributions of a categorical covariate in treated and control groups are as close as possible to the fine balance (Yang et al., 2012).

When covariates are continuous and exact and coarsened-exact matching methods perform poorly, there are a few methods to restrict matchings. A caliper is a common method of restricting control unit selection that can impact treatment effect estimation. A caliper works by removing matches that have a dissimilarity greater than some prespecified threshold (Imbens and Rubin, 2015). As an alternative approach, (Zubizarreta et al., 2014) proposes cardinality matching in which the amount of covariate imbalance allowable is constrained and the matching maximizes the number of pairs used in the matching. Of note, these methods may remove treated units from the sample. In this case, the quantity estimated by

the matching estimator may be altered. See section 3.4.2 for details.

The most relevant work to this study is that on imposing minimax constraints on optimal pair matching in observational studies (Rosenbaum, 2017). Among all matchings that allow all treated units to be matched to control units, the method aims to find the matching that minimizes the MWPD. A binary search involving repeatedly constructed matched samples is then used to find the minimax value. As an extension, Yu (2021) derives an algorithm for computing an interval efficiently that is guaranteed to contain this minimax value.

3.4 Notation and preliminaries

Suppose there are N units under study, numbered 1 to N , with N_T units, numbered 1 to N_T , given treatment and N_C units given control. Unit i has a response Y_i and a K -dimensional covariate $\mathbf{x}_i = (x_{i1}, \dots, x_{iK})$.

We view statistical matching as a graph theory problem. Let $G = (V = (V_T, V_C), E)$ denote a *bipartite graph*; each node $i \in V_T$ denotes a treated unit and each node $j \in V_C$ denotes a control unit and edges $ij \in E$ are only permitted to join a node in V_T to a node in V_C . We may refer to elements in V_T and V_C as nodes or units interchangeably. Often, few restrictions are made on E ; it is to assume G is *complete*, where $ij \in E$ for every pair $i \in V_T, j \in V_C$.

Each edge ij has a non-negative weight w_{ij} equal to the dissimilarity between the corresponding units i and j . These weights w_{ij} may be any dissimilarity measurement; most commonly used measures include the weighted Euclidean distance, the Mahalanobis distance, and the absolute differences in the *propensity score*—the probability that a unit i is treated given its covariate vector $\mathbf{x}_i$ (Rosenbaum and Rubin, 1983; Sekhon, 2009). Lower weights indicate pairs of units that are more similar. The dissimilarity choice must satisfy $w_{ij} \geq 0$ for $ij \in E$. To impose a constraint on the MPWD, we also consider the largest caliper dissimilarity.

A *matching* M is a subset of edges $M \subset E$. In the case where matching is performed without replacement—as is the case with optimal matching (Rosenbaum, 1989)—each unit

$i \in V$ may not be included in the matching multiple times.

3.4.1 Largest Caliper Dissimilarity

The *dissimilarity threshold* δ_k is a researcher-specified value on each covariate x_k denoting how much values of x_k can differ within an “acceptable match”—if two units i and j have $|x_{ik} - x_{jk}| > \delta_k$, then i cannot be matched to j . The value ω_{ij}^∞ represents the largest factor to which i differs from j across all univariate covariates.

Dissimilarity thresholds can be used to form a dissimilarity measure ω^∞ known as the largest caliper dissimilarity (LCD) (Mahmood, 2020) as follows:

$$\omega_{ij}^\infty = \max_k \frac{|x_{ik} - x_{jk}|}{\delta_k}. \quad (3.1)$$

Unit j is then considered “similar-enough” to unit i if and only if $\omega_{ij}^\infty \leq 1$. Hence, ω^∞ is a way to encompass all user-specified thresholds in a single dissimilarity measure. Moreover, while ω_{ij}^∞ may be intuitive for continuous-valued covariates, it may still be useful for categorical variables. For example, if covariate k is a dummy variable, setting $\delta_k < 1$ ensures that $\omega_{ij}^\infty \leq 1$ if and only if the matching is exact on k .

3.4.2 Model of Response: Neyman-Rubin Causal Model

We assume that the response of unit i follows the Neyman-Rubin Causal Model (NRCM) of response (Holland and Rubin, 1987; Rubin, 1974; Splawa-Neyman et al., 1990):

$$Y_i = y_{i1}T_i + y_{i0}(1 - T_i). \quad (3.2)$$

Here, y_{i1} and y_{i0} are potential outcomes—the hypothetical responses of unit i given treatment and control respectively, and T_i is a 0–1-valued treatment indicator, where $T_i = 1$ denotes that unit i received treatment.

The most common estimand for observational studies under this model is the *average*

treatment effect for the treated (ATT):

$$\text{ATT} = \frac{1}{N_T} \sum_{i=1}^{N_T} (y_{i1} - y_{i0}). \quad (3.3)$$

Note that this average is taken over only treated units.

Observe that, if a matching omits treated units, then the estimator no longer estimates the ATT. Instead, a conditional average treatment effect (CATT) is estimated. In this case, efforts should be taken to ensure interpretability of the subsequent conditional estimand ([Fogarty et al., 2016](#); [Hill and Su, 2013](#)).

3.5 Bottleneck subgraph matching with largest caliper dissimilarity

We now introduce bottleneck subgraph matching (BSM), which allows the researcher to impose a bottleneck constraint on the allowable dissimilarity on individual covariates within a match.

3.5.1 BSM as a linear integer program

A bottleneck constraint with LCD ensures that units that are matched together are sufficiently similar on each of the K covariates (with respect to the researcher’s preference). Finding an optimal statistical matching (without replacement) with a bottleneck constraint can be viewed as solving two linear integer programs. First, we solve a version of cardinality matching ([Zubizarreta et al., 2014](#)) in which the number of matched pairs is maximized under a constraint that all pairs must include sufficiently similar units. Then, optimal matching is solved. The optimal matching may use a different dissimilarity measure than LCD.

The cardinality optimization problem is to find variables $\{z_{ij} : ij \in E\}$ satisfying

$$m^\dagger = \max \sum_{(i,j) \in E} z_{ij},$$

under constraints,

$$\begin{aligned} z_{ij} &\in \{0, 1\}, \\ \sum_{i=1}^t z_{ij} &\leq 1 \quad \forall j \in V_C, \\ \sum_{j=1}^c z_{ij} &\leq 1 \quad \forall i \in V_T, \\ w_{ij}^\infty z_{ij} &\leq 1 \quad \forall ij \in E. \end{aligned} \tag{3.4}$$

The most critical aspect of this optimization problem is finding $m^\dagger$, the maximum number of pairs possible where, within each pair ij , $|x_{ik} - x_{jk}| \leq \delta_k$ for each covariate k .

Once the maximum cardinality $m^\dagger$ is found, the optimal match is then found by solving

$$\min \sum_{(i,j) \in E} w_{ij} \cdot z_{ij}$$

with constraints,

$$\begin{aligned} z_{ij} &\in \{0, 1\}, \\ \sum_{i=1}^t z_{ij} &\leq 1 \quad \forall j \in V_C, \\ \sum_{j=1}^c z_{ij} &\leq 1 \quad \forall i \in V_T, \\ \sum_{ij \in E} z_{ij} &= m^\dagger, \\ w_{ij}^\infty z_{ij} &\leq 1 \quad \forall ij \in E. \end{aligned} \tag{3.5}$$

Again, the weights w_{ij} used in the objective function may differ from the LCD w_{ij}^∞ . Once this optimization problem is solved, the optimal match $M^\dagger$ is found by matching unit i to j if and only if $z_{ij} = 1$.

While these two optimization problems precisely define a bottleneck subgraph matching with LCD, additional intuition can be garnered by viewing this problem in terms of graph theory.

3.5.2 Graph theory interpretation of BSM

We begin with a definition of a bottleneck subgraph under LCD,

Definition 2. A graph $G_\lambda = (V, E_\lambda) \subset G$ is a bottleneck constraint subgraph under LCD if $ij \in E_\lambda$ if and only if $w_{ij}^\infty \leq \lambda$.

Note that for G_1 , for each pair of units $i \in V_T$, $j \in V_C$ with $ij \in E_1$, $|x_{ik} - x_{jk}| \leq \delta_k$. That is, for a matching M to satisfy sufficient similarity on each covariate, it is necessary and sufficient for M to be *embedded* in G_1 —that is, $M \subset E_1$.

Theorem 1. A matching M has $|x_{ik} - x_{jk}| \leq \delta_k$, $k = 1, \dots, K$, for all $ij \in M$ if and only if $M \subset E_1$.

Proof. Suppose that $M \subset E_1$. By definition, for any match $ij \in M$, we have $ij \in E_1$, and hence $w_{ij}^\infty \leq 1$. This implies that

$$\max_k |x_{ik} - x_{jk}| \leq \delta_k \quad \text{for } k = 1, \dots, K.$$

Now suppose that $M \not\subset E_1$. That is, there exists $i'j' \in M$ with $i'j' \notin E_1$. By definition, for this match,

$$\omega_{i'j'}^\infty > 1,$$

implying for some covariate x_k that

$$\max_k |x_{i'k} - x_{j'k}| > \delta_k.$$

□

Of particular note, bottleneck subgraphs tend to be sparse, which can tremendously improve the computational cost required by optimal matching algorithms. Hence, a standard approach for solving these types of problems is to find the bottleneck subgraph G_1 and store it as an adjacency list. The list can then be imported into software packages designed to solve optimal matching problems.

3.5.3 Software tools for solving BSM

The `rcbalance` R package is the only R package to our knowledge that supports the adjacency list format when finding optimal pair matching (Pimentel et al., 2022). First, the bottleneck subgraph G_1 is found and is stored as an adjacency list. Then, the adjacency list is included in the call to the `rcbalance()` function in the `rcbalance` package. The `exclude.treated` feature in the `rcbalance` function may allow removing treatment units if no potential control units are matched to that treatment unit within the maximum threshold value. Also, by making `exclude.treated` equal `TRUE`, a minimal number of treatment units may be dropped to make the feasible matched sample. After removing unmatchable treatment units, the remaining treatment units can be used to estimate a local or conditional average treatment effect.

3.6 Illustration with data example

The example in this section illustrates how to find the optimal pair matching constrained by the maximum dissimilarity threshold value with Study to Understand Prognoses and Preferences for Outcomes and Risks of Treatments (SUPPORT) data.

Right Heart Catheterization (RHC) is a test for measuring how well the heart pumps blood to the lungs. The SUPPORT is an observational study regarding effectiveness of applying RHC in the initial care for critically ill patients (Connors et al., 1996). The SUPPORT data set contains information of hospitalized adult patients from five medical centers in the

USA. Detailed description of the data may be found in Appendix C.

The treatment is whether or not RHC is applied to a patient within 24 hours of admission where the patient qualified for the SUPPORT study. A binary outcome was defined based on whether a patient died at any time up to 180 days since admission. The SUPPORT contains information for 2184 RHC-applied patients and 3551 RHC-not applied patients. The goal of the study was to estimate the average treatment effect of RHC on patients' survival.

The original analysis used a binary logistic model to develop a propensity score that matched RHC patients with non-RHC patients. The results provided some evidence that patients receiving RHC had decreased survival time, and the sensitivity analysis indicated that any unmeasured confounder would have to be somewhat strong to explain away the results (Lin et al., 1998).

3.6.1 Results for LCD

To explain the proposed new algorithm, MWPD was applied to the SUPPORT data first. The method first used $\delta_k = 0.01 \times s$, where s is the standard deviation for each covariate k . Then by using the `rcbalance` package, the optimal pair matching was found in the resulting bottleneck subgraph. Thus, the within-pair dissimilarity does not pass the MWPD value in the matched sample.

Also, we solved the matching problem for two user-specified (Age, Sex, Aps) dissimilarity sets as shown in Table 3.8. The Matching 1 in Table 3.8 is for the set (3, 0.5, 5), which means Age can differ for at most three years, Sex should have exact matches, and Aps variable can differ at most five scores in the matched sample.

Assume that a researcher wants to match the treatment unit with covariate value x_k to any control unit with covariate values in $[x_k - w, x_k + w]$ interval. The proposed new algorithm can be used to produce matched samples for these choices as well. First, initialize the maximum dissimilarity for each covariate k and keep the caliper value as $\omega_{ij}^\infty = 1$.

Table 3.8 shows covariate balance and the number of treatment units in the matched sample for two user-defined MWPD value sets. The Matching 1 in Table 3.8 indicates Age

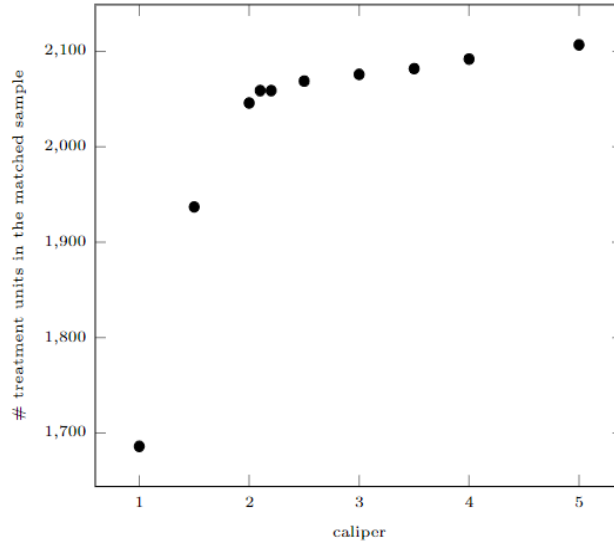


Figure 3.3: Number of treatment units in the matched sample for different caliper values

Table 3.8: Covariate balance(SMD) for two matching samples with user specified MWPD

	Matching 1		Matching 2	
	MWPD	SMD	MWPD	SMD
Age	3	0.001	5	0.011
sex	0.5	< 0.001	0.5	< 0.001
Aps1	5	0.045	8	0.078
# treatment units	2072		2131	

can differ for at most three years, Sex should have exact matches, and Aps variable can differ at most five scores in the matched sample.

While the SUPPORT data contains information for 72 variables, for simplicity, only a shorter list of confounders may use to illustrate this data example. Consider three confounding covariates to check the balance with MWPD: age, gender, and the APACHE score (aps1) variables. The APACHE score assesses the initial health of a patient. The $\delta_k = 0.01 \times s$ will be the bound for the analysis, where s is the standard deviation for each covariate k . As usual, both numeric and graphical diagnostics may assess the covariate balance.

In the raw data (before matching/before adjustment), the SUPPORT data contains two independent samples for treatment and control groups with different sample sizes. Hence,

Table 3.9: Two sample t-test results for each covariate before matching (Raw Data, n=2184)

	t-statistics	(mean of X, mean of Y)	95% CI	p-value for H_a
Sex	3.4199	(0.58516, 0.53900)	(0.01970, 0.07262)	0.00063
Age	-2.2296	(60.7498, 61.7609)	(-1.9001, -0.1221)	0.0258
Aps1	18.604	(60.7390, 50.9335)	(8.7722, 10.8387)	2.2e-16

Table 3.10: Paired t-test results for each covariate after matching with MWPD (n=1696)

	t-statistics	mean of X	95% confidence interval	p-value for testing H_a
Sex	NaN	0	-	NA
Age	1.3425	0.0247	(-0.01139, 0.0608)	0.1796
Aps1	2.9839	0.05338	(0.01829, 0.08846)	0.0028

consider a two sample t -test to evaluate each covariate's balance in the full data set (Kleyman, 2009). The null hypothesis for each covariate tested is that there is no difference in population means on that covariate between the treatment and control groups. The test statistics for two sample t -test assuming unequal variance is

$$T = \frac{\bar{x}_T - \bar{x}_C}{\sqrt{\frac{s_T^2}{n_T} + \frac{s_C^2}{n_C}}}, \quad T \sim t\left(\frac{(s_T^2/n_T + s_C^2/n_C)^2}{(s_T^2/n_T)^2/(n_T - 1) + (s_C^2/n_C)^2/(n_C - 1)}\right)$$

After adjustment with pair matching, it is common to use the paired t-test as a way to assess the balance. The null hypothesis being tested for each covariate is the population mean difference on that covariate μ_D is equal to zero. The test statistic is

$$T = \frac{\bar{x}_D}{\frac{s_D}{\sqrt{n}}}, \quad T \sim t(n - 1)$$

Table 3.9 and Table 3.10 show results for the covariate balance test for each variable before matching and after matching with MWPD.

Table 3.9 shows a significant imbalance between the treatment and control groups for all covariates. Large t-statistics in the Aps1 variable indicates a high imbalance between

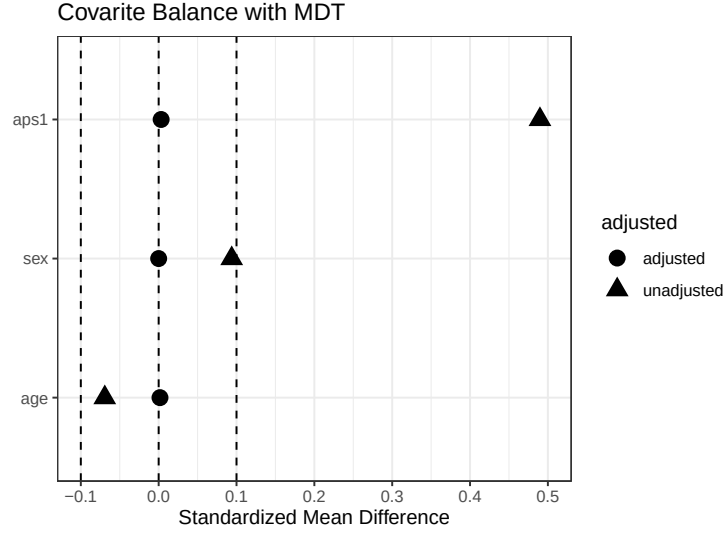


Figure 3.4: Standardized mean difference before and after matching

treatment and control groups in the Aps1 variable.

However, Table 3.10 shows that the sex variable perfectly matched with the same number of male and female units in the matched sample. The mean difference of the Age variable is insignificant at the significant level of $\alpha = 0.05$. Further, based on the resulted 95% confidence interval, the Aps1 variable has achieved a better balance than raw data.

Graphical diagnostics are also helpful for a quick assessment of the covariate balance. Hence, a plot of the standardized means differences is a visual representation to assess the balance. Figure 3.4 illustrates the standardized mean differences for before matching and after matching. The filled triangles represent the unadjusted (before matching) standardized mean difference, while the filled dots represent standardized mean differences on the covariates after matching with MWPD. It is clear from graph 3.4 that after matching, standardized differences (filled dots) are much closer to zero than unadjusted differences (filled triangles). Moreover, after matching with MWPD, the standardized means difference has decreased for each covariate significantly.

Suppose a researcher only uses optimal pair matching (without MWPD) to obtain the matched sample. Figure 3.5 visually summarizes covariates balance before and after matching with the `optmatch` R package (Hansen, 2007).

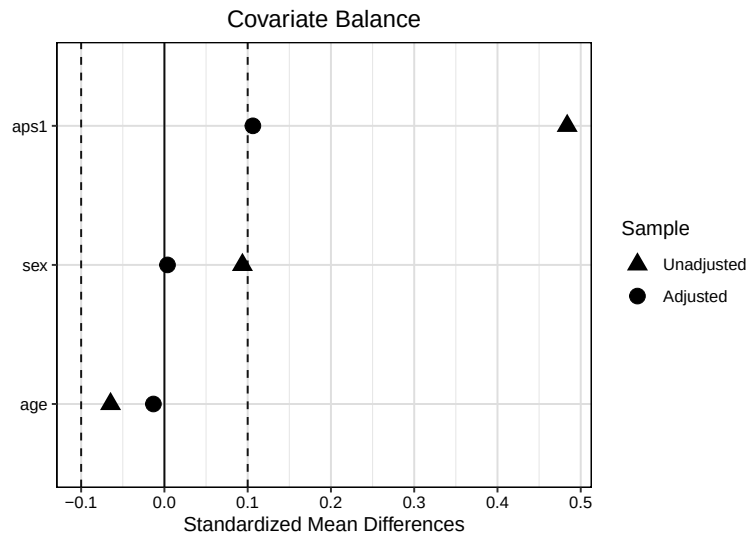


Figure 3.5: Standardized mean difference before and after matching

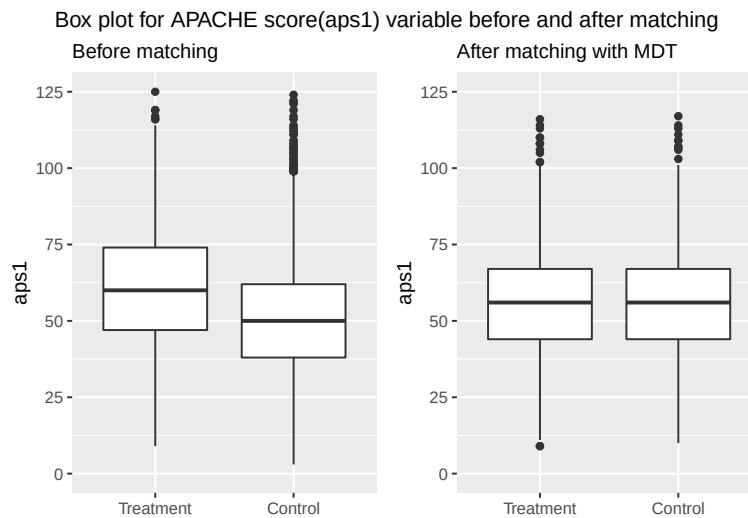


Figure 3.6: Box plot for APACHE score (aps1) variable before and after matching

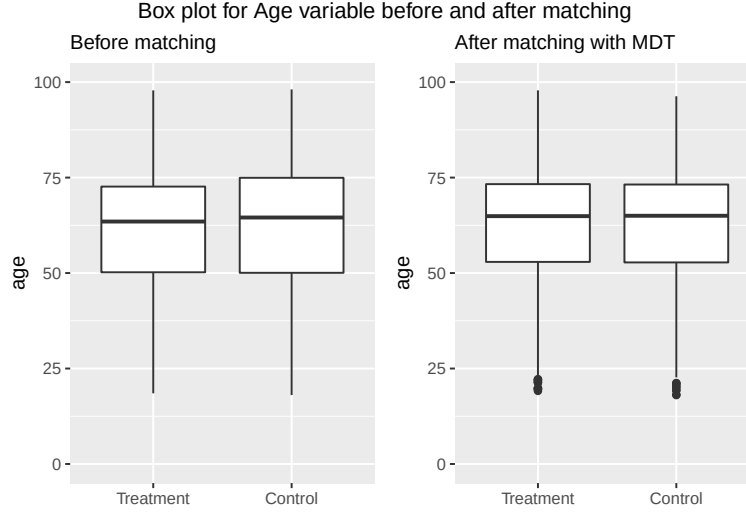


Figure 3.7: Box plot for Age variable before and after matching

The box plots that depict the distribution of APACHE score (aps1) and Age variables are presented in Figure 3.6 and Figure 3.7. For both variables, the proposed matching algorithm, optimal pair matching with MWPD, achieves a better balance. Additionally, variation in covariate balance, measured by the difference between the upper and lower quartiles (also referred to as the interquartile ranges) in each boxplot for the control group, becomes smaller with matching with MWPD.

3.6.2 Inflating the LCD to increase sample size

In the previous analysis, only 1696 of the 2184 treated units were included in the optimal match. Following Rosenbaum (2017), we examine how inflating can increase the number of units to be included in the study.

Table 3.11 shows the details about standardized mean differences for each covariate for four different calipers. The results show a significant difference in variables before matching with SMD greater than 0.1, a threshold being recommended for declaring imbalance.

After matching, the SMD values were less than 0.1 for all variables for any caliper within [1, 2.5], indicating a well-balanced sample. According to Table 3.11, by increasing the caliper from 1 to 1.5, 251 more unmatched treated units can be added to the matched sample. Also,

Table 3.11: Covariate balance(SMD) for different caliper with number of matched and unmatched treated units

	pair matching without MRPD	Pair Matching with MRPD			
		caliper =1	caliper=1.5	caliper=2	caliper=2.5
Matched Treated	2184	1686	1937	2046	2069
Unmatched Treated	0	498	247	138	115
age	0.012	0.002	0.002	0.005	0.005
sex	0.004	< 0.001	< 0.001	0.015	0.018
Aps1	0.112	0.003	0.009	0.025	0.034

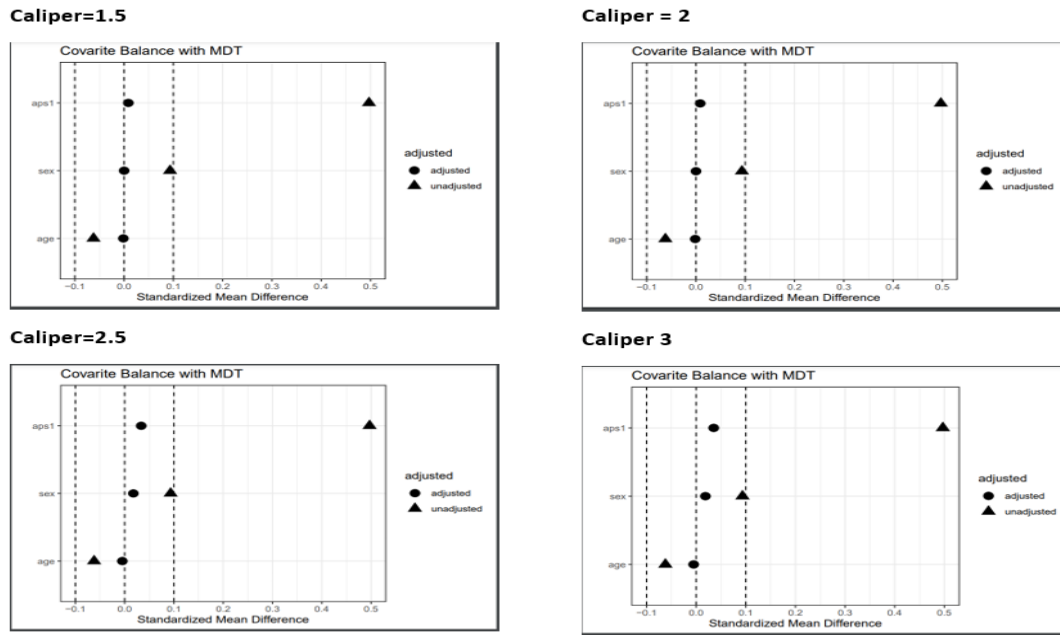


Figure 3.8: Covariate balance for three variables for different caliper values

compared to covariate balance in Figure 3.4, covariate balance in Figure 3.8 did not deviate significantly.

According to Figure 3.3, by making $\omega_{ij}^\infty < 2$, about 93.7% of the treatment units can be included in the matched sample. As a result, researchers are highly recommended to compare different calipers with their covariate balance (tests/ graphs) to decide the best caliper for a particular study.

According to Figure 3.9, tightening the MRPD value causes the covariate balance to

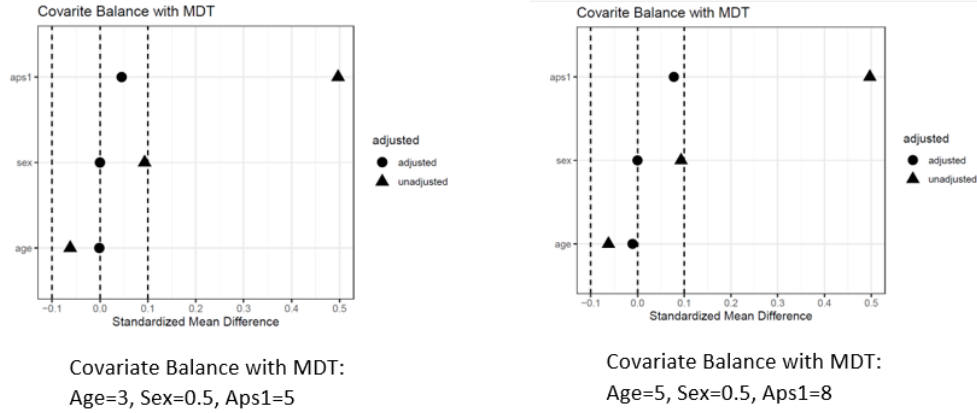


Figure 3.9: Covariate balance for three variables with different user-defined MWPD values near the covariate balance of optimal pair matching without MWPD, shown in Figure 3.5.

3.7 Conclusion

Constraining the maximum within-pair dissimilarity (MWPD) can produce a well-balanced matched sample while preventing poor matching in the matched sample. We propose a new method to find the optimal pair matching with MWPD; it is easy to implement, and the caliper can be changed according to researcher interests. Furthermore, the example on SUPPORT data demonstrates that optimal pair matching with MWPD helps solve the pair-matching problem in big data settings. Finally, the proposed new algorithm can be used to simultaneously create covariate balance for k covariates with a prespecified MWPD.

Chapter 4

Negative Alternating Paths of Even Length for the Minimum Weight-Maximum Cardinality Matching Problem

4.1 Introduction

Many matching methods have found their way into optimization theory. This development is natural since the matching problems are easy to formulate as combinatorial optimization problems. Hence, many statistical matching approaches explore a connection between statistics and optimization theory. Furthermore, developing new algorithms for statistical matching problems require expertise in statistics, optimization theory, and computer programming. Developing matching methods and implementing them with software tools benefit numerous research areas, including health, political science, social networks, etc.

This chapter proposes a new algorithm to solve the minimum cost-optimal pair matching problem with bottleneck constraints. The optimal pair matching with bottleneck constraints can run using the *rebalance* package and it allows adjacency list format to process a sparser

graph. Moreover, *rcbalance* use *rrelaxiv* and *rlemon* solvers to find the optimal pair matches in which the solution comes from the network flow framework. The proposed new algorithm in this study uses the adjacency list format but is not designed to use the network flow framework in the solution path. In general, the time required to find Maxcard is smaller compared to the time needed to find the Mincost in an optimal pair matching algorithm.

Maximum cardinality algorithms are briefly reviewed in Section 4.2. Usually using methods to find minimum cost algorithms are discussed in Section 4.3.1 and a brief explanation about the proposed new algorithm can be found in Section 4.3.2. Section 4.4 reviews notations and preliminaries for the proposed new algorithm with a theorem and proof. Section 4.6 has the simulation work for the proposed algorithm in 4.5. Finally, Section 4.7 includes our concluding remarks.

4.2 Maximum matching (MaxCard)

As previously stated, $G = (V = (V_T, V_C), E)$ be a bipartite graph, and matching $M \subset E$ contains matching edges such that no two edges in M have a vertex in common. An augmenting path is a path where the edges are alternating between matched and unmatched edges. Furthermore, many Maxcard algorithms are formed based on the augmenting paths in the residual network flow graphs. Augmenting path algorithms search for augmenting paths and augment until they find the maximum matching. A potential augmenting path in the residual graph can still increase the cardinality of the existing match. Hence, augmentation can be continued until no augmented path has been found in the residual graph to obtain the maximum cardinality.

Different algorithms use different methods for searching for augmenting paths, and thus, have different run times. For instance, the Ford Fulkerson algorithm can be used to find the maximum matching in a bipartite graph with n vertices and m edges in $\mathcal{O}(mn)$ time (Ford and Fulkerson, 1957). Hopcroft Karp's algorithm is another famous algorithm that can be used to find the maximum matching in a bipartite graph with $\mathcal{O}(m\sqrt{n})$ execution time (Hopcroft and Karp, 1973).

4.3 Minimum weight (MinCost)

4.3.1 Common approach

Once MaxCard is solved in an optimal pair matching problem, a cost-minimizing procedure can then apply to find the minimum total weight for that maximum matching. As previously discussed in Chapter 2, many MinCost procedures are implemented based on flow network frameworks. Recall that the cycle canceling with the Bellman-Ford algorithm intends to cancel negative cycles detected by the Bellman-Ford method to find MinCost in an optimal pair matching.

The Hungarian method is another way to compute the minimum weight perfect-matching in a weighted bipartite graph in $\mathcal{O}(mn)$ time (Kuhn, 1956). Also, the weight-scaling algorithm(or GT-Algorithm) can be used to find the minimum cost perfect matching in a weighted bipartite graph in $\mathcal{O}(m\sqrt{nC})$ time, where C is the bounded weight for all edges (Gabow and Tarjan, 1989).

Lahn and Raghvendra (2019) developed another algorithm to find minimum weight perfect bipartite matching in planner graph with $\mathcal{O}(n^{\frac{6}{5}})$ time, based on an r-clustering. In general, most algorithms designed to find minimum cost are implemented for perfect matches. If the algorithm failed to find any perfect matching, it is turned out to be reported as an infeasibility error; see Figure 4.1.

However, as discussed in Section 3.5.3, the `exclude.treated` feature in `rcbalance()` function plays a central role in finding optimal pair matching with unmatchable treatment units in statistical matching. The `exclude.treated` may facilitate removing treatment units if there is no possible match using all treatment units.

This study proposes a new algorithm to find the minimum cost-optimal pair matching using negative cycles and negative alternating paths of even length (NAPELs).

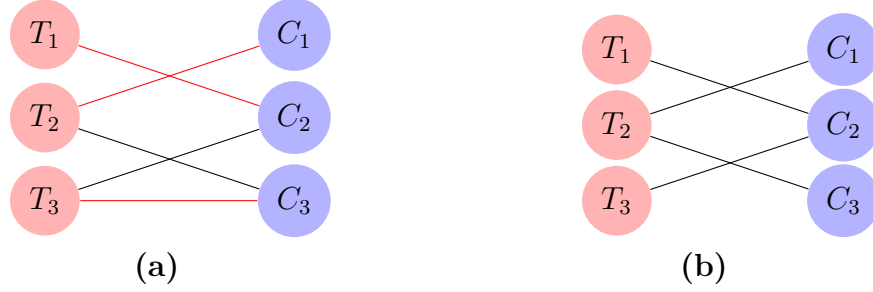


Figure 4.1: (a). Bipartite graph with feasible maximum cardinality pair matching. (b). The pair matching is infeasible for all treatment units

4.3.2 The proposed method

The proposed algorithm searches both negative cycles and NAPELs in the bipartite residual graph. Since this procedure uses a bipartite residual graph, it is not required to add the source(s) and sink(t) vertices to convert the graph into a commonly used network flow format. Moreover, adding extra edges from the source vertex to all treatment vertices and all control vertices to the sink vertex is avoidable.

The algorithm executes iterations for finding negative cycles first in the residual graph. The searching process may start with a matched control unit and must stop in the same matched control unit to complete the cycle with a total negative weight. The total weight may reduce further by flowing through another side of the cycle in the presence of a negative cycle. The residual graph can then update by swapping the unmatched and matched edges. Each cycle within the residual graph may have the same number of forward edges traveling from a treated unit to a control unit as backward edges traveling from a control unit to a treated unit. Thus, if the algorithm can find a negative cycle, the method can reduce total weight by swapping the unmatched and matched edges while MaxCard remains unchanged,

Next, the algorithm runs to search for NAPELs in the bipartite residual graph. The searching procedure may start from a matched control unit and alternate with matched and unmatched edges until it reaches an unmatched control unit with total negative weights in the path; see Figure 4.4. If the algorithm can find a NAPEL in the bipartite residual graph, the total weight can be further reduced by swapping the unmatched and matched edges without changing the Maxcard.

The searching process may continue until there is neither negative cycle nor NAPEL in the updated bipartite residual graph. Hence detecting negative cycles and NAPELs can reduce the total weight in the matched sample while keeping the same (maximum) cardinality. Detailed description of the algorithm may be found in section 4.5.

4.4 Preliminaries and notations for the proposed method

Definition 3. *A weighted bipartite graph*

Let $G = (V, E)$ be a weighted bipartite digraph composed of a weight function $w : E \rightarrow R$ and all edge capacities are equal to one since this study discusses the 1:1 matching problem.

Definition 4. *A residual bipartite graph*

Let $G = (V, E)$ be a weighted bipartite graph and let M be a matching in G . First transform G to a digraph $G' = (V, E'')$ —that is, each edge in E'' is now directed. Specifically, we allow edges in E'' to travel from a node in V_T to a node in V_C , but not the other direction: for $i \in V_T$, $j \in V_C$, and $ij \in E$, we have $\vec{ij} \in E''$, but $\vec{ji} \notin E''$. Given the matching M , the residual bipartite graph G_f is defined to be $G_f = (V, E_f)$ where,

$$E_f = \left\{ \vec{ij} \mid \vec{ij} \in E'' \right\} \cup \left\{ \vec{ji} \mid \vec{ij} \in M \right\}. \quad (4.1)$$

The residual capacities are one for all edges since we consider 1:1 matching for this study.

The weight function $w : E \rightarrow R$ is extended in to a weight function $w_{ij}^f : E_f \rightarrow R$ (see figure 4.2) as follows:

$$w_{ij}^f = \begin{cases} w_{ij}, & \vec{ij} \in E'' \text{ and } \vec{ij} \in M, \\ w_{ij}, & \vec{ij} \in E'' \text{ and } \vec{ij} \notin M, \\ -w_{ij}, & \text{otherwise.} \end{cases} \quad (4.2)$$

Definition 5. *Symmetric Difference Δ of two matching sets*

Let M and M' be two matching sets, then $M \Delta M' = (M - M') \cup (M' - M)$ and every component of the symmetric difference of two matching sets $M \Delta M'$ is a path or cycle (see

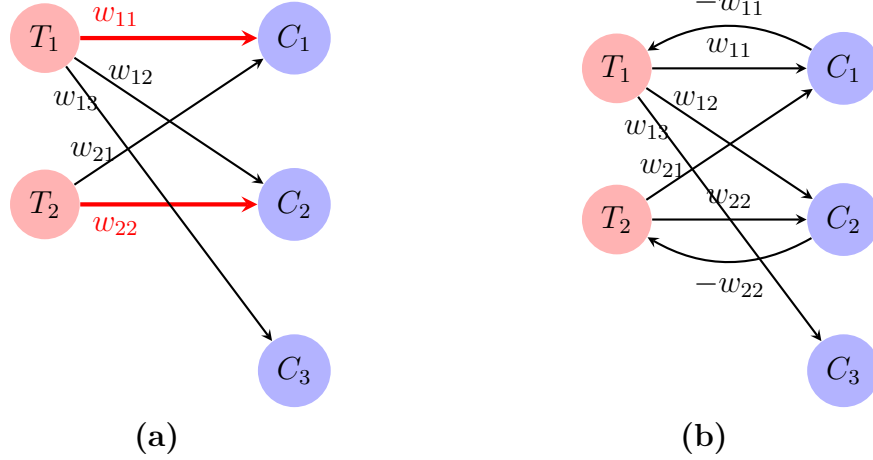


Figure 4.2: (a). Bipartite digraph with the initial matching. (b). The residual bipartite graph for graph (a). In Figure (b), if $-w_{11} + w_{12} - w_{22} + w_{21} < 0$, $C_1 - T_1 - C_2 - T_2 - C_1$ cycle is a negative cycle. In Figure (b), if $-w_{11} + w_{13} < 0$, $C_1 - T_1 - C_3$ path is a negative alternative path with even-length

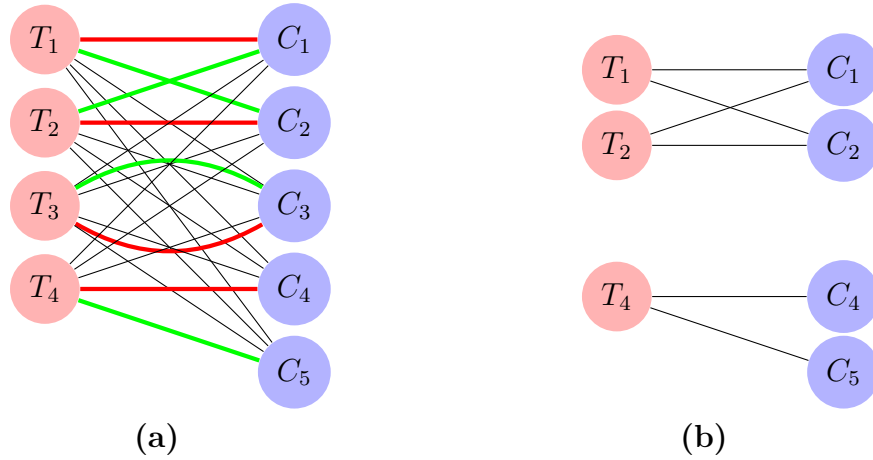


Figure 4.3: (a). Bipartite graph with two matching sets M and M' . (b). The symmetric difference of two matching sets $M \Delta M'$

Figure 4.3).

Definition 6. An Alternating augmenting paths and alternating cycles in a weighted bipartite graph with respect to a given matching M .

1. A directed path P start from a matched control unit and end in a unmatched control unit in G_f residual bipartite graph with $|P| > 2$ is said to be an alternating augmenting path.

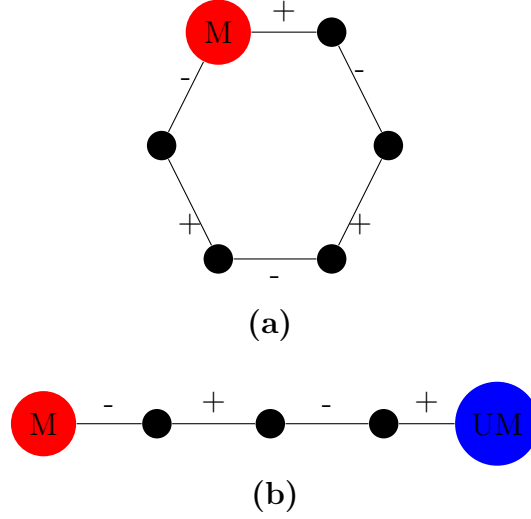


Figure 4.4: (a). Alternating cycle start from a matched control unit and ended up with the same control unit. (b). Alternating even path start from a matched control unit and ended up with an unmatched control unit

2. A directed path C started from a matched control unit and ended in the same control unit with $|C| > 2$ in G_f is said to be an alternating cycle.

Definition 7. Negative cycles and negative alternative path of even length

Let M be a matching in the weighted bipartite graph G .

1. Let C be an alternating cycle and “toggling” the edges of C with respect to M means removing the edges of $C \cap M$ and plugging in the edges of $C \setminus M$. This may yields a new matching M' that matches exactly the same set of vertices or another matching with same cardinality as of M .
2. Let P be an alternating path and “toggling” the edges of P with respect to M means removing the edges of $P \cap M$ and plugging in the edges of $P \setminus M$. This may yields a new matching M' that matches exactly the same set of vertices or another matching with same cardinality as of M .
3. An alternating cycle C is negative if the edges in the matching have higher cost than those outside the matching: $\sum_{ij \in C \cap M} w_{ij} > \sum_{ij \in C \setminus M} w_{ij}$.

Then, C is said to be a negative cycle.

4. An alternating path P is negative if the edges in the matching have higher cost than those outside the matching: $\sum_{ij \in P \cap M} w_{ij} > \sum_{ij \in P \setminus M} w_{ij}$.

Then, P is said to be a NAPEL.

Theorem 2. Let G be a weighted bipartite graph with the maximum cardinality matching M in G and the total dissimilarity $w(M)$. M is a minimum weight-maximum cardinality matching of G if and only if the residual bipartite graph G_f contains neither negative cycle C nor negative alternative path of even length P .

Proof. Suppose C is a negative cycle in weighted bipartite graph G with maximum cardinality matching M . If we toggle the edges of M with edges in negative cycle C to get another maximum cardinality matching M' , dropping the edges from $C \cap M$ saves us a cost of $\sum_{ij \in C \cap M} w_{ij}$, while adding the edges of $C \setminus M$ cost us $\sum_{ij \in C \setminus M} w_{ij}$. since $\sum_{ij \in C \cap M} w_{ij} > \sum_{ij \in C \setminus M} w_{ij}$, toggling the edges of the cycle yields a cheaper matching M' with same maximum cardinality ($|M| = |M'|$).

Similarly, suppose P is a NAPEL in weighted bipartite graph G with maximum cardinality matching M . If we toggle the edges of M with edges in NAPEL P to get another maximum cardinality matching M'' , dropping the edges from $P \cap M$ saves us a cost of $\sum_{ij \in P \cap M} w_{ij}$, while adding the edges of $P \setminus M$ cost us $\sum_{ij \in P \setminus M} w_{ij}$. since $\sum_{ij \in P \cap M} w_{ij} > \sum_{ij \in P \setminus M} w_{ij}$, toggling the edges of the cycle yields a cheaper matching M' with same maximum cardinality ($|M| = |M''|$).

Conversely, suppose M is a maximum cardinality matching in weighted bipartite graph G neither contains negative alternating cycle C nor NAPEL P in the residual bipartite graph G_f . Let M' be any other maximum cardinality matching; we want to show that the cost of M' is at least that of M . Consider $M \Delta M'$ meaning the symmetric difference of M , M' . In general, $M \Delta M'$ is a union of disjoint cycles and paths. Since taking the symmetric difference with the same set two times in a row recovers the initial set, $(M \Delta M') \Delta M' = M$. Since $M \Delta M'$ is a disjoint union of cycles and paths, taking the symmetric different with $M \Delta M'$ just means toggling the edges in each of its cycles or path. Each of these cycles and paths is alternating, and by assumption each is nonnegative. Thus toggling the edges of the

cycles or paths can only produce a more expensive maximum cardinality matching M' or $w(M') \geq w(M)$.

Hence, M must be a minimum-cost maximum cardinality matching in G . $\square$

4.5 Algorithm with NAPELs to solve Minimum weight- Maximum cardinality pair matching in a Bottle- neck subgraph

1. Initialize the dissimilarity measure w , a dissimilarity tolerance ω_k for all $k = 1, 2, \dots, K$.
2. Find the acceptable matches: Find all pairs of units with opposite treatment statuses that have dissimilarity less than ω .
3. Form a bottleneck subgraphs where edges join acceptable matches.
4. Perform Ford Fulkerson algorithm and find the maximum initial matching in the bottleneck subgraph.
5. Use the bipartite residual graph to check the existence of negative weight cycles and NAPELs in the graph.
6. If a negative cycle or NAPEL has been detected, remove it by flowing a bottleneck capacity through all the edges in the cycle or NAPEL respectively.
7. Update the bipartite residual graph and repeat Step 5 and 6 until there is no negative cycle or NAPEL in the updated residual bipartite graph.
8. Calculate the sum of weights for the final match in the graph from Step 7 and return.

Dataset	Number of Units		Total weight	
	Treated	Control	Proposed Method	Pair Matching Method
D1	300	300	29201	29201
D2	400	400	40172	40172
D3	500	500	51572	51572
D4	1000	1000	102741	102741
D5	2000	2000	201621	201621
D6	2500	2500	254515	254515
D7	3000	3000	302939	302939

Table 4.1: Simulated results for optimal pair matching and the proposed new algorithm with NAPELS(Step 4 to Step 8)

4.6 Simulation study

This simulation study is to access the performance of the proposed algorithm in Section 4.5. We generated data from the multinomial distribution. Each unit i is composed of two covariates, $X_i = (X_{i1}, X_{i2})$ where X_i was generated by multivariate normal distribution $N(\mu, \Sigma)$ where,

$$\mu = (0, 0), \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The weights were derived by Euclidean distances.

By using the total dissimilarity for optimal pair matching, the proposed algorithm is checked with the `pairmatch` function in `optmatch` R package (Hansen, 2007).

As shown in Table 4.1, the proposed algorithm can be used to find the optimal pair matching by using an adjacency list format.

4.7 Conclusion

If implemented carefully, the algorithm in Section 4.5 may solve the optimal pair matching problem efficiently. First, the MWPD constraint eliminates many edges in the original graph, resulting in a sparse graph to find the optimal pair matching. Second, the proposed algorithm uses the adjacency list format instead of the matrix format. Finally, the proposed algorithm

does not first transform the problem into a network flow problem, which may save some space and time. Hence, finding optimal pair matching in a sparse graph by inputting the least information in list format may result in a good time-bound for the proposed algorithm.

The implemented version gives promising run time for small data sizes. However, the run time increases as the size of data increases. The performance of the implemented version may increase by using the forest of alternating trees concept. For example, to solve the *weighted b-matching* problem, [Müller-Hannemann and Schwartz \(1999\)](#) used the concept of the alternating tree.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Statistical matching produces comparable treatment and control groups for treatment effect estimation while balancing confounding covariates in an observational study. While the greedy algorithm is one of the simplest algorithms to implement statistical matching, optimal matching has more solid reasons for gaining robust matching than greedy matching. The optimal pair matching is easy to perform with linear or integer programming approaches. However, when the data size is getting large and run time becomes an issue, finding an alternative algorithm is an exciting topic in the literature.

In modern society, large amounts of data are created every day. There are no standard techniques for designing an efficient algorithm for a matching problem for a given situation. But, developing an efficient algorithm for different types of matching problems is the key to effectively serving the statistical matching community for real-world applications.

Chapter 2 described the available algorithms and techniques for statistical matching. Several methods exist in optimization theory to improve and support statistical matching, but at present, adding some clarity to the black box of statistical matching and possibly enlightening the path towards future advances is essential. Also, this dissertation explains the importance of using adjacency list format in statistical matching than using the commonly

used dissimilarity matrix framework. Existing algorithms do not sufficiently facilitate ways to solve minimum weight-maximum cardinality matching on a sparse graph with an adjacency list format. The *rcbalance* is competitive compared to *optmatch*, which we accept as a good source for optimal pair matching in big data sets because of the adjacency list format. Finally, demystifying the existing matching algorithms will lead to new implementations in statistical matching. The discussion in Chapter 2 is then about valuable materials in statistical matching algorithms from the optimization theory side.

Chapter 3 of this dissertation presented a new method to obtain balance in k confounding covariates simultaneously. This chapter presented a novel approach that finds optimal pair matching within a bottleneck subgraph so that within-pair dissimilarity will not exceed the threshold value. Hence, the proposed bottleneck subgraph matching method with MDT may restrict the within-pair maximum dissimilarity in the final matched sample. Also, we applied the proposed approaches to a real case study using SUPPORT data. The illustration with SUPPORT data confirmed that the proposed method provides an effective way to restrict poor matching to being paired. More importantly, the proposed new concept of MDT is a researcher-oriented method and can be used to balance K covariates simultaneously.

Finally, Chapter 4 proposed a new algorithm for solving the minimum weight-maximum cardinality pair matching problem. The proposed method uses an adjacency list format to obtain the optimal pair matching without using a commonly used network flow framework with an adjacency matrix format. When the causal graph is getting bigger to construct in this setting, but we would like to make statistical matching, the combination of MDT and algorithm discussed in Section 4.5 will offer an advantageous way to a solution.

5.2 Future Work

This study has shown that few challenges need to be addressed in future work. This dissertation mainly focused on the use of bottleneck subgraph for pair matching without replacement to prevent poor matching with higher dissimilarity in the matched sample. It could be interesting to extend this study for matching with replacement but with some control over the number of times that one control unit can reuse in the matching process. The second challenge is improving the run time of the implemented version of the solution for the algorithm in fourth chapter. This improvement can be done by modifying the forest of alternating trees concept in the matching literature. For example, to solve the *weighted b-matching* problem, Müller-Hannemann and Schwartz (1999) used the concept of the alternating tree. Finally, as a third goal, to improve user friendliness and increased usage of our method, we aim to develop an R package for our method.

Bibliography

- Lemon 1.3.1 documentation. <https://lemon.cs.elte.hu/trac/lemon>. Accessed: 2022-07-02.
- Pankaj K Agarwal and R Sharathkumar. Approximation algorithms for bipartite matching with metric and geometric costs. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 555–564, 2014.
- Peter C Austin. Some methods of propensity-score matching had superior performance to others: results of an empirical investigation and monte carlo simulations. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 51(1):171–184, 2009.
- Peter C Austin. A comparison of 12 algorithms for matching on the propensity score. *Statistics in medicine*, 33(6):1057–1069, 2014.
- Peter C Austin and Dylan S Small. The use of bootstrapping when using propensity-score matching without replacement: A simulation study. *Statistics in medicine*, 33(24):4306–4319, 2014.
- Dimitri P Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.
- Dimitri P Bertsekas, Paul Tseng, et al. Relax-iv: A faster version of the relax code for solving minimum cost flow problems. 1994.
- Daniele Bottigliengo, Ileana Baldi, Corrado Lanera, Giulia Lorenzoni, Jonida Bejko, Tomaso Bottio, Vincenzo Tarzia, Massimiliano Carrozzini, Gino Gerosa, Paola Berchiolla, et al. Oversampling and replacement strategies in propensity score matching: a critical review focused on small sample size in clinical settings. *BMC medical research methodology*, 21(1):1–16, 2021.

- William G Cochran and Donald B Rubin. Controlling bias in observational studies: A review. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 417–446, 1973.
- Alfred F Connors, Theodore Speroff, Neal V Dawson, Charles Thomas, Frank E Harrell, Douglas Wagner, Norman Desbiens, Lee Goldman, Albert W Wu, Robert M Califf, et al. The effectiveness of right heart catheterization in the initial care of critically ill patients. *Jama*, 276(11):889–897, 1996.
- Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
- George B Dantzig. Origins of the simplex method. In *A history of scientific computing*, pages 141–151. 1990.
- Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Virkumar Vazirani. *Algorithms*. McGraw-Hill Higher Education New York, 2008.
- Rajeev H Dehejia and Sadek Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161, 2002.
- Alexis Diamond and Jasjeet S Sekhon. Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies. *Review of Economics and Statistics*, 95(3):932–945, 2013.
- Jack Edmonds and Ellis L Johnson. Matching: A well-solved class of integer linear programs. In *Combinatorial Optimization—Eureka, You Shrink!*, pages 27–30. Springer, 2003.
- Colin B Fogarty, Mark E Mikkelsen, David F Gaieski, and Dylan S Small. Discrete optimization for interpretable study populations and randomization inference in an observational study of severe sepsis mortality. *Journal of the American Statistical Association*, 111(514):447–458, 2016.
- Lester Randolph Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8:399–404, 1956.

- Lester Randolph Ford and Delbert R Fulkerson. A simple algorithm for finding maximal network flows and an application to the hitchcock problem. *Canadian journal of Mathematics*, 9:210–218, 1957.
- David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.
- Harold N Gabow and Robert E Tarjan. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036, 1989.
- Sebastian Galiani, Paul Gertler, and Ernesto Schargrotsky. Water for life: The impact of the privatization of water services on child mortality. *Journal of political economy*, 113(1):83–120, 2005.
- Richard E Gliklich, Michelle B Leavy, and Nancy A Dreyer. Tools and technologies for registry interoperability, registries for evaluating patient outcomes: A user’s guide, addendum 2 [internet]. 2019.
- Andrew V Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of algorithms*, 22(1):1–29, 1997.
- Andrew V Goldberg and Robert E Tarjan. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM (JACM)*, 36(4):873–886, 1989.
- Ben B Hansen. Optmatch: Flexible, optimal matching for observational studies. *New Functions for Multivariate Analysis*, 7(2):18–24, 2007.
- Jennifer Hill and Yu-Sung Su. Assessing lack of common support in causal inference using bayesian nonparametrics: Implications for evaluating the effect of breastfeeding on children’s cognitive outcomes. *The Annals of Applied Statistics*, pages 1386–1420, 2013.
- Daniel E Ho, Kosuke Imai, Gary King, and Elizabeth A Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political analysis*, 15(3):199–236, 2007.

- Paul W Holland and Donald B Rubin. Causal inference in retrospective studies. *ETS Research Report Series*, 1987(1):203–231, 1987.
- John E Hopcroft and Richard M Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.
- Stefano M Iacus, Gary King, and Giuseppe Porro. Multivariate matching methods that are monotonic imbalance bounding. *Journal of the American Statistical Association*, 106(493):345–361, 2011.
- Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.
- Morton Klein. A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220, 1967.
- Yevgeniya N Kleyman. *Testing for covariate balance in comparative studies*. PhD thesis, University of Michigan, 2009.
- Péter Kovács. Minimum-cost flow algorithms: an experimental evaluation. *Optimization Methods and Software*, 30(1):94–127, 2015.
- Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- Harold W Kuhn. Variants of the hungarian method for assignment problems. *Naval research logistics quarterly*, 3(4):253–258, 1956.
- Nathaniel Lahn and Sharath Raghvendra. A faster algorithm for minimum-cost bipartite matching in minor-free graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 569–588. SIAM, 2019.
- Danyu Y Lin, Bruce M Psaty, and Richard A Kronmal. Assessing the sensitivity of regression results to unmeasured confounders in observational studies. *Biometrics*, pages 948–963, 1998.

- Sharif Mahmood. The performance of largest caliper matching through monte carlo simulation and an application to support data. *Journal of Data Science*, pages 345–360, 2020.
- Matthias Müller-Hannemann and Alexander Schwartz. Implementing weighted b-matching algorithms: Towards a flexible software design. *Journal of Experimental Algorithmics (JEA)*, 4:7–es, 1999.
- John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- Samuel D Pimentel, Rachel R Kelz, Jeffrey H Silber, and Paul R Rosenbaum. 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(510):515–527, 2015.
- Samuel D Pimentel, Maintainer Samuel D Pimentel, et al. Package ‘rcbalance’. 2022.
- Edward M Reingold and Robert E Tarjan. On a greedy heuristic for complete matching. *SIAM Journal on Computing*, 10(4):676–681, 1981.
- Paul R Rosenbaum. Optimal matching for observational studies. *Journal of the American Statistical Association*, 84(408):1024–1032, 1989.
- Paul R Rosenbaum. Imposing minimax and quantile constraints on optimal matching in observational studies. *Journal of Computational and Graphical Statistics*, 26(1):66–78, 2017.
- Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.
- Paul R Rosenbaum, Richard N Ross, and Jeffrey H Silber. 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, 2007.

- Paul R Rosenbaum et al. *Design of observational studies*, volume 10. Springer, 2010.
- Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.
- Fredrik Sävje. Consistency and matching without replacement. *arXiv preprint arXiv:1907.07288*, 2019.
- Fredrik Sävje, Michael J Higgins, and Jasjeet S Sekhon. Generalized full matching. *Political Analysis*, 29(4):423–447, 2021.
- Jasjeet S Sekhon. Opiates for the matches: Matching methods for causal inference. *Annual Review of Political Science*, 12(1):487–508, 2009.
- Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pages 465–472, 1990.
- Giancarlo Visconti and José R Zubizarreta. Handling limited overlap in observational studies with cardinality matching. *Observational Studies*, 4:217–249, 2018.
- Dan Yang, Dylan S Small, Jeffrey H Silber, and Paul R Rosenbaum. Optimal matching with minimal deviation from fine balance in a study of obesity and surgical outcomes. *Biometrics*, 68(2):628–636, 2012.
- MJ Yetley. Causal model analysis: a particular use of regression analysis. *Irish Journal of Agricultural Economics and Rural Sociology*, pages 117–133, 1969.
- Ruoqi Yu. *Contributions to Multivariate Matching in Observational Studies*. PhD thesis, University of Pennsylvania, 2021.
- José R Zubizarreta. 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, 2012.

- José R Zubizarreta, Caroline E Reinke, Rachel R Kelz, Jeffrey H Silber, and Paul R Rosenbaum. Matching for several sparse nominal variables in a case-control study of readmission following surgery. *The American Statistician*, 65(4):229–238, 2011.
- José R Zubizarreta, Ricardo D Paredes, Paul R Rosenbaum, et al. Matching for balance, pairing for heterogeneity in an observational study of the effectiveness of for-profit and not-for-profit high schools in chile. *Annals of Applied Statistics*, 8(1):204–231, 2014.
- JR Zubizarreta, C Kilcioglu, and JP Vielma. designmatch: Matched samples that are balanced and representative by design. *R package version 0.3*, 1, 2018.

Appendix A

Cardinality matching followed by minimum distance pairing with Bottleneck constraint

A study about cardinality matching followed by minimum distance pairing was proposed by Jose R. Zubizarreta, Ricardo D. Paredes and Paul R. Rosenbaum ([Zubizarreta et al., 2014](#)). The proposed optimal cardinality matching solves an Integer Programming problem to maximize the cardinality or size of the matched samples subject to constraints on covariate balance. The covariate balance was obtained by linear inequalities in function of the observed covariates.

As the constraints on average covariate balance alone is not sufficient, this study suggest to add bottleneck constraint to the cardinality matching model.

First we will consider the Cardinality matching model without the bottleneck constraint. So the optimization problem can be written as follows:

$$\begin{aligned} & \max \sum_{i=1}^t \sum_{j=1}^c z_{ij} \text{ subject to } z_{ij} \in \{0,1\}, i = 1, \dots, t, j = 1, \dots, c, \\ & \text{with the constraints,} \\ & \sum_{i=1}^t z_{ij} \leq 1 \text{ for } j = 1, \dots, c, \\ & \sum_{j=1}^c z_{ij} \leq 1 \text{ for } i = 1, \dots, t, \end{aligned}$$

$$\mathbb{B}_k, k = 1, \dots, K$$

Here, $\mathbb{B}_k : -b_k \sum_{i=1}^t \sum_{j=1}^c z_{ij} \leq \sum_{i=1}^t \sum_{j=1}^c z_{ij} v_{kij} \leq b_k \sum_{i=1}^t \sum_{j=1}^c z_{ij}$,

for $k = 1, \dots, K$ with $v_{kij} = f(X_{V_{Ti}}) - f(X_{V_{Cj}})$ for some function $f(\cdot)$. Where v_{kij} is the k^{th} of K functions of observed covariates and $b_k \geq 0$ is a given constant.

Let's consider the Cardinality matching model with the bottleneck constraint. The optimization problem can be adjust as follows:

$$\max \sum_{i=1}^t \sum_{j=1}^c z_{ij} \text{ subject to } z_{ij} \in \{0,1\}, i = 1, \dots, t, j = 1, \dots, c,$$

with the constraints,

$$\sum_{i=1}^t z_{ij} \leq 1 \text{ for } j = 1, \dots, c,$$

$$\sum_{j=1}^c z_{ij} \leq 1 \text{ for } i = 1, \dots, t,$$

$$\mathbb{B}_k, k = 1, \dots, K$$

$$\max \left(\frac{|v_{ktc}|}{\delta_k} \right) \leq 1, k=1, 2, \dots, K.$$

Here,

$$\mathbb{B}_k : -b_k \sum_{i=1}^t \sum_{j=1}^c z_{ij} \leq \sum_{i=1}^t \sum_{j=1}^c z_{ij} v_{kij} \leq b_k \sum_{i=1}^t \sum_{j=1}^c z_{ij},$$

for $k = 1, \dots, K$ with $v_{kij} = f(X_{V_{Ti}}) - f(X_{V_{Cj}})$ for some function $f(\cdot)$.

This optimization model can be rectified as,

$$\max \sum_{i=1}^t \sum_{j=1}^c z_{ij} \text{ subject to } z_{ij} \in \{0,1\}, i = 1, \dots, t, j = 1, \dots, c,$$

with the constraints,

$$\sum_{i=1}^t z_{ij} \leq 1 \text{ for } j = 1, \dots, c,$$

$$\sum_{j=1}^c z_{ij} \leq 1 \text{ for } i = 1, \dots, t,$$

$$\mathbb{B}_k, k=1, \dots, K,$$

$$\mathbb{G}_k, k=1, \dots, K.$$

Here,

$$\mathbb{B}_k : -b_k \sum_{i=1}^t \sum_{j=1}^c z_{ij} \leq \sum_{i=1}^t \sum_{j=1}^c z_{ij} v_{kij} \leq b_k \sum_{i=1}^t \sum_{j=1}^c z_{ij},$$

$$\mathbb{G}_k : -\delta_k \leq z_{ij} v_{kij} \leq \delta_k$$

for $k = 1, \dots, K$ with $v_{kij} = f(X_{V_{Ti}}) - f(X_{V_{Cj}})$ for some function $f(\cdot)$. Where v_{kij} is the k^{th} of K functions of observed covariates and $\delta_k \geq 0$ is a given constant.

The bottleneck constraints set $\mathbb{G}_k$ will prevent having some arbitrarily poor matches in M . This set of K bottleneck constraints will force the matched sample to have maximum dissimilarity in the treated and control groups within $[-\delta_k, \delta_k]$ interval; here δ_k is a constant value fixed by the researcher.

Appendix B

Simulation study for optimal pair matching problem using Ford Fulkerson and cycle cancelling algorithms

By combining Ford Fulkerson algorithm, cycle cancelling with Bellman Ford algorithm can create the minimum weight-maximum cardinality bottleneck subgraph matching algorithm as follow:

1. Perform Ford Fulkerson algorithm and find the maximum flow of the graph.
2. Use the Bellman - Ford algorithm on the resulted graph from step 1 to check the existence of negative weight cycles in the graph.
3. If a negative cycle has been detected in Step 2, remove it by flowing a bottleneck capacity through all the edges in the cycle.
4. Update the residual graph and repeat Step 2 and 3 until there is no negative cycles in the updated residual graph.

Dataset	Number of Units		Elapsed Time		Total weight	
	Treated	Control	MWMC	PM	MWMC	PM
D1	5	5	3.70	0.78	7.65379	7.65379
D2	50	50	2.98	0.77	67.39048	67.39048
D3	250	250	4.89	0.82	323.2791	323.2791
D4	500	500	22.61	0.70	657.2276	657.2276
D5	1000	1000	276.58	12.14	1292.092	1292.092
D6	2000	2000	1751.09	104.64	2642.193	2642.193
D7	2500	2500	3693.38	297.62	3260.922	3260.922
D8	3000	3000	6461.95	823.50	3932.748	3932.748
D9	3400	3400	7938.39	-	4358.455	-
D10	5000	5000	10942.27	-	8742.735	-

Table B.1: Simulated results for optimal pair matching(PM) and minimum weight-maximum cardinality method(MWMC).

5. Calculate the sum of weights for the flow in the graph from Step 4 and return.

We generated data from the multinomial distribution. Each unit i is composed of two covariates, $X_i = (X_{i1}, X_{i2})$ where X_i was generated by multivariate normal distribution $N(\mu, \Sigma)$ where,

$$\mu = (0, 0), \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The weights were derived by Euclidean distances. The performance was evaluated by using the `pair match` function in `optmatch` R package.

As an initial step in the implementation of maximum cardinality- minimum weight matching algorithm developed in R. R is a open source programming language mainly used for statistical computing. Since we are experiencing rapid growth in the size of the data sets, the simulation will use the provided application programming interface(API) for extension and benchmark R with C++. Typically, C++ can address problems that require advanced data structures and algorithms that R does not provide. Rcpp, written by Dirk Eddelbuettel and Romain Francois, makes it very simple to connect C++ to R. Simulated results are presented in Table B.1. Table B.1 shows the results of the algorithm run time and minimum weights. In general, the current implementation of optimal matching takes a smaller amount

of run time compared to the proposed algorithm. The main reason to have a longer run time is, this implementation is not the best solution and needs to improve R-code to find Maximum cardinality- Minimum weight matching with enhanced run time. But it can create a competitive algorithm since the proposed algorithm can work on both a sparse graph and a complete bipartite graph. Also, the resulted minimum total weight is the same for both methods. Thus these results are promising to have optimal matching on a sparse graph by improving the R function.

The most important fact is that for Dataset 9 and Dataset 10, the Pair matching function doesn't have an answer since the data set exceeds the limit of the space. But the proposed algorithm can handle more units than the existing method of Pair matching.

Appendix C

Detail description for RHC Data

RHC is a diagnostic procedure used for critically ill patients. The SUPPORT study collected data on hospitalized adult patients at 5 medical centers in the U.S. Based on information from a panel of experts a rich set of variables relating to the decision to perform the RHC, as well as detailed outcome data, were collected. Table in below lists the covariates we use in our analysis. Further information about the study can be found in Connors et al. (1996) and Murphy and Cluff (1990).

SUPPORT covariates

age	Age (years)	dnr1	Do Not Resuscitate status on day 1
sex	Female	aps1	APACHE score
raceblack	Black	scoma1	Glasgow coma score
raceother	Other	wtkilo1	Weight
edu	Education (years)	temp1	Temperature
income1	Income 11–25k	meanbp1	Mean Blood Pressure
income2	Income 25–50k	resp1	Respiratory Rate
income3	Income > 50k	hrt1	Heart Rate
ins-care	Medicare	pafi1	PaO ₂ =FI _O ₂ ratio
ins-pcare	Private and Medicare	paco21	PaCO ₂
ins-caid	Medicaid	ph1	PH
ins-no	No Insurance	wblc1	WBC
ins-careaid	Medicare and Medicaid	hema1	Hematocrit
cat1-copd	COPD	sod1	Sodium
pot1	Potassium	ortho	Orthopedic diagnosis
cat1-mosfsep	MOSF w=Sepsis	crea1	Creatinine
cat1-mosfmal	MOSF w=Malignancy	bili1	Bilirubin
cat1-chf	CHF	alb1	Albumin
cat1-coma	Coma	chfhx	Congestive Heart Failure
cat1-cirr	Cirrhosis	card	Cardiovascular diagnosis
cat1-lung	Lung Cancer	neuro	Neurological diagnosis
cat1-colon	Colon Cancer	gastr	Gastrointestinal diagnosis
cat2-mosfsep	MOSF w=Sepsis	renal	Renal diagnosis
cat2-coma	Coma	liverhx	Cirrhosis, hepatic failure
cat2-mosfmal	MOSF w=Malignancy	gibledhx	Upper GI bleeding
cat2-lung	Lung Cancer	meta	Metabolic diagnosis
cat2-cirr	Cirrhosis	ca-meta	Cancer — metastatic
cat2-colon	Colon Cancer	ca-yes	Cancer — localized
resp	Respiratory diagnosis	wt0	weight = 0
hema	Hematological diagnosis	seps	Sepsis diagnosis
trauma	Trauma diagnosis	amihx	Definite myocardial infarction
das2d3pc	DASI — Duke Activity Status Index	surv2md1	Estimate of prob. of surviving 2 months
transhx	transfer (> 24 hours) from another hospital	renalhx	Chronic renal disease, chronic hemodialysis or peritoneal dialysis
dementhx	Dementia, stroke or cerebral infarct, Parkinson's disease	psychhx	Psychiatric history, active psychosis or severe depression
malighx	Solid tumor, metastatic disease, chronic leukemia/myeloma, acute leukemia, lymphoma	immunhx	Immunosuppression, organ transplant, HIV, Diabetes, Mellitus, Connective Tissue Disease
chrpulhx	Chronic pulmonary disease, severe pulmonary disease		