DATA ANALYTICS IN COMPLEX DATA ENVIRONMENTS: METHODS TOWARDS MISSING VALUES AND DYNAMIC DATA PATTERNS

PENG JIAXU

(M.M. in MS&E, Beihang University)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF INFORMATION SYSTEMS AND ANALYTICS
NATIONAL UNIVERSITY OF SINGAPORE

2020

Supervisor:
Associate Professor Hahn Jungpil

Examiners:
Associate Professor Rudy Setiono
Assistant Professor Vaibhav Rajan
Associate Professor Jingjing Zhang, Indiana University

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Peng Jiaxu

18 May 2020

ACKNOWLEDGEMENTS

It would not have been possible to accomplish this thesis without the help and support I received during my Ph.D. journey. I take this opportunity to express my heartful thanks, although it is possible to only mention some of them here.

First of all, I would like to express my deepest gratitude to my thesis advisor, Professor Jungpil Hahn for his insightful guidance, valuable advice, and unwavering encouragement. When I was entering the final stage of developing my Ph.D. works into the thesis, I faced challenges in further enhancing and positioning my Ph.D. works. He gave me immense support so that I regain the confidence to reboot my research and finish my thesis. Every research discussion with him returned me into a more sharpened and deepened-thinking mind. His insights in multiple aspects such as research theorization and research design not only help me to finish my thesis, but also largely enrich and enhance my research. He motivated me to be the best that I could be. I learned from his rigor and passion towards impactful research, his kindness towards everyone, his open mind and great courage to break through the challenges and to make contribution to our societal community.

I would also like to thank my former advisor Professor Ke-Wei Huang who has supported me during the development of my Ph.D. studies over the years. He has spent his valuable time patiently with me and provided very helpful comments in enhancing my research. He motivated me to expand my knowledge and abilities. His passion in seeking out important research questions and targeting at research challenges continues to inspire me.

Next, I would like to convey my profound thanks to the members of my thesis committee, Professors Rudy Setiono and Vaibhav Rajan. Their constructive comments and insightful criticism have been extremely helpful. They helped me to refine my dissertation and to develop better ways to present the overall research

framework. I would also like to extend my appreciation to Professor Khim Yong Goh and Professor Tuan Quang Phan. During the early development of my research as a graduate research paper, they provided very helpful feedback and pointed me to informative directions in further improving the research method of my study.

No journey is complete without a supportive and welcoming environment. I would like to express my thanks to all faculty members of NUS School of Computing, who are always passionate about providing support and ready to share knowledge and comments. I also gratefully acknowledge the support that I received from the administrative and technical staff at the School of Computing, NUS IT, and many other divisions who provide critical support during my studies. My heartful thanks to my peers. It has been a great journey accompanied with supporting and responsible young scholars.

Last but not the least, thanks to my parents. They cultivated the confidence and energy in me from a very young age so that I always seek to see the positive side and actively respond to challenges in my life. I also especially thank my friend Dr. Fan Feng who gives me immense support during my Ph.D. journey of knowledge and exploration. Without them, I would not reach this point in my life.

To All, Thank You.

TABLE OF CONTENTS

SUMMARY

The rapid accumulation of data and advances in data analytics methods create not only opportunities but also challenges for data analytics. One fundamental challenge arises from the heterogeneity in data patterns. This thesis investigates two frequently recurring problems that result in hard-to-be-observed data heterogeneity: missing values (the focus of the first study), and dynamic changing data patterns (the focus of the second study).

In my first study entitled "Handling Missing Values without Assuming Missing at Random," I propose approaches to handling missing values that occur not at random. Traditional imputation models are often built on complete records – i.e., records in the dataset without missing values. However, if missing values do not occur at random, estimates of parameters would be biased. In the proposed approaches, including a missing value imputation method based on semi-supervised learning, and a Monte Carlo likelihood estimation approach for correcting estimation bias caused by missing values, I explicitly incorporate the missingness mechanism into the data analytics process. I analytically demonstrate that, accommodating the missingness mechanism generates comparatively better imputation and statistical estimates than traditional methods that ignore the missingness mechanism. In the context of two real-world prediction tasks, results show that the proposed semi-supervised missing value imputation generates higher prediction accuracy compared to benchmark imputation methods. In the bias correction problem of regression analysis, the proposed Monte Carlo based approach generates unbiased estimation of regression coefficients under different missingness mechanisms.

My second study entitled "Transfer Learning in Dynamic Business Environments: Trade-offs in Response to Changes" takes up the challenge that, in changing data environments, we often have little information to adjust statistical prediction models in a timely matter. In this study, I investigate the question of

whether and how we can make use of all of the source data (including the same-distribution recent source data and the remaining diff-distribution past source data) to achieve better prediction accuracy for a target task when there is only a small amount of same-distribution source data that exhibits the target data pattern. In this study, I bridge the research gap in theoretically understanding when and to what extent transfer learning works by using a sample selection perspective to represent changes in data patterns. Based on the sample selection model, I derive a probabilistic weighting scheme using the large source data set. Moreover, I conduct simulation analyses to examine two fundamental trade-offs when changes are detected – 1) whether we should use transfer learning to adjust the prediction model, and 2) whether we should adjust the prediction model immediately or at a later point in time when more same-distribution source data becomes available. The results, implications, and contributions are discussed.

Throughout my dissertation, I seek to understand the underlying mechanisms of the heterogeneity in data patterns arising from missing values and changing data environments, and to provide theoretical insights on how to approximate and make use of the often-overlooked mechanisms. By unveiling the underlying theory and assumptions, this dissertation promotes more robust application of data analytics in complex data environments.

Keywords: missing values, data quality, missingness mechanism, not missing at random, semi-supervised learning, Monte Carlo, maximum likelihood, transfer learning, predictive analytics, machine learning, sample selection, dynamic data environments

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION AND OVERVIEW

1.1 Research Background and Motivation

Advances in data analytics techniques and rapid emergence of big data have created new opportunities for uncovering hidden knowledge, improving decision making, and supporting strategic planning in various business applications (Chen et al. 2012; Chiang et al. 2018). Nowadays, with increasing accessibility of data analytics tools, managers and analysts can quickly build models and discover patterns in data that can aid decision making. However, challenges remain in achieving valid inferences and maintaining the viability of our models in the complex data environments. In my dissertation, I investigate two problems that add complexity to the data: missing values and the dynamic data patterns in business environments.

An important driver of the challenges in data analytics comes from unobservability of the data generating process. In data analytics, analysts approximate real-world phenomena by building statistical models. With more advanced machine learning and econometrics models, it is possible to approximate reality to a large extent. In general, data analytics is built upon the assumption that the data follows a uniform data pattern (even if the data pattern per se is assumed to be a complex one such as in multimodal analysis). However, there exist scenarios where the heterogeneity in data patterns exists. Such heterogeneity may result in a distorted data pattern being investigated. In the missing values problem, the heterogeneity in data patterns arises from the potential differences between the observable and unobservable information. In dynamic changing data environments, the heterogeneity stems from the fact that the model developed using the source data may no longer fit the unseen target data (i.e., the source data is different to the target data).

Missing values are unobservable information in our databases. For example, respondents in a household survey may refuse to report income (Little and Rubin

2014); patients' medical records may be subject to missing values since certain medical tests are not administered to all patients (Hall et al. 2007). In these examples, it is natural to treat the values that are not observed as missing, in the sense that there are actual underlying values that would have been observed if respondents made full disclosure during the survey or the medical tests had been administered.

The fundamental problem with missing values is that it is not possible to know whether the unobserved missing values follow the same data pattern as the observed values. This results in uncertainty with various aspects of data analytics such as model specification and distributional assumptions. If the purpose of the data analysis is to uncover patterns for the overall data, using only the observed information will inevitably generate partial insights and may even distort the inferences we generate from the observed patterns. For instance, in estimating the average income for a consumer group, if the income variable is more likely to be missing when it is larger, then the estimated mean using the observed values would be downwardly biased. Therefore, given an incomplete data set, analysts need to consider and reflect on whether and how the missing values reshape the data pattern we observe.

With the presence of dynamically changing data patterns, the source data, which often consists of historical records, may not represent the same data pattern as the target data, which involves the variable to be predicted but is realized in future unseen data. Data analytics heavily rely on statistical analysis (Chen et al. 2012). In traditional data analysis tasks, such as developing a prediction model using supervised machine learning, we use the source data to obtain a function or a prediction model that maps from predictors to the variable being predicted, and then apply the built model to target data to make predictions of future events or of variables of interest. This practice of data analytics assumes that the data being analysed follows a stable pattern (Pan and Yang 2010; Zhang et al. 2017).

In typical business environments, this assumption is violated more often than not. For instance, in predicting firms' future earnings during recession periods, both the distribution of predictors and the functional relationship between predictors and the dependent variable is likely to change (Li and Mohanram 2014). Ignoring the potential changes in data pattern may jeopardize the validity of our models and the conclusions we draw from the data analysis.

Throughout this dissertation, I hold the argument that the approximate nature of data analytics models must always be borne in mind and we should always be concerned about the often over-looked heterogeneity in data patterns and the mechanisms that influence what information is being observed and therefore what data patterns are uncovered.

1.2 Challenges

The heterogeneity in data patterns arising from missing values and changing data patterns is difficult, if not impossible, to be examined directly. This is due to limited or even no information that can be used in understanding how the data patterns might have changed from observed to unobserved information, or between the source data and target data. In the missing values problem, to investigate whether and how the unobserved information is different to the observed information, we can run a statistical test, such as the mean difference test, between the observed and unobserved values. However, the unobserved values are unknown in the first place. In dynamic changing environments, even if it is strongly suspect that we have entered into a new data regime due to a change in policy or changes in the economic environment, we have little information in identifying and quantifying the changes in the data patterns at the early stages of the new data regime.

Perhaps one of the most reliable and rigorous ways to identify the heterogeneity in data patterns is to use interventions to expand the available information, such as gathering the underlying truth of missing values (Glynn et al.

1993), and obtaining more data examples for the target domain (Pan and Yang 2010). However, the interventions or additional data collection are often cost prohibitive or even impossible in some data analytics contexts. For instance, to unveil the missing values in patients' health record, additional medical tests could be administered; however, medical resources are often limited (Zhang et al. 2005). In dynamic data environments, since data generation takes place in the business environment (rather than in a controlled laboratory setting), analysts may not actively collect data but rather wait to see additional information generated in the new data regime. Thus, there emerges a trade-off between sparse information and lagged response.

Given the practical difficulties in augmenting the data, research attention has been focused on how to enhance the validity and viability of statistical models using the available observational data. The open problems still include the design of theoretically and computationally appropriate methods that apply to more general situations so that analytics methods would be more robust to the difficult-to-be-observed heterogeneity in data patterns. Moreover, the methods should be able to capture what pieces of information are useful to depict the underlying process and find out the ways to employ such information in business analytics contexts.

1.3 Research Objectives and Methods

Researchers have long been exploring how the often-over-looked heterogeneity in data patterns influences the validity of research results. To solve such problems, one needs to gain insights into the underlying process behind the given data. Both the missing values and the data mining literatures encompass different perspectives to explore this problem. In this section, I provide an overview of the problems being investigated, motivate research objectives by identifying research gaps, and describe research methods employed in my thesis.

1.3.1 Handling Missing Values Not at Random

The missing values problem is ubiquitous in many application contexts. For instance, in electronic commerce, customers frequently neglect to provide ratings for products they have purchased and consumed (Ying et al. 2006), which results in missing values for product recommendation systems. In survey-based empirical research, researchers often have to deal with missing values (i.e., skipped responses) for certain items in their questionnaires (Sivo et al. 2006). Even empirical research using firm-level archival databases such as Compustat (e.g., Havakhor et al. 2019) frequently encounter missing values for certain important data fields such as research and development (R&D) expenses (Koh and Reeb 2015).

Missing values introduce two potential problems in statistical analysis – *information loss* and *biasedness*. Information loss is an obvious outcome. From a statistical perspective, it is reflected by increased variance (uncertainty) of statistical estimates compared to the estimates that would have been obtained if there were no missing values.[1] Biasedness typically arises when the observed values differ systematically from the unobserved missing values. For instance, in the simple estimation of the mean of a variable, if the variable is more likely to be missing when it is larger, then the estimated mean using the observed values would be downwardly biased. Even though the volume of data is often sufficiently large (or even excessive) in today's big data environment, minimizing information loss is still a common challenge in data analytics. Moreover, biasedness (or inconsistency) arguably raises much greater concerns for data analysts since it cannot be resolved by simply increasing the sample size.

Among various methods for handling missing values, the traditional and one of the most popular methods is *case deletion* (also known as *listwise deletion*, or

[1] Little and Rubin (2014, in Chapter 3.2), define metrics to measure the proportional increase in variance from the loss of information.

complete-case analysis), where incomplete records or observations are simply discarded. The main virtue of case deletion is its simplicity. However, researchers need to be cautious against whether case deletion will bias statistical estimation and subsequently lead to incorrect inferences. Consider the estimation of the parameters of the regression of y on $x_1, x_2, \dots, x_p$ from data with missing values on y and/or xs, and assume that the regression function is correctly specified. If the probability of being a complete case depends on $x_1, x_2, \dots, x_p$ but not y, then estimates of the regression coefficients after case deletion will not be subject to bias. However, the same cannot be said with respect to other measures of association between y and the xs such as correlation coefficients (Schafer and Graham 2002). Moreover, case deletion results in biased estimation of regression coefficients if the probability of being a complete case depends on y conditional on the covariates $x_1, x_2, \dots, x_p$ (Little and Rubin 2014, Chapter 3.2).

In the past several decades, there have been active efforts in the statistics literature seeking to minimize the biasedness introduced by missing values. To better understand the mechanisms of missing values, Rubin (1976) formally identified three scenarios for the mechanisms of missing values – 1) missing completely at random (MCAR), where the missingness does not depend on any variable, 2) missing at random (MAR), where the missingness does not depend on the incomplete variable with missing values but can depend on variables without missing values, and 3) not missing at random (NMAR), where the missingness depends on the incomplete variable with missing values after conditioning on variables without missing values. Under the MCAR mechanism, case deletion estimates are generally valid (i.e., unbiased). However, missing values generally become problematic and result in non-ignorable bias when the underlying mechanism of missing values is MAR or NMAR, since the observed values and the unobserved ones follow different data distributions.

Under the MAR assumption, two broad approaches have been widely

acknowledged to be appropriate (e.g., Allison 2009; Newman 2014; Schafer and Graham 2002) – maximum likelihood estimation and multiple imputation. Maximum likelihood estimation is shown to be consistent and approximately efficient under MAR and valid joint distributional assumption of all the variables in the data matrix (Dempster et al. 1977; Rubin 1976). Theoretically, multiple imputation generates consistent estimations and have high asymptotic efficiency in large samples with proper Bayesian sampling approaches (Rubin 1987, p. 131).

Unfortunately, the theoretical properties of maximum likelihood estimation or multiple imputation cannot be ensured under the NMAR mechanism. Numerical results from simulations show that maximum likelihood and multiple imputation could lead to biased estimations under NMAR (Schafer and Graham 2002). Thus, the key remaining challenge in the missing data literature is how to handle missing data without restrictive assumptions on the mechanism of missing values. In recent years, we have seen newer developments for dealing with missing data that are NMAR. However, research results for NMAR are not quite as mature as those for MAR. If we cannot assume a MAR mechanism, additional (restrictive) assumptions of the mechanism of missing data, such as requiring the parameters of the missingness mechanism to be known (Kim and Yu 2011; Rotnitzky et al. 1998), often need to be imposed. However, to satisfy this requirement, unless the missingness mechanism is under control, analysts need to conduct additional follow up studies (Kim and Yu 2011), which is typically infeasible in data analytics practice.[2]

Motivated by this research gap, the research objective of the first study is *to develop approaches that can deal with missing values under the NMAR mechanism without pre-specifying the parameters of the missingness mechanism.* Since missing

[2] Recent studies propose data masking mechanisms for protecting users' information privacy (e.g., Banholzer et al. 2018; Li and Sarkar 2011). These mechanisms can be similarly used to control the missingness mechanism by which sensitive information is concealed. However, in many practical data analytic tasks, the missingness mechanism may not be under the control of analysts and the corresponding parameters are often unknown.

values are ubiquitous in both business analytics practice and empirical academic research, wherein the former often targets for predictive accuracy and the latter aims at obtaining unbiased statistical inference (Shmueli and Koppius 2011), I develop approaches along two sub-objectives.

The first sub-objective is to enhance the imputation accuracy of missing values for data analytics tasks. To achieve this goal, I propose a missing value imputation method built upon a semi-supervised learning strategy which employs both complete records and incomplete records for missing value imputation. In this study, I demonstrate the reduction of imputation error from my proposed method through theoretical and simulation analysis. Moreover, the imputation accuracy of the proposed method is indirectly shown in two real-world applications – the prediction of credit defaults and earnings. By imputing the important predictor variable for each of the two applications, the proposed method generates higher prediction accuracy compared to benchmark imputation approaches based on different machine learning methods. The second sub-objective is to correct the bias caused by missing values during the estimation of regression coefficients. I propose to employ a Monte Carlo Likelihood approach to obtain coefficient estimation in linear regression model with missing values. By incorporating the missingness mechanism into the coefficient estimation process, the Monte Carlo Likelihood approach can generate unbiased estimation under different missingness mechanisms including the NMAR mechanism.

The missing values problem, like the sample selection problem or others alike that may incur biasedness, is still undergoing extensive research. I expect that by providing theoretical analyses, results from simulations, and insights from real-world applications, data analysts will be able to better appreciate the potential bias caused by missing values, which will motivate more rigorous research designs to minimize the occurrence of missing values, or to obtain more information on the mechanisms of missing values when incomplete / missing data is inevitable.

The concept of missingness mechanism has a similar statistical origin with the sample selection model which will be used in the second essay. That is, data from different patterns can be discriminated by an underlying probabilistic mechanism. Incorporating this mechanism makes the data analysis results more robust to the heterogeneity in data patterns.

1.3.2 Adapting Statistical Learning in Dynamic Data Environments

As machine learning algorithms become more sophisticated, they have attracted much attention from information systems researchers and decision-making practitioners (Chen et al. 2012; Chiang et al. 2018). In machine learning, data is used to find interdependences in the world, with the goal of predicting future outcomes. In particular, in the supervised machine learning setting, we use historical data to learn the relationship between the predictor variables (often denoted with $\boldsymbol{x}$) and the variable to be predicted (often denoted with y). This relationship is typically denoted with a function $f(\boldsymbol{x};\boldsymbol{\theta})$ that maps from $\boldsymbol{x}$ to y, where $\boldsymbol{\theta}$ are parameters to be estimated from historical data. Then function $f(\boldsymbol{x};\boldsymbol{\theta})$ is used to predict future y based on currently observed values of $\boldsymbol{x}$.

Although machine learning is able to learn more complex relationship than traditional data analytics, as highlighted by Schölkopf (2017), the relationship uncovered by machine learning may not be robust to changes in real world datasets (Zhang et al. 2013). A simple solution to adapt to changes in the data pattern is to ignore all the historical source data and to retrain the machine learning model from scratch using the current data in the target data regime. However, current data is scarce at the early stage of the new data regime. The data sparsity becomes a critical concern in obtaining a reliable model for the new data regime especially since many machine learning models rely heavily on large amounts of data.

To take on this challenge, a new subfield of machine learning called transfer

learning has emerged.[3] The vision of transfer learning is to extract knowledge from a *source data* set and to apply that knowledge to a *target data* set. Given that only a small amount of the source data will exhibit the same data pattern as the target data (i.e., *same-distribution source data*), transfer learning aims at balancing the usage of the small amount of *same-distribution source data* and the often-large but less relevant source data that follows a data pattern different from the target data (i.e., *diff-distribution source data*).

Different methods have been proposed to balance the usage of source and target data. However, these studies often focus on the situation where the distribution of predictors changes across the source and target data while the relationship between predictors and the variable to be predicted is assumed to be fixed, a situation known as transductive transfer learning or as the covariate drift problem (Huang, Gretton, Borgwardt, Schölkopf, & Smola, 2007; Zadrozny, 2004). For the more general inductive transfer learning problem (Pan and Yang 2010), where both the distribution of predictors and the functional relationship change, the extant literature is relatively immature. Dai et al. (2007) develop a TrAdaBoost classifier for inductive transfer learning to exploit the same-distribution source data, and Pardoe and Stone (2010) extend this approach to regression problem. TrAdaBoost adjusts the iterative process of the popular AdaBoost algorithm, by increasing the weights of same-distribution source data and decreasing the weights of the remaining diff-distribution source data, which is consistent with our intuition in using all of the source data. However, as pointed out by the authors, TrAdaBoost does not guarantee to always improve upon AdaBoost, since the quality of diff-distribution source data is not certain.

Motivated by this research gap in the extant literature on transfer learning, *the first objective of the second essay is to design a theoretically driven transfer learning*

[3] According to ACM Computing Classification System (CCS 2012), transfer learning is a subfield of multi-task learning within machine learning paradigms.

method that aims at improving prediction performance on the target data. In changing data environments, an open question is how to represent changes. If we had information on how the data pattern changes (e.g., ideally complete information on how the distribution of variables changes), we can then adjust the model to the desired direction to the future data pattern. I propose a sample selection perspective to approximate changes in terms of a general inductive transfer learning problem. An underlying sample selection model is used to represent the probability that a data point represents a data pattern different from the target data given its values of predictors and the value to be predicted. Based on the probability, I derive a weighting approach to adjust the prediction model trained on the source data to let it fit the target data. The proposed method is theoretically driven by empirical risk minimization (ERM) for the target data distribution.

Sample selection models are widely used in improving causality identification (Heckman, 1979). Although data analytics for prediction do not aim at identifying causality (Shmueli and Koppius 2011), the sample selection perspective has been preliminarily adopted in transductive transfer learning studies (e.g., Zadrozny 2004) and helps to derive more robust model when there is potential heterogeneity in different data sets. In particular, the sample selection model provides the theoretical guidelines in driving the weighting method that minimizes empirical risk in the target data.

Another gap in the literature is that, although different transfer learning algorithms have empirically demonstrated successful implementation in changing data environments (Ganin et al., 2016; Pan, Zheng, Yang, & Hu, 2008), there is a lack of theoretical understanding on when and to what extent transfer learning works. By building on the sample selection perspective, another objective of the second essay is to *gain theoretical insights into when and to what extent transfer learning works in changing data environments.*

In practical applications, when changes are detected, analysts face the trade-off between two alternative strategies – 1) to re-train a model using a small but more relevant same-distribution source data set, or 2) to use transfer learning and leverage a larger sample of source data that may include some relevant same-distribution data and some diff-distribution source data. This is analogous to the fundamental bias-variance trade-off. Moreover, this challenge arises due to the scarcity of useful information on data heterogeneity, i.e., the same-distribution source data, as discussed in Section 1.2. Therefore, analysts can naturally consider waiting for a time period and to incorporate more same-distribution source data to train a more accurate model for the target task. However, this benefit comes at the cost of deteriorating prediction performance before the model adjustment is made. Therefore, another trade-off faced by analysts is with the time dimension – whether to make an adjustment immediately or at a later time point until more same-distribution source data becomes available. These two tradeoffs are examined through a simulation study conducted in a change detection context. Theoretical insights obtained in the second essay help to promote proper use of transfer learning and the understanding on challenges of statistical learning in real-world changing data environments.

Overall, the two studies in my dissertation investigate fundamental problems in data heterogeneity arisen from missing values and dynamic data environment. In the first essay on the missing values problem, the missing value mechanism is the device that models the latent mechanism that influences the data pattern we obtain based on the observed information. In the second essay on changing data environments, I adopt the sample selection perspective to reflect changes across source and target data. Moreover, in dynamic data environments, I examine the trade-offs in augmenting usable information within the context of data pattern change detection.

1.4 Outline of the Dissertation

This chapter (Chapter 1) has introduced the research background of the dissertation, highlights ongoing data analytics challenges with respect to the missing values problem and to dynamism in data environments, and identifies the research gaps.

Chapter 2 presents the first study of the dissertation. It provides a comprehensive review of the existing missing values literature and proposes approaches to handle missing values while accounting for the missingness mechanism. The proposed semi-supervised missing value imputation method demonstrates enhance imputation accuracy through theoretical analysis, simulation and real-world experimentation. The proposed computational approach using a Monte Carlo likelihood estimation method is able to correct biases caused by missing values.

Chapter 3 contains the second study of the dissertation. It reviews the literature on change detection and transfer learning that are related to dynamic data environments. I propose a sample selection framwork for the general inductive transfer learning problem. In the systematic simulation work, we show the effectiveness of transfer learning under different settings of dynamic data environments.

Finally, chapter 4 concludes the dissertation with a discussion on the contribution of the studies and directions for future research.

CHAPTER 2 HANDLING MISSING VALUES WITHOUT ASSUMING MISSING AT RANDOM

2.1 Introduction

Advances in data mining and machine learning technologies create new opportunities for information systems researchers and practitioners. However, in spite of the widely deployed information systems that are able to create large volumes of data for analytics (i.e., "big data"), a fundamental data challenge that persists is the missing values problem. For instance, in healthcare practice, paper-based medical records may not be digitized completely and successfully, which results in missing values in electronic health records (Baird et al. 2017). In finance, accounting or strategy studies, data on research and development (R&D) expenses is missing in 42% of the financial reports of NYSE-listed firms (Koh and Reeb 2015). As another example, in safety management research, over ninety percent of variables in aviation safety reporting systems contain missing values to varying extents (Shi et al. 2017). Compared to the prevalence of missing values, methodological studies regarding how to handle missing values are still left with many limitations.

In a seminal study, Rubin (1976) developed a typology of missingness mechanisms, which captures the relationship between the missingness indicator and the variables in the data matrix.[4] Three types of missingness mechanisms are defined as follows. The first type is missing completely at random (MCAR), where the missingness indicator does not depend on any variable. The second type is missing at random (MAR), where the missingness indicator does not depend on the incomplete variable with missing values but can depend on variables without missing values. The

[4] The missingness is defined as an indicator variable. It equals one if the incomplete variable is observed and zero otherwise. The data matrix is a rectangular data set. Traditionally, the rows of the data matrix represent records, also called observations or instances, and the columns represent variables, also called features, that are measured for each record. To be consistent in terminology, we refer to the rows as records and the columns as variables.

third type is the most general case, not missing at random (NMAR), where the missingness indicator depends on the incomplete variable with missing values after conditioning on variables without missing values. Under the MCAR mechanism, case deletion estimates are generally valid. However, missing values generally become problematic when the underlying mechanism of missing values is MAR or NMAR. Most of the existing literature, as will be discussed in the literature review section below, assumes MAR as the missingness mechanism.

Unfortunately, in many real-world data sets, the MAR assumption may not hold. For instance, in online recommendation systems, early buyers self-select products that they believe they would enjoy, thus the reviews we observed in the buyer-product matrix can be biased (Li and Hitt 2008). In healthcare research, albumin values are NMAR, since the albumin test is more accessible to sicker patients (Hall et al. 2007). Failing to accommodate the NMAR mechanism results in biased estimates that lead to suboptimal or even erroneous predictions or inferences (Little and Rubin 2014, p. 18). For instance, in the simple estimation of the mean of a variable, such as information technology investment, if the variable is more likely to be observed/reported when it has a larger value, then the estimated mean using the observed values would be upwardly biased.

Recently, statisticians have started to seriously consider NMAR. However, to handle the NMAR mechanism, additional (restrictive) assumptions are often imposed. For instance, Kim and Yu (2011) require the parameters of the missingness mechanism to be known. However, in many practical data analytic tasks, the missingness mechanism may not be under the control of analysts and the corresponding parameters are often unknown. Motivated by this research gap, we *develop approaches to handle missing values that are robust to the NMAR mechanism without pre-specifying parameters of the missingness mechanism.*

Handling missing values for the purpose of business analytics practice and

empirical academic research tends to have different focus, since the former often aims at obtaining high predictive accuracy using machine learning algorithms, and the latter often aims at obtaining unbiased coefficients in regression analysis (Shmueli and Koppius 2011). Therefore, in this chapter, we develop approaches along two sub-objectives.

The first sub-objective is to develop a missing value imputation approach focusing on enhancing imputation accuracy for data analytics tasks. To enhance imputation accuracy, we propose a missing value imputation method based on a semi-supervised learning strategy. With traditional imputation methods using supervised learning, the relationship, such as a simple linear relationship, between the incomplete variable and other complete variables are modeled using the complete records. To capture the potentially complex relationship and to achieve accurate imputation, machine learning algorithms can be applied. Different machine learning algorithms have been employed in the data analytics literature, such as decision tree, Bayesian method, linear regression, neural networks, SVM, and random forest (Ding and Simonoff 2010; Farhangfar et al. 2008; García-Laencina et al. 2010; Li 2009; Luengo et al. 2012; Saar-Tsechansky and Provost 2007; Stekhoven and Bühlmann 2012). This relationship is then applied to the incomplete records to impute missing values of the incomplete variable.

Compared to supervised imputation where only complete data records are used to infer the relationship between the incomplete variable and complete variables, semi-supervised imputation may make use of the incomplete data records as well. In this way, we make use of all available information and more importantly, this allows us to explicitly model the NMAR missingness mechanism. After all, with only the complete data records, the missingness indicator will always be equal to one. By incorporating the incomplete data records, it is possible to model the variation of the missingness indicator which is explained by the missingness mechanism. Therefore,

an important contribution of our method is to incorporate the NMAR missingness mechanism into the traditional imputation process through a semi-supervised learning wrapper. From a probabilistic view, we formally define the missing values problem and analytically demonstrate that the proposed method can enhance imputation accuracy over traditional methods which impute missing values by supervised learning methods and assume MAR.

We demonstrate the performance of our method using both simulations and real-world applications. With simulations, we show that our approach achieves higher imputation accuracy under varying missingness mechanisms and percentages of missing values of the incomplete variable. In the real-world applications, we employ two data sets. The first data set is drawn from a Kaggle competition of home credit default prediction.[5] The task of this competition is to predict whether each applicant will repay his/her loan given the applicant's records. In this data set, an important predictor variable, the credit score from a third-party credit rating agency, is missing for 56% of the applicants because only a previously approved loan applicant with good credit could be assessed using credit scores (Abdou and Pointon 2011; Verstraeten and Van den Poel 2005). In the second application, the task is to predict the quarterly earnings of US-listed firms based on their financial statements and analyst forecasts. In this data set, the analyst consensus forecast is missing for 40.7% of the observations because analyst consensus forecast is often not available for small firms and financially distressed firms (Diether et al. 2002). Given the possible NMAR mechanism in the two data sets, we impute the target incomplete predictor variables with our proposed method. Results show that our semi-supervised imputation method provides more accurate predictions compared to the traditional imputation methods, including support vector machine (SVM), random forest, and gradient boosting.

The second sub-objective is to employ a Monte Carlo likelihood estimation

[5] See https://www.kaggle.com/c/home-credit-default-risk.

approach to alleviate the bias in estimating coefficients caused by missing values for empirical regression analysis. Researchers have historically handled missing values primarily by dropping the observations whose information is incomplete (called *listwise deletion* or *complete case analysis*) or by editing the data (e.g., substituting missing values with the mean of the variable in question or even with zeros) to lend an appearance of completeness. However, such handling of missing values may lead to inference problems where incorrect conclusions are drawn from the analysis. Extant literature has well demonstrated that, under the MAR mechanism, two broad approaches, Maximum Likelihood (ML, Rubin 1976; Dempster et al. 1977) and Multiple Imputation (MI, Rubin 1987) provide consistent parameter estimation. However, these two approaches would lead to bias when the missingness mechanism is NMAR.

To accommodate the NMAR mechanism, we propose a computational approach using a Monte Carlo likelihood estimation method. Our method generates unbiased estimation for regression coefficients under different missingness mechanisms. Our proposed method is built upon recent theoretical advances showing that for certain types of model specifications, parameters can be identified using maximum likelihood estimation incorporating the NMAR mechanism (Miao et al. 2016). Monte Carlo likelihood estimation has been employed in a variety of maximum likelihood estimation problems where the likelihood function is difficult to be calculated directly due to missing values under the MAR mechanism (Sung and Geyer 2007) or when latent variables are involved (Booth and Hobert 1999). We extend the application to the NMAR situation and provide evidence of its superior performance in parameter estimation with simulation study.

In summary, challenges in deploying advanced data analytics to outperform competitors exist in every aspect of analytics such as data collecting and processing (Chiang et al. 2018; Clark and Provost 2016). Moreover, biasedness (or

inconsistency) caused by missing values, raises great concerns in both data analytics practice and empirical research since it cannot be resolved by simply increasing the sample size. Our missing value handling approaches offer important theoretical and practical contributions. The main contribution is that both proposed approaches are among the very few studies that handle the missing values problem without assuming a MAR mechanism. The semi-supervised imputation approach focuses on enhancing the imputation accuracy of supervised imputation using different machine learning algorithms, which will contribute to data analytics practice in missing value handling. The Monte Carlo likelihood estimation is based on established statistical theory and focuses on correcting biases in estimating regression coefficients caused by missing values. The improvements, in terms of both imputation accuracy in predictive data analytics and bias correction in empirical regression analysis, are achieved by explicitly modeling and estimating the missingness mechanism, which makes the approach more robust to the NMAR mechanism. By providing theoretical analyses, results from simulations, and insights from real-world applications, our research will motivate more rigorous data collecting and analyzing process to minimize the occurrence of missing values, or to obtain more information on the mechanisms of missing values when incomplete / missing data is inevitable.

2.2 Relevant Literature

A large number of scholarly works in the statistics and machine learning literatures have examined the missing values issue. In this section, we first review a typology of missingness mechanisms proposed in a seminal study by Rubin (1976): MCAR, MAR, and NMAR. Then we introduce the well-established statistical models for handling missing values. Finally, we present a brief overview of semi-supervised learning as it is related to the proposed approaches especially the semi-supervised learning approach to imputing missing values.

2.2.1 Typology of Missingness Mechanism

Let the variables in the data set be $(z, x_1, x_2, \dots, x_k)$ where $x_1, x_2, x_3, \dots, x_k$ are k complete variables without missing values z is the incomplete variable with missing values. Let s be the missingness indicator for z, where $s = 1$ if z has a value and $s = 0$ otherwise. To be concise, we let vector $\boldsymbol{x}$ denote all the complete variables, namely, $\boldsymbol{x} = (x_1, x_2, x_3, \dots, x_k)$.

The missingness mechanism model is often described with a conditional probability function of the missingness indicator s given values of z and $\boldsymbol{x}$, which is denoted by:

$$p(s|z, \boldsymbol{x}; \boldsymbol{\psi}), \tag{2.1}$$

where $\boldsymbol{\psi}$ is the unknown parameter in the probability function. A concrete example of the missingness mechanism is the logistic model, where $p(s = 1|z, \boldsymbol{x}; \boldsymbol{\psi}) = \frac{1}{1+e^{-(\psi_0+\psi_1 z+\boldsymbol{\psi}_2 \boldsymbol{x})}}$.

All records in the data set are assumed to be independently and identically distributed (i.i.d.). The three types of missingness mechanisms are defined as follows:

(1) MCAR is defined as $p(s|z, \boldsymbol{x}; \boldsymbol{\psi}) = p(s; \boldsymbol{\psi})$. Under this mechanism, s does not depend on any variable of the data set.

(2) MAR is defined as $p(s|z, \boldsymbol{x}; \boldsymbol{\psi}) = p(s|\boldsymbol{x}; \boldsymbol{\psi})$. Under this mechanism, s depends on $\boldsymbol{x}$ but does not depend on z.

(3) The most general mechanism is NMAR. Under this mechanism, s depends on both $\boldsymbol{x}$ and z. In other words, $p(s|z, \boldsymbol{x}; \boldsymbol{\psi}) = p(s|\boldsymbol{x}; \boldsymbol{\psi})$ does not hold.

Among the various methods for handling missing values, the most frequently used is listwise deletion, where incomplete records or observations are simply discarded. The main virtue of listwise deletion is its simplicity. Under the MCAR

mechanism, listwise deletion estimates are generally valid. However, missing values generally become problematic when the underlying mechanism of missing values is MAR or NMAR. Consider the estimation of the coefficients of the regression of y on z and $\boldsymbol{x}$, and assume that the regression function is correctly specified. If the probability of being a complete case depends on y (in this case, the missingness mechanism is MAR and may even be NMAR if the missingness also depends on z), then listwise deletion will result in biased estimates of the regression coefficients (Little and Rubin 2014, Chapter 3.2). Moreover, listwise deletion leads to a loss of a large amount of information contained in the incomplete records. Suppose that z is missing in 50% of the data records, listwise deletion leads to a loss of almost half of the information, which largely reduces the efficiency of information usage. In the next subsection, we review the statistical models to deal with missing values under various mechanisms of the missingness typology. We especially focus on the methods under the MCAR and the MAR mechanisms given the relatively mature statistical theories.

2.2.2 Statistical Models for Handling Missing Values

Under the MCAR and MAR mechanisms, two broad approaches, maximum likelihood and multiple imputation, have been acknowledged by statisticians to derive less biased estimations than traditional missing value handling approaches, such as list-wise deletion and mean imputation (Allison 2009; Newman 2014; Schafer and Graham 2002).

The maximum likelihood approach is shown to be approximately efficient under MAR and valid joint distributional assumption of all the variables in the data matrix (Dempster et al. 1977; Rubin 1976). However, to be useful in practice, the maximum likelihood method requires careful algebraic manipulations and efficient programming. Much research efforts have been devoted to the task of maximizing the

objective likelihood function. Little and Rubin (2014, Chapter 8), provide a review of methods for maximizing the likelihood function.

To implement maximum likelihood for missing data, one needs to assume a joint distribution for all variables (including z and $\boldsymbol{x}$) and to select an optimization method for maximizing the observed data likelihood and obtaining the parameters' estimations. Maximum likelihood estimates can be directly found by differentiating the likelihood with respect to the parameters of interest and then obtaining estimates by letting the resulted first order condition equal to zero. However, it is often difficult to obtain such estimates directly. Therefore, an iterative method that is computationally simple and feasible, such as the expectation maximization (EM) method (Dempster et al. 1977),[6] is often used to obtain the maximum likelihood estimate. The EM algorithm formalizes a relatively old ad hoc idea for handling missing data: (1) replacing missing values by estimated values, (2) estimating parameters, (3) re-estimating the missing values assuming the new parameter estimates are correct, (4) re-estimating parameters, and so forth, iterating until convergence. A detailed description of the EM algorithm for maximum likelihood estimation with missing values is presented in Appendix 1.1.1.

(Bayesian) Multiple Imputation (MI), proposed by Rubin (1987), is a flexible alternative to maximum likelihood methods. Multiple imputation theoretically generates consistent estimations in large samples and with a proper Bayesian sampling approach (Rubin 1987, p. 131). Moreover, the imputed data sets created by the multiple imputation approach can be easily used for subsequent data analysis (Melville and McQuaid 2012). However, current multiple imputation methods often

[6] It is acknowledged that similar idea had been proposed in earlier study (e.g., Hartley 1958). As illustrated by Dempster et al. (1977), application of the EM method is not limited to the missing data issue. Problems that can be conquered by EM are remarkably broad, such as grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

need to assume MAR, a normal distribution for the incomplete variable, and a prior distribution for the parameters of interest (Schafer 1997).

To implement this method, ideally, the analyst should draw multiple imputations according to the following protocol (Little and Rubin 2014, p. 86). The multiple imputations (e.g., m imputations) of the missing component of the variable z, say z_{mis}, are m repetitions from the posterior predictive distribution of z_{mis}; each repetition corresponds to an independent random draw of the parameters and missing values. After the m sets of imputations are generated, the analyst obtains the estimates of parameters in each data set using standard complete-data estimation methods and then combines the estimates obtained from the multiple imputed data sets using a simple method. Details for implementing multiple imputation are presented in Appendix 1.1.2.

Although maximum likelihood and multiple imputation are established methods with theoretical results, the MAR assumption required by these two methods may not hold in real-world data sets. For example, in online marketing analytics, customers may provide review ratings only for products that they either like or dislike very much (Ying et al. 2006); in healthcare analytics, albumin values are less likely to be missing for sicker patients (Hall et al. 2007). Failing to accommodate NMAR would result in biased estimates (Little and Rubin 2014, p. 18). Recently, statistical models have emerged to consider NMAR. However, these studies either assume a joint distribution of variables (Ibrahim et al. 1999) or impose assumptions such as requiring the parameters of the missingness mechanism to be known (Kim and Yu 2011).[7] In this study, we incorporate the missingness mechanism into the likelihood function. Moreover, compared with the existing maximum likelihood methods for

[7] Recent studies propose data masking mechanisms for protecting users' information privacy. (e.g., Banholzer et al. 2018; Li and Sarkar 2011). These mechanisms can be similarly used to control the missingness mechanism by which sensitive information is concealed. However, in many practical data analytic tasks, the missingness mechanism may not be under the control of researchers and the corresponding parameters are often unknown.

handling missing values, we do not impose the assumption of joint distribution for all variables.

2.2.3 Semi-Supervised Learning

To accomplish the goal of enhancing missing value imputation accuracy, we employ a semi-supervised learning approach to incorporate the NMAR missingness mechanism into the traditional imputation process. Unlike supervised learning which only uses the training set, the key idea of semi-supervised learning is to use both the training set and the test set to build a functional model (Hosmer Jr 1973; McLachlan 1977).

Before we introduce semi-supervised learning, we first describe how supervised learning can be used to handle missing values. Let us consider the scenario of imputing the incomplete variable z using other completely observed variables $\boldsymbol{x}$. In the following discussion, we divide the data set into two portions: (1) the "complete records" where the focal incomplete variable z is not missing (analogous to the training set), and (2) the "incomplete records" where the focal incomplete variable z is missing (analogous to the test set). In the traditional imputation approach based on supervised learning, analysts can use the complete records to build an imputation model that captures the relationship mapping from $\boldsymbol{x}$ to z; and then this imputation model is used to estimate (i.e., impute) missing values in the incomplete records.

In this section, we adopt a general probabilistic view to describe the semi-supervised learning strategy. In the above scenario with missing values, $\boldsymbol{x}$ and z are the variables of interest, thus both supervised and semi-supervised learning strategies aim at estimating a joint distribution of complete variables $\boldsymbol{x}$ and the incomplete variable z, indicated as follows:

$$f(\boldsymbol{x}, z; \boldsymbol{\omega}), \tag{2.2}$$

where $\boldsymbol{\omega}$ is the vector of unknown parameters in the probability density function.

The supervised learning strategy estimates parameter $\boldsymbol{\omega}$ using the training set. To be specific, $\boldsymbol{\omega}$ can be solved by maximizing the log-likelihood in the following equation:

$$\boldsymbol{\omega} = \arg\max_{\boldsymbol{\omega}} \sum_{i=1}^{m} \ln f(\boldsymbol{x}_i, z_i; \boldsymbol{\omega}), \tag{2.3}$$

where i indexes each record and m is the number of training records. After estimating parameter $\boldsymbol{\omega}$ of the joint distribution of $(\boldsymbol{x}, z)$, the unknown value of z in the test set can be inferred from the predictor $\boldsymbol{x}$ by maximizing the conditional distribution of z (Zhu and Goldberg 2009, Sect. 3.1), namely:

$$f(z|\boldsymbol{x}; \boldsymbol{\omega}) = \frac{f(\boldsymbol{x}, z; \boldsymbol{\omega})}{f_{\boldsymbol{x}}(\boldsymbol{x}; \boldsymbol{\omega})}, \tag{2.4}$$

where $f_{\boldsymbol{x}}(\boldsymbol{x}; \boldsymbol{\omega})$ denotes the marginal density function of $\boldsymbol{x}$, which is the integration of the joint distribution $f(\boldsymbol{x}, z; \boldsymbol{\omega})$ with respect to z.

The semi-supervised learning strategy uses both the training and test sets. $\boldsymbol{\omega}$ is then solved by maximizing an alternative log-likelihood function as:

$$\boldsymbol{\omega} = \arg\max_{\boldsymbol{\omega}} \sum_{i=1}^{m} \ln f(\boldsymbol{x}_i, z_i; \boldsymbol{\omega}) + \sum_{i=m+1}^{m+n} \ln f_{\boldsymbol{x}}(\boldsymbol{x}_i; \boldsymbol{\omega}), \tag{2.5}$$

where n is the number of records in the test set.

By leveraging the test set, the semi-supervised learning incorporates more information and generates more accurate estimation of $\boldsymbol{\omega}$ under general assumptions.[8] In the next section, to enhance imputation accuracy and accommodate the NMAR missingness mechanism, we propose a semi-supervised missing value imputation approach which makes use of "complete records" and "incomplete records" simultaneously.

[8] The general assumptions are the semi-supervised smoothness assumption, the cluster assumption, the low-density assumption, and the manifold assumption (Balcan and Blum 2010).

2.3 Proposed Approaches to Handling Missing Values without Assuming the MAR Mechanism

This section formally introduces two approaches to handling missing values: a semi-supervised approach for missing value imputation, and the Monte Carlo likelihood estimation for estimating regression coefficients with missing values.

In presenting the semi-supervised imputation approach, we first introduce a conceptual framework of our semi-supervised imputation approach. Next, we demonstrate that the semi-supervised imputation framework is theoretically superior to the traditional supervised imputation framework that ignores the missingness mechanism. Finally, we propose a nonparametric method for implementing our imputation approach. This nonparametric extension complements the semi-supervised imputation framework so that the proposed semi-supervised imputation approach can be used to enhance the imputation accuracy of traditional supervised imputation based on machine learning algorithms.

The semi-supervised imputation framework is theoretically superior to the supervised imputation framework. However, the flexibility of nonparametric implementation with machine learning algorithms makes it challenging to derive solid theoretical property, such as the convergence property in parameter estimation. Although there is emerging literature exploring the maximum likelihood estimation involving the machine learning algorithm such as random forest (Athey et al. 2019), the related study is under exploration which would be a promising future research direction.

To handle the missing values encountered during regression analysis, a theoretically grounded approach is in need to correct the bias in regression coefficients caused by missing value. This motivates us to leverage the Monte Carlo maximum likelihood estimation which is supported by the statistical literature on the identifiability of maximum likelihood estimation under the NMAR mechanism (Miao

et al. 2016). The Monte Carlo computation approach has been employed in a variety of maximum likelihood estimation problems where the likelihood function is difficult to be calculated directly due to missing values under the MAR mechanism (Sung and Geyer 2007) or when latent variables are involved (Booth and Hobert 1999). We develop and demonstrate the application of Monte Carlo maximum likelihood estimation to handle missing values under the NMAR mechanism.

2.3.1 A Semi-Supervised Learning Approach to Missing Value Imputation

2.3.1.1 Missing Value Imputation under the Semi-Supervised Learning Framework

In the proposed method, we handle the situation where one variable contains missing values. Let z be the incomplete variable with missing values, and vector $\boldsymbol{x}$ denotes all the complete variables. Without loss of generality, we assume that the incomplete variable z is a categorical random variable. Our theoretical inferences in Section 2.3.1.2 can be generalized to the case of z being a continuous variable. The imputation process for continuous incomplete variable z is presented in Section 2.3.1.3. The relationship between the incomplete variable z given complete variables $\boldsymbol{x}$ is represented as a conditional probability function of the incomplete variable z given complete variables $\boldsymbol{x}$, namely:

$$q(z|\boldsymbol{x};\boldsymbol{\theta}), \tag{2.6}$$

where $\boldsymbol{\theta}$ denotes unknown parameters of the conditional probability density function.

Let s be the missingness indicator for z, where $s = 1$ if z has a value and $s = 0$ otherwise. The missingness mechanism is denoted as $p(s|z,\boldsymbol{x};\boldsymbol{\psi})$. The proposed method enhances the imputation accuracy by incorporating the missingness mechanism into the imputation process. An overview of the proposed solution is provided below:

Step 1: We first predict and impute the missing values of z using complete records. This step is like the traditional imputation process. Almost any classification algorithm can be used in this imputation step as long as the classification algorithm outputs the predicted probabilities of each possible value of z.

Step 2: Given the predicted probabilities of each possible value of z, we use a unique likelihood function inspired by semi-supervised learning to estimate the model of the missingness mechanism described in Expression (2.1).

Step 3: Given the results of Steps 1 and 2, we compute the final imputed value of z.

Step 1: Traditional Imputation

The typical goal of a traditional imputation method is to learn the conditional probability function of the incomplete variable z given complete variables $\boldsymbol{x}$ in Expression (2.6). From a probabilistic view, we estimate $\boldsymbol{\theta}$ using the maximum likelihood method. Using a supervised learning framework, the maximum likelihood function for the imputation model in Equation (2.6) is given by:

$$L^s(\boldsymbol{\theta}):\ L(\boldsymbol{z}^C|\boldsymbol{X}^C;\boldsymbol{\theta}) = \prod_{i=1}^{m} q(z_i|\boldsymbol{x}_i;\boldsymbol{\theta}), \tag{2.7}$$

where subscript i indexes each record, and the superscript C indicates that the data comes from the complete records. $\boldsymbol{z}^C$ and $\boldsymbol{X}^C$ denote the values of the incomplete variable and complete variables of the complete records, respectively. m is the number of complete records. Let $\boldsymbol{\theta}^*$ denote the optimal solution, which is derived by maximizing L^S. Given $\boldsymbol{\theta}^*$, the imputed value z_i of the incomplete record under the traditional imputation process can be solved as follows:

$$z_i = \arg\max_z q(z|\boldsymbol{x}_i;\boldsymbol{\theta}^*), i = m+1, \dots, m+n, \tag{2.8}$$

where n is the number of complete records.

One important limitation of the traditional imputation process under the supervised learning setting is that the missingness mechanism is completely ignored.

However, this step provides important intermediate results for us to derive the semi-supervised learning solution.

Step 2: Semi-Supervised Learning

Motivated by semi-supervised learning, we incorporate the incomplete records into the maximum likelihood function as follows:

$$L^{s-s}(\boldsymbol{\theta}, \boldsymbol{\psi}): \quad L(\boldsymbol{z}^C, \boldsymbol{s}^C, \boldsymbol{s}^{IC} | \boldsymbol{X}^C, \boldsymbol{X}^{IC}; \boldsymbol{\theta}, \boldsymbol{\psi}) = \prod_{i=1}^{m} \Pr(z_i, s_i | \boldsymbol{x}_i; \boldsymbol{\theta}, \boldsymbol{\psi}) \cdot \prod_{i=m+1}^{m+n} \Pr(s_i | \boldsymbol{x}_i; \boldsymbol{\psi}), \tag{2.9}$$

where the superscript IC indicates that the data comes from the incomplete records. $\boldsymbol{X}^{IC}$ denotes the values of the complete variables in the incomplete records. $\boldsymbol{s}^C$ and $\boldsymbol{s}^{IC}$ are the values of the missingness indicator in the complete records and incomplete records, respectively. Namely, $\boldsymbol{s}^C$ and $\boldsymbol{s}^{IC}$ are constant vectors of 1 and 0, respectively. n is the number of incomplete records. $\Pr(z_i, s_i | \boldsymbol{x}_i; \boldsymbol{\theta}, \boldsymbol{\psi})$ denotes the joint probability of z being z_i and s being s_i. $\Pr(s_i | \boldsymbol{x}_i; \boldsymbol{\psi})$ is the marginal probability of s being s_i.

Compared to the likelihood function $L^s(\boldsymbol{\theta})$ in traditional supervised learning settings, our objective likelihood function $L^{s-s}(\boldsymbol{\theta}, \boldsymbol{\psi})$ additionally includes the information from the incomplete records $\boldsymbol{X}^{IC}$ as well as the missingness indicators $\boldsymbol{s}^C$ and $\boldsymbol{s}^{IC}$. In this way, $L^{s-s}(\boldsymbol{\theta}, \boldsymbol{\psi})$ incorporates all available information, whereas $L^s(\boldsymbol{\theta})$ only considers the information of z and $\boldsymbol{x}$ for complete records. However, maximizing an objective function such as $L^{s-s}(\boldsymbol{\theta}, \boldsymbol{\psi})$ is challenging with unknown parameter $\boldsymbol{\psi}$. It is worth noting that, since z is unknown for incomplete records, we cannot directly estimate $\boldsymbol{\psi}$ of the missingness mechanism $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$. To overcome this challenge, we temporally fill up the missing values of z so that we can proceed with the estimation of the missingness mechanism $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ (e.g., by estimating a logit model regressing s on z and $\boldsymbol{x}$). However, directly filling the missing values of z

according to Equation (2.8), although intuitively correct, does not have theoretical support for increasing L^{s-s}. Therefore, we propose using the following sampling approach to fill missing values of z to estimate $\boldsymbol{\psi}$.

More specifically, for each incomplete record, let the missing values of z to be filled with all the possible values of z. We take the example of z being a binary variable (i.e., $z_i \epsilon \{0,1\}$). Each incomplete record i ($i = m+1, \ldots, m+n$), denoted with $(\boldsymbol{x}_i, ?)$, is expanded into two records: $(\boldsymbol{x}_i, 0)$ and $(\boldsymbol{x}_i, 1)$. The $(\boldsymbol{x}_i, 0)$ record is assigned with a weight $q(z = 0|\boldsymbol{x}_i; \boldsymbol{\theta}^*)$, while the $(\boldsymbol{x}_i, 1)$ record is assigned with a weight $q(z = 1|\boldsymbol{x}_i; \boldsymbol{\theta}^*)$. For the complete records, each of them is kept as is and is assigned a weight of 1. In this sense, there will be $m + n \times 2$ records accompanied by their weights. Then one can proceed to the estimation of parameter $\boldsymbol{\psi}$.[9]

Like Step 1, in practical implementation, we can use machine learning to model $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$. Here, we denote the estimate for parameter $\boldsymbol{\psi}$ as $\boldsymbol{\psi}^*$.

Step 3: Final Imputation

Given estimated parameters $(\boldsymbol{\theta}^*, \boldsymbol{\psi}^*)$, the imputed value z_i of the incomplete record under the semi-supervised learning setting can be solved as follows:

$$\begin{aligned} z_i &= \arg max_z \Pr(z, s_i = 0|\boldsymbol{x}_i; \boldsymbol{\theta}^*, \boldsymbol{\psi}^*) \\ &= \arg max_z \, p(s_i = 0|z, \boldsymbol{x}_i; \boldsymbol{\psi}^*) q(z|\boldsymbol{x}_i; \boldsymbol{\theta}^*), \; i=m+1, \ldots, m+n, \end{aligned} \tag{2.10}$$

where $\Pr(z, s_i = 0|\boldsymbol{x}_i; \boldsymbol{\theta}^*, \boldsymbol{\psi}^*)$ denotes the joint probability of $z_i = z$ and $s_i = 0$. The equation of the second line is derived by Bayes' rule. In this final imputation step, Equation (2.10) incorporates the missingness mechanism into the traditional imputation process while the traditional imputation approach from Equation (2.8) ignores the missingness mechanism.

[9] The sampling approach can be analogously extended to the situation when z has multiple possible values. As the number of possible values of z increases, however, we would expect more random samples to be drawn to accurately approximate the conditional distribution of z.

Pseudocode of the imputation steps above is detailed in Table 2-1.

Algorithm 2.1: Semi-supervised imputation for categorical variable

Data:

Z, a column vector of length $m+n$, denoting the incomplete variable
// z_i denotes the value of variable z for the ith record, $i = 1, 2, \dots, m+n$.
// Records are sorted such that the first m values of Z are observed and the last n values are missing.

X, a $(m+n) \times k$ matrix containing $m+n$ records and k complete variables
// $\boldsymbol{x}_i$ denotes values of k complete variables for the ith record, $i = 1, 2, \dots, m+n$.

S, a column vector of length $m+n$, indicating the missingness of Z
// s_i is the missingness indicator for z_i, where $s_i = 1$ if z_i has a value and $s_i = 0$ otherwise, $i = 1, 2, \dots, m+n$.

Input:

b, the number of all possible values of z specified by the user

ω, an empty list to store weights for records

ImputeMdl, user specified model of the relationship between z and $\boldsymbol{x}$
// For categorical variable z, *ImputeMdl* generates the probability distribution of z over $\{z_{(1)}, z_{(2)}, \dots, z_{(b)}\}$ given $\boldsymbol{x}$.

Output:

$\hat{Z}$, a vector of length $m+n$ of the imputed variable, initialized as Z
// The missing values in $\hat{Z}$ are replace with $\hat{z}_i$, $i = m+1, \dots, m+n$.

// **Step 1:** Traditional imputation

estimate imputation model *ImputeMdl*, $q(z|\boldsymbol{x}; \boldsymbol{\theta}^*)$, using m complete records;

// **Step 2:** Semi-supervised learning

for each i in $\{1, 2, \dots, m\}$ // complete records

  assign the complete record $(z_i, \boldsymbol{x}_i)$ with weight one, and store the weight to the list ω;

end for

for each i in $\{m+1, m+2, \dots, m+n\}$ // incomplete records

  expand the incomplete record $(?, \boldsymbol{x}_i)$ to b records: $(z_{(1)}, \boldsymbol{x}_i)$, $(z_{(2)}, \boldsymbol{x}_i)$, … $(z_{(b)}, \boldsymbol{x}_i)$;

  assign weight $q(z_{(j)}|\boldsymbol{x}_i; \boldsymbol{\theta}^*)$ to each of the expanded record $(z_{(j)}, \boldsymbol{x}_i)$, $j = 1, 2, \dots, b$, and store the weight to the list ω;

end for

regress s on $(z, \boldsymbol{x})$ using expanded data ($m+n \times b$ records) with the corresponding weight ω, and get estimated model $p(s|z, \boldsymbol{x}; \boldsymbol{\psi}^*)$;

// **Step 3:** Final Imputation

for each i in $\{m+1, m+2, \dots, m+n\}$

  replace missing values in Z with $\hat{z}_i$ according to Equation (2.10), namely $\hat{z}_i \leftarrow \text{argmax}_z p(s_i = 0|z, \boldsymbol{x}_i; \boldsymbol{\psi}^*) q(z|\boldsymbol{x}_i; \boldsymbol{\theta}^*)$;

end for

return $\hat{Z}$

Table 2-1 Algorithm of Semi-Supervised Imputation for Categorical Variable

2.3.1.2 Comparing Supervised and Semi-Supervised Imputation Methods

In Theorem 2.1 presented below, we prove that our approach generates a greater likelihood value L^{s-s} compared to L^{s} from the traditional approach. The solution of the traditional approach is a special case of our solution (when $\boldsymbol{\psi} = \mathbf{0}$, i.e., $L^{s}(\boldsymbol{\theta}^{*}) = L^{s-s}(\boldsymbol{\theta}^{*}, \mathbf{0})$), and formally, we show that $L^{s-s}(\boldsymbol{\theta}^{*}, \boldsymbol{\psi}^{*}) > L^{s-s}(\boldsymbol{\theta}^{*}, \mathbf{0})$ in Theorem 2.1. Moreover, the parameters estimate $(\boldsymbol{\theta}^{*}, \mathbf{0})$ in the right-hand side term is conceptually employed by the traditional imputation of Equation (2.8), as shown in Theorem 2.2.

Lemma 2.1. Let $\mathbb{Z}$ be the set of all possible values of the categorical variable z. $|\mathbb{Z}|$ denotes the number of elements in set $\mathbb{Z}$. We define two functions, $g(\boldsymbol{\psi})$ and $\hat{g}(\boldsymbol{\psi})$, as follow:

$$g(\boldsymbol{\psi}) = \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}) \text{, and}$$

$$\hat{g}(\boldsymbol{\psi}) = \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \sum_{\tilde{z}\in\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^{*}) \ln p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}).$$

The inequality $g(\boldsymbol{\psi}^{*}) \geq g(\mathbf{0})$ holds if $\boldsymbol{\psi}^{*}$ is optimum for maximizing function $\hat{g}(\boldsymbol{\psi})$.

In the expression of $g(\boldsymbol{\psi})$, $\Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi})$ indicates the marginal probability of $s = s_i$ conditional on $\boldsymbol{x}_i$, which is obtained by summing the joint probability of z_i and s_i conditional on $\boldsymbol{x}_i$ with respect to z_i, namely $\sum_{\tilde{z}\in\mathbb{Z}} \Pr(s_i, \tilde{z}|\boldsymbol{x}_i; \boldsymbol{\psi}, \boldsymbol{\theta}^{*})$.

Proof sketch:

First, it can be proved that $g(\boldsymbol{\psi}) \geq \hat{g}(\boldsymbol{\psi})$:

$$\begin{aligned}
g(\boldsymbol{\psi}) &= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}) && \text{(i)} \\
&= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}, \boldsymbol{\theta}^{*}) && \text{(ii)} \\
&= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \sum_{\tilde{z}\in\mathbb{Z}} \Pr(s_i, \tilde{z}|\boldsymbol{x}_i; \boldsymbol{\psi}, \boldsymbol{\theta}^{*}) && \text{(iii)} \\
&= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \sum_{\tilde{z}\in\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^{*}) p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}) && \text{(iv)} \\
&\geq \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \sum_{\tilde{z}\in\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^{*}) \ln p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}) && \text{(v)} \\
&= \hat{g}(\boldsymbol{\psi}). && \text{(vi)}
\end{aligned}$$

Equality (i) holds since the relationship between parameter $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$ are not constrained (e.g., by imposing a prior assumption). Inequality (v) is supported by Jensen's inequality. When $\boldsymbol{\psi} = \mathbf{0}$, there is $p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi} = \mathbf{0})$ being constant with respect to z_i. Under this condition, $g(\boldsymbol{\psi}) \geq \hat{g}(\boldsymbol{\psi})$ holds with equality. Therefore, there is $g(\mathbf{0}) = \hat{g}(\mathbf{0})$. Since $\boldsymbol{\psi}^*$ is optimum for maximizing $\hat{g}(\boldsymbol{\psi})$, there is $\hat{g}(\boldsymbol{\psi}^*) \geq \hat{g}(\mathbf{0})$. Since $g(\boldsymbol{\psi}) \geq \hat{g}(\boldsymbol{\psi})$ holds for general cases, there is $g(\boldsymbol{\psi}^*) \geq \hat{g}(\boldsymbol{\psi}^*) \geq \hat{g}(\mathbf{0}) = g(\mathbf{0})$.

Q.E.D.

Theorem 2.1. The inequality $L^{s-s}(\boldsymbol{\theta}^*, \boldsymbol{\psi}^*) \geq L^{s-s}(\boldsymbol{\theta}^*, \mathbf{0})$ holds if $\boldsymbol{\theta}^*$ is optimum for maximizing L^s and $\boldsymbol{\psi}^*$ is optimum for maximizing function $\hat{g}(\boldsymbol{\psi})$.

In Theorem 2.1, $\hat{g}(\boldsymbol{\psi})$ is the objective function for maximization in Step 2. The first summation of $\hat{g}(\boldsymbol{\psi})$ indicates the sum of the log-likelihoods of observing the missingness indicator of the complete records. The second summation describes that each incomplete record is expanded by filling the missing z_i with all possible realizations in $\mathbb{Z}$ and assigning each expanded record with the weight $q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^*)$.

Proof sketch:

First, decompose the joint probability $\Pr(z_i, s_i|\boldsymbol{x}_i; \boldsymbol{\theta}, \boldsymbol{\psi})$ as follows:

$$\Pr(z_i, s_i|\boldsymbol{x}_i; \boldsymbol{\theta}, \boldsymbol{\psi}) = p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) q(z_i|\boldsymbol{x}_i; \boldsymbol{\theta}).$$

Substitute the above equation into L^{s-s}, there is:

$$\begin{aligned} L^{s-s}(\boldsymbol{\theta}, \boldsymbol{\psi}) &= \prod_{i=1}^{m} [p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) q(z_i|\boldsymbol{x}_i; \boldsymbol{\theta})] \cdot \prod_{i=m+1}^{m+n} \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}) \\ &= \{\prod_{i=1}^{m} p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) \cdot \prod_{i=m+1}^{m+n} \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi})\} \cdot \{\prod_{i=1}^{m} q(z_i|\boldsymbol{x}_i; \boldsymbol{\theta})\} \\ &= \{\prod_{i=1}^{m} p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) \cdot \prod_{i=m+1}^{m+n} \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi})\} \cdot L^s. \end{aligned}$$

Taking $\ln(\cdot)$ on both sides of the above equation, there is:

$$\begin{aligned} \ln\left(L^{s-s}(\boldsymbol{\theta}, \boldsymbol{\psi})\right) &= \ln\{\prod_{i=1}^{m} p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) \cdot \prod_{i=m+1}^{m+n} \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi})\} + \ln\left(L^s(\boldsymbol{\theta})\right) \\ &= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}) + \ln\left(L^s(\boldsymbol{\theta})\right) \end{aligned}$$

$$= g(\boldsymbol{\psi}) + \ln\left(L^{s}(\boldsymbol{\theta})\right).$$

According to **Lemma 2.1**, there is $g(\boldsymbol{\psi}^*) \geq g(\mathbf{0})$. Hence,

$$\begin{aligned}\ln\left(L^{s-s}(\boldsymbol{\theta}^*, \boldsymbol{\psi}^*)\right) &= g(\boldsymbol{\psi}^*) + \ln\left(L^{s}(\boldsymbol{\theta}^*)\right) \geq g(\mathbf{0}) + \ln\left(L^{s}(\boldsymbol{\theta}^*)\right) \\ &= \ln\left(L^{s-s}(\boldsymbol{\theta}^*, \mathbf{0})\right).\end{aligned}$$

Given the monotone property of the $\ln(\cdot)$ function, there is $L^{s-s}(\boldsymbol{\theta}^*, \boldsymbol{\psi}^*) \geq L^{s-s}(\boldsymbol{\theta}^*, \mathbf{0})$.

Q.E.D.

Lemma 2.1 can be generalized to the case of z being a continuous variable, with $\hat{g}(\boldsymbol{\psi}) = \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \int_{\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^*) \ln p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}) d\tilde{z}$, where $\mathbb{Z}$ is the value space of the continuous variable z. Detailed proof is shown in Appendix 1.2. Since Theorem 2.1 is valid under general forms of $\hat{g}(\boldsymbol{\psi})$, it is also generalizable to the case of z being continuous.

Theorem 2.2 The final imputation of Equation (2.10), using the parameter estimates $(\boldsymbol{\theta}^*, \mathbf{0})$, generates the same results as the traditional imputation of Equation (2.8).

Proof sketch:

First, it is worth noting that, when $\boldsymbol{\psi} = \mathbf{0}$, the missingness mechanism $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ is invariant with respect to z. This statement obviously holds for general functional forms of $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$, thus we provide a conceptual discussion here. For example, let $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ be a logistic model and we split the parameter vector $\boldsymbol{\psi}$ into two parts, $\boldsymbol{\psi}^z$ for the coefficient of z, and $\boldsymbol{\psi}^{\boldsymbol{x}}$ for the coefficients of $\boldsymbol{x}$. Then there is $p(s = 1|z, \boldsymbol{x}; \boldsymbol{\psi}) = \frac{1}{\left(1+e^{-(z\times\boldsymbol{\psi}^z + \boldsymbol{x}\times\boldsymbol{\psi}^{\boldsymbol{x}})}\right)}$. In this sense, $\boldsymbol{\psi} = \mathbf{0}$ keeps the $p(s = 1|z, \boldsymbol{x}; \boldsymbol{\psi})$ constant. Similarly, we can infer that under $\boldsymbol{\psi} = \mathbf{0}$, $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ becomes constant when $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ is modeled by a neural network or SVM, thus $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ is invariant to z. Then we can infer that the solution of z_i in Equation (2.10) under

$(\boldsymbol{\theta} = \boldsymbol{\theta}^*; \boldsymbol{\psi} = \mathbf{0})$ only depends on the term $q(z|\boldsymbol{x}_i, \boldsymbol{\theta}^*)$, which is just the solution of z_i under the traditional imputation of Equation (2.8).

Q.E.D.

By combining Theorems 2.1 and 2.2, we demonstrate that our semi-supervised parameter estimates are theoretically superior to the ones generated by the traditional imputation approach. It is also worth noting that, when obtaining an estimate of $\boldsymbol{\psi}^* = \mathbf{0}$, our imputation method degrades to the traditional imputation method. Therefore, the traditional approach's solution can be viewed as a special case of our solution.

In a practical implementation, various machine learning algorithms can be used to model $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$ and $q(z|\boldsymbol{x}; \boldsymbol{\theta})$, such as SVMs and neural networks. However, it is worth noting that Steps 1-3 are illustrated in the case of z being a categorical random variable. When z is a continuous random variable, the implementation of Step 1 to obtain $q(z|\boldsymbol{x}, \boldsymbol{\theta}^*)$ requires additional computational processes unless a conditional distribution of z given $\boldsymbol{x}$ is pre-assumed (e.g., normal distribution). In the next subsection, we propose a solution for the situation where z is a continuous variable without imposing distributional assumption of z.

2.3.1.3 Nonparametric Estimation of the Conditional Distribution

As discussed in Section 2.3.1.1, two critical elements of our imputation process are the missingness mechanism, $p(s|z, \boldsymbol{x}; \boldsymbol{\psi})$, and the conditional distribution of the incomplete variable, $q(z|\boldsymbol{x}; \boldsymbol{\theta})$. When z is a continuous variable, let the imputed value obtained by a machine learning algorithm (which can also be a simple regression algorithm for predicting continuous variables) in Step 1 be denoted by $\hbar(\boldsymbol{x}; \boldsymbol{\theta}^*)$. As a result, the residual term ε_i is given by:

$$\varepsilon_i = z_i - \hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*), i = 1,2, \dots, m.$$

We assume that the residual ε is i.i.d., then we discretize the continuous residual ε by constructing a histogram for ε.[10] The discretized ε is denoted with ε^d. The range of ε is taken as the interval from $min(\varepsilon_i)$ to $max(\varepsilon_i)$, where $min(\varepsilon_i)$ and $max(\varepsilon_i)$ are respectively the minimum and maximum values among the residuals ε_i, $i = 1,2, \dots, m$. We divide the range of ε to b bins with equal length and let each bin be represented by its midpoint. The value space of ε^d is denoted with the set $\{\varepsilon^d_{(1)}, \varepsilon^d_{(2)}, \dots, \varepsilon^d_{(b)}\}$. The probability of each bin, $\Pr(\varepsilon^d_{(j)}), j = 1,2, \dots, b$, is its corresponding frequency.

During Step 2, the value space of the random sampling for missing z, given $\boldsymbol{x}$, is $\{\varepsilon^d_{(0)} + \hbar(\boldsymbol{x}; \boldsymbol{\theta}^*), \varepsilon^d_{(1)} + \hbar(\boldsymbol{x}; \boldsymbol{\theta}^*), \dots, \varepsilon^d_{(b)} + \hbar(\boldsymbol{x}; \boldsymbol{\theta}^*)\}$. The probability of each value $\varepsilon^d_{(j)} + \hbar(\boldsymbol{x}; \boldsymbol{\theta}^*)$ is the corresponding probability of $\varepsilon^d_{(j)}, j = 1,2, \dots, b$. Like Step 2 for the case of categorical incomplete variable, we can obtain estimates for parameter $\boldsymbol{\psi}$ using complete records and randomly drawn observations for incomplete records.

Finally, in Step 3, missing values of z are imputed by adding the discretized residual to fitted value $\hat{z} = \hbar(\boldsymbol{x}; \boldsymbol{\theta}^*)$, wherein the discretized residual is solved by maximizing $\Pr(\hat{z} + \varepsilon^d, s = 0 | \boldsymbol{x}; \boldsymbol{\theta}^*, \boldsymbol{\psi}^*) = \Pr(s = 0 | \hat{z} + \varepsilon^d, \boldsymbol{x}; \boldsymbol{\theta}^*, \boldsymbol{\psi}^*) \Pr(\varepsilon^d)$, which is the joint probability of the imputed value and the missingness indicator.

The nonparametric approach provides a distribution-free solution to implement our method proposed in Section 2.3.1.1 to handle continuous incomplete variables. It is worth noting that our solution can work with machine learning algorithms to model the potentially complex realizations of $q(z|\boldsymbol{x}, \boldsymbol{\theta})$. Pseudocode of semi-supervised imputation for continuous variable is detailed in Table 2-2.

[10] A more advanced density estimation for ε can also be used to achieve improved approximation accuracy. For instance, the density of ε can be approximated with the nonparametric kernel density estimates $\frac{1}{mh} \sum_{i=1}^{m} K\left(\frac{\varepsilon - \varepsilon_i}{h}\right)$, where $K(\cdot)$ is a normal kernel and h is the bandwidth (Cameron et al. 2005, p. 299).

Algorithm 2.2: Semi-supervised imputation for continuous variable

Data:

Z, a column vector of length $m+n$, denoting the incomplete variable
// z_i denotes the value of variable z for the ith record, $i = 1, 2, \dots, m+n$.
// Records are sorted such that the first m values of Z are observed and the last n values are missing.

X, a $(m+n) \times k$ matrix containing $m+n$ records and k complete variables
// $\boldsymbol{x}_i$ denotes values of k complete variables for the ith record, $i = 1, 2, \dots, m+n$.

S, a column vector of length $m+n$, indicating the missingness of Z
// s_i is the missingness indicator for z_i, where $s_i = 1$ if z_i has a value and $s_i = 0$ otherwise, $i = 1, 2, \dots, m+n$.

Input:

b, the number of all possible values of z
// b is specified by user for discretizing the continuous variable z to b bins.

ω, an empty list to store weights for records

ImputeMdl, user specified model of the relationship between z and $\boldsymbol{x}$
// For continuous variable z, *ImputeMdl* generates fitted value of z given $\boldsymbol{x}$.

Output:

$\tilde{Z}$, a vector of length $m+n$ of the imputed variable, initialized as Z
// The missing values in $\tilde{Z}$ are replace with $\tilde{z}_i$, $i = m+1, \dots, m+n$.

// **Step 1:** Traditional imputation

estimate *ImputeMdl* to generate functional relationship between z and $\boldsymbol{x}$, denoted with $\hbar(\boldsymbol{x}; \boldsymbol{\theta}^*)$, using m complete observations;

// **Step 2:** Semi-supervised learning

for each i in $\{1, 2, \dots, m\}$ // complete records

  assign the complete record $(z_i, \boldsymbol{x}_i)$ with weight one, and store the weight to the list ω;

  $\hat{\varepsilon}_i \leftarrow z_i - \hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*)$; // calculate the residual term

end for

using residual terms of m records to discretize ε to the value space $\{\varepsilon^d_{(1)}, \varepsilon^d_{(2)}, \dots, \varepsilon^d_{(b)}\}$ and obtain the corresponding probability $q(\varepsilon^d_{(j)})$, $j = 1, 2, \dots, b$, through nonparametric estimation;

for each i in $\{m+1, m+2, \dots, m+n\}$ // incomplete records

  expand the incomplete record $(?, \boldsymbol{x}_i)$ to b records: $(\hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*) + \varepsilon^d_{(1)}, \boldsymbol{x}_i)$, $(\hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*) + \varepsilon^d_{(2)}, \boldsymbol{x}_i)$, … $(\hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*) + \varepsilon^d_{(b)}, \boldsymbol{x}_i)$;

  assign weight $q(\varepsilon^d_{(j)})$ to each of the expanded record $(\hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*) + \varepsilon^d_{(j)}, \boldsymbol{x}_i)$, $j = 1, 2, \dots, b$, and store the weight to the list ω;

end for

regress s on $(z, \boldsymbol{x})$ using expanded data ($m+n \times b$ records) with the corresponding weight ω, and get estimated model $p(s|z, \boldsymbol{x}; \boldsymbol{\psi}^*)$;

// **Step 3:** Final Imputation

for each i in $\{m+1, m+2, \dots, m+n\}$

  replace missing values in $\tilde{Z}$ with $\tilde{z}_i$, wherein $\tilde{z}_i \leftarrow \hat{z}_i + \operatorname{argmax}_{\varepsilon^d_{(j)}} p(s_i = 0 | \hat{z}_i + \varepsilon^d_{(j)}, \boldsymbol{x}_i; \boldsymbol{\psi}^*) q(\varepsilon^d_{(j)})$ and $\hat{z}_i = \hbar(\boldsymbol{x}_i; \boldsymbol{\theta}^*)$;

end for

return $\tilde{Z}$

Table 2-2 Algorithm of Semi-Supervised Imputation for Continuous Variable

2.3.2 Monte Carlo Likelihood Estimation to Correct Bias Caused by Missing Values

Although maximum likelihood (ML) and multiple imputation (MI) are theoretically sound for large samples, they still rest on a critical assumption that the missingness mechanism is MAR. In most cases we should expect departure from MAR (Schafer and Graham 2002), as in the self-selection case of missing healthcare records and product review ratings. In the following discussion, we present the Monte Carlo maximum likelihood estimation approach to handling missing values under NMAR.

Likelihood estimation for MNAR models is difficult from a computational standpoint. For incomplete observations, integration over the unobserved z value is required to compute the likelihood. Researchers have long been concerned that, under the NMAR mechanism, parameter estimation is often difficult or impossible (Rotnitzky et al. 1998; Wang et al. 2014). However, recent theoretical analyses show that it is possible to identify the parameters of interest, at least for certain types of model specifications (Miao et al. 2016). Specifically, parameters are identifiable when we can assume a normal model for the incomplete variable and a monotone missingness mechanism (e.g., the common logit or probit model). In the proposed method, we follow the above assumptions for the incomplete variable and the missingness mechanism.

To distinguish between our proposed method and the traditional maximum likelihood assuming MAR, we denote the latter with ML-MAR. As with the ML-MAR and MI methods, our estimation process does not distinguish between the dependent and independent variables during the parameter estimation process. Our estimation involves the parameters for the conditional distribution of the incomplete variable given complete variables. The regression coefficients are computed after this parameter estimation process.

Consider a data set with three variables of interest x, y, and z, where variable z is missing for certain observations. Without loss of generality, the regression model of interest has y being the dependent variable while x is the dependent variable and z is the control variable. We estimate parameters of the conditional distribution of z before proceeding with estimating coefficients for the regression model.

Particularly, we model the conditional distribution of the incomplete variable and the missingness mechanism in parametric form. The conditional distribution of z given variables (x, y) is normal and the probability density function is given by:

$$f(z|x,y) = N(\alpha_0 + \alpha_1 x + \alpha_2 y, \delta_z^2), \tag{2.11}$$

where parameters $(\alpha_0, \alpha_1, \alpha_2, \delta_z^2)$, denoted with $\boldsymbol{\theta}$, are unknown and to be solved.

The missingness mechanism is modeled as the conditional distribution of s given variables (x, y, z) (e.g., specified as a logit model). The unknown parameters in the missingness mechanism, denoted with $\boldsymbol{\psi}$, then consist of the intercept and slopes.

Under the NMAR mechanism, valid estimation requires that the missingness mechanism be modeled as part of the parameter estimation process (Rubin 1976). Then parameters are estimated by maximizing the joint likelihood of the two models through MCMC approaches. In this example above, the joint likelihood is:

$$\begin{aligned} & l_{full}(z_{obs}, s|x, y; \boldsymbol{\theta}, \boldsymbol{\psi}) \\ & = \textstyle\sum_{i=1}^{m} \ln[\Pr(s_i, z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \sum_{i=m+1}^{m+n} \ln[\Pr(s_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] \\ & = \textstyle\sum_{i=1}^{m} \ln[\Pr(s_i|x_i, y_i, z_i; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \\ & \quad \textstyle\sum_{i=m+1}^{m+n} \ln[\int \Pr(s_i|x_i, y_i, z; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})\, dz] \end{aligned} \tag{2.12}$$

In Equation (2.12), the first term denotes the summation of log-likelihood over the complete observations ($i = 1, \dots, m$), and the second term denotes the summation of log-likelihood over the incomplete observations ($i = m+1, \dots, m+$

n). The second term involves the integration over the joint probability $\Pr(s_i = 0, z|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})$ with respect to z since the underlying z_i is unknown. In our method, variables x and y are always conditioned on since they are complete variables without missing values.

The likelihood function of (2.12) can be maximized using the expectation maximization (EM) algorithm (Dempster et al. 1977). During each iteration, the expectation (E) step involves the calculation of the expected log-likelihood over the posterior distribution of z given observed values of x, y and s as well as the estimation of $(\boldsymbol{\theta}, \boldsymbol{\psi})$ at the current iteration. The maximization (M) step maximizes the expectation outcome in the E step and obtains updated parameter estimation for $(\boldsymbol{\theta}, \boldsymbol{\psi})$. Due to the difficulty in obtaining a closed form formula for the expectation step, we employ the Monte Carlo EM algorithm (Wei and Tanner 1990; Neath 2013) to numerically approximate the expectation outcome by sampling from the posterior distribution of z using the Metropolis–Hastings algorithm, an MCMC method.[11] We checked the convergence of the EM algorithm and the iterations are terminated when the absolute difference between the *t*-th and the *t*+1-th iteration for parameter $\boldsymbol{\theta}$ is less than a threshold (e.g., 10^{-3}). Table 2-3 presents the pseudocode for obtaining maximum likelihood estimators for the parameters in the conditional distribution of z and in the missingness mechanism.

After obtaining the estimation of $(\boldsymbol{\theta}, \boldsymbol{\psi})$, we derive the mean, variance, and correlation with other variables of z. These estimates can be substituted to the estimation of the regression coefficients of the linear regression model.

In presenting the Monte Carlo maximum likelihood estimation approach, the missing values occur in the independent variable. The Monte Carlo maximum likelihood estimation can also be used to handle missing values in the dependent

[11] The technical details of implementing the Monte Carlo likelihood estimation is provided in Appendix 1.3.

variable. In this situation, obtaining the estimates for $\boldsymbol{\theta}$ already reaches the goal of estimating the regression coefficients.

Algorithm 2.3: Monte Carlo maximum likelihood estimation to correct bias caused by missing values

Data:

Z, a column vector of length $m+n$, denoting the incomplete variable
// Records are sorted such that the first m values of Z are observed and the last n values are missing.

X, a $(m+n) \times k$ matrix containing $m+n$ records and k complete variables
// The algorithm does not distinguish dependent and independent variables. Regression coefficients are calculated after obtaining output of the algorithm, as noted in Section 2.3.2. For the estimation of regression coefficients in Section 2.5.1, since the dependent variable and an independent variable are both complete, they compose X.

S, a column vector of length $m+n$, indicating the missingness of Z

Input:

StoppingCriterion, convergence condition of EM iterations specified by user, initialized as *FALSE*

c, the number of samples to be drawn for each incomplete record

Output:

$\widehat{\boldsymbol{\theta}}$, parameter estimates in the conditional distribution model:

$$f(z|\boldsymbol{x}, y) = N(\alpha_0 + \alpha_1 \boldsymbol{x} + \alpha_2 y, \delta_z^2)$$

$\widehat{\boldsymbol{\psi}}$, parameter estimates in the model of missingness mechanism:

$$\Pr(s|\boldsymbol{x}, y, z; \boldsymbol{\psi})$$

1 $t \leftarrow 0$
2 initialize parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ to be $\boldsymbol{\theta}^t$ and $\boldsymbol{\psi}^t$
3 **while** *StoppingCriterion* == false **do**
// **Expectation based on MCMC sampling**
4 **for** each i in $\{m+1, m+2, \ldots, m+n\}$ // incomplete records
5 draw c samples for the incomplete record $(?, \boldsymbol{x}_i)$, where the sampled values of $\tilde{z}$ are drawn from the posterior distribution of z given $\boldsymbol{x}$, y and s, namely $\Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})$;
6 **end for**
// **Maximization**
7 combine the randomly sampled records $(n \times c)$ with the m complete records, letting weights for each randomly sampled record to be $1/c$ while for each complete record to be 1;
8 $t \leftarrow t+1$;
9 estimate parameter $\boldsymbol{\theta}$ using the combined records with corresponding weights and generate updated estimates $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta^t}$;
10 estimate parameter $\boldsymbol{\psi}$ using the combined records with corresponding weights and generate updated estimates $\widehat{\boldsymbol{\psi}} = \boldsymbol{\psi^t}$;
11 evaluate *StoppingCriterion*;
12 **end while**
13 **return** $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\psi}}$

Table 2-3 Algorithm of Monte Carlo Maximum Likelihood Estimation to Correct Bias Caused by Missing Values

In summary, the two proposed approaches in this section aim at incorporating the missingness mechanism and maximizing effective usage of observable information including the complete and incomplete records. The semi-supervised imputation approach focuses on imputation accuracy for data analytics tasks, the Monte Carlo maximum likelihood method focuses on obtaining valid estimation of regression coefficients. Motivated by different objectives, the former approach gains the flexibility in imputing missing values with different machine learning algorithms, and the latter approach maintains high statistical validity to promote rigorous empirical analysis with missing values.

2.4 Evaluation of Semi-Supervised Missing Value Imputation Method

In this section, we evaluate the imputation accuracy of the proposed semi-supervised imputation approach. We first demonstrate the improved imputation accuracy of the proposed method in simulation studies, then evaluate the method in real-world data to show that the proposed imputation approach promotes better prediction accuracy in data analytics tasks.

2.4.1 Numerical Analysis using Simulations

This sub-section corroborates our theoretical analysis in Section 2.3.1.2 using numerical analysis. Moreover, simulation results suggest that our semi-supervised imputation method outperforms traditional imputation methods when used with different machine learning algorithms such as neural network and gradient boosting.

2.4.1.1 Evaluation in Terms of the Likelihood Function

Data Generation

Without loss of generality, we consider the case where there is only one complete variable x. This variable can be considered a composite variable of several complete variables. The incomplete variable z is simulated from a linear function of x, namely:

$$z = \beta_0 + \beta_1 x + \varepsilon,$$

where $x \sim N(3, 1.5)$ and $\varepsilon \sim N(0, 1)$ are randomly drawn from the normal distributions with sample size being 2,000. Coefficients β_0 and β_1 are set as 1, -1, respectively.

The missingness indicator s is simulated by a logistic model as follows:

$$p(s = 1|z, x; \boldsymbol{\psi}) = \frac{1}{(1 + e^{-(\psi_0 + \psi_1 z + \psi_2 x)})}.$$

Parameter ψ_2 is set as 0.5. We vary the missingness mechanism by allowing the coefficient ψ_1 to take its value from {0, 1, 2, …, 10}. As ψ_1 increases, the missingness indicator is correlated to the incomplete variable to a larger extent. The intercept term ψ_0 is tuned so that the missing value percentage reaches the desired level. In this simulation setting, the missing value percentage is set as 20%, 30%, 40%, 50%, 60%, 70%, or 80%.[12]

Parameter Estimation Methods

After the generation of variables and the missingness, we proceed with the parameter estimation. More specifically, for the proposed method, the functional form of the conditional distribution, $q(z|\boldsymbol{x}, \boldsymbol{\theta})$, and the missingness mechanism, $p(s = 1|z, x; \boldsymbol{\psi})$, are correctly specified as the underlying data generation models. In this simulation analysis, $\boldsymbol{\theta}$ contains the coefficients (β_0, β_1), and $\boldsymbol{\psi}$ contains the coefficients (ψ_0, ψ_1, ψ_2). The parameters $\boldsymbol{\theta}$ are estimated with ordinary regression; and the parameters $\boldsymbol{\psi}$ are estimated with logistic regression. We obtain the estimates of (β_0, β_1) and (ψ_0, ψ_1, ψ_2) in Step 1 and Step 2, respectively. In Step 2, since the

[12] For the logit model, the missing percentage is monotonic to ψ_0, so a binary search for ψ_0 using the trial method within a large enough range will generate the desired value of ψ_0. King and Zeng (2001) formally discussed the implication of the intercept for logistic regression. When all variables are normally distributed, then an appropriate value of ψ_0 can be calculated by letting the expectation of s, to be the desired response percentage. For instance, if the missing percentage is 20%, then the intercept term ψ_0 is determined by solving $E(s|x, z) = 1 - 20\%$.

incomplete variable z is continuous, the residual term $\varepsilon_i = z_i - (\beta_0 + \beta_1 x)$ is discretized to one hundred bins (by dividing the range of the residual to one hundred equal length intervals); therefore, each incomplete record is expanded to one hundred rows and the weight of each bin is the percentage of its occurrence frequency, as depicted in Section 2.3.1.3.

The benchmark case assumes MAR: the estimate for ψ_1 is set as 0, and the estimates for (ψ_0, ψ_2) are set as the coefficients obtained from regressing s on x with a logistic model.

Results of Negative Log-likelihood

Figure 2-1 illustrates the decrease percentage of the negative log-likelihood generated by our semi-supervised imputation method compared to the benchmark, which is calculated as the reduction in negative log-likelihood scaled by the value of the benchmark. Each line is plotted with the 95% confidence interval over 200 simulation replications.

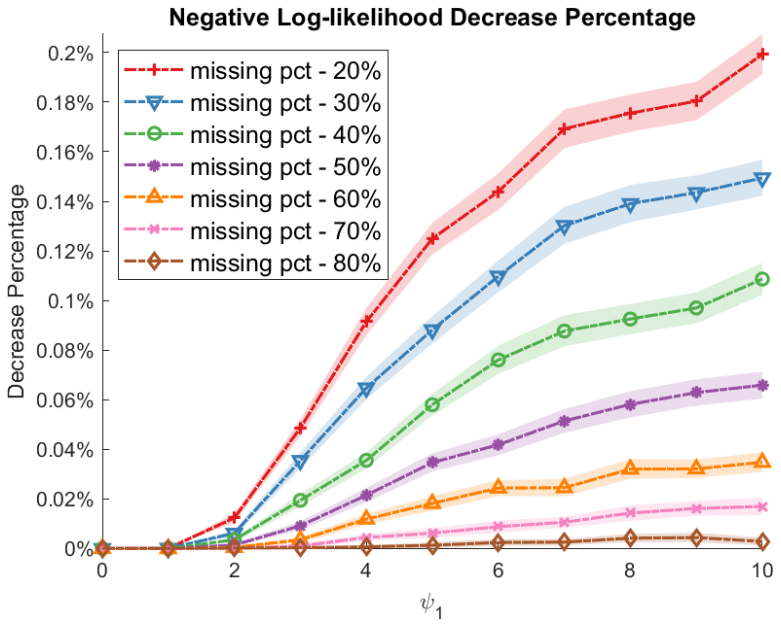


Figure 2-1 Decrease Percentage of Negative Log-likelihood

Results show that the proposed semi-supervised imputation method increases the likelihood function L^{s-s} under different missingness mechanisms and percentages

of missing values, which is consistent with Theorem 2.1. Moreover, as the missing value percentage increases, the likelihood function L^{s-s} is enhanced to a lesser extent, which is consistent with the theory in missing values literature that the information loss is related to the missing value percentage (Rubin 1987, p.132).

In Figure 2-2, we show the decrease percentage of the negative log-likelihood under misspecification of the missingness mechanism. Namely when the underlying true mechanism is a probit model instead of a logistic model. Results show that the missing specification of the classification model (logistic or probit model) has little influence on the likelihood function outcome.

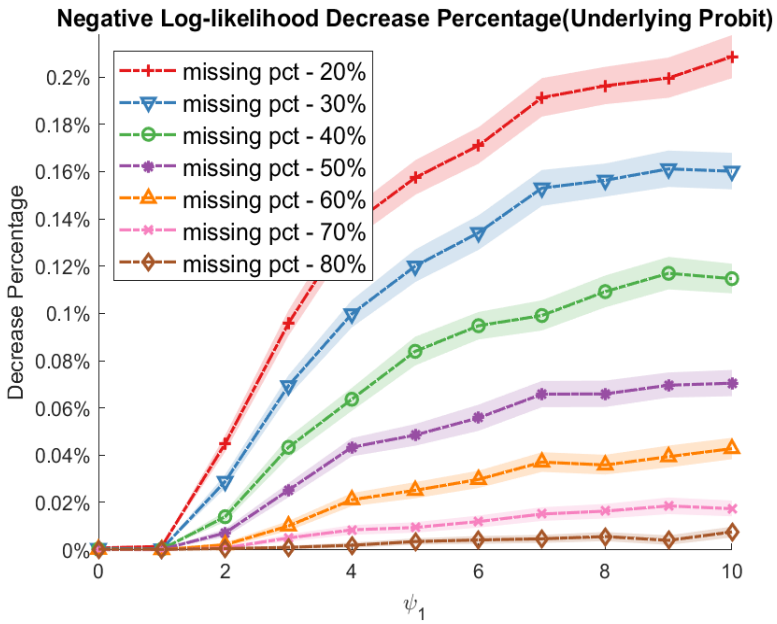


Figure 2-2 Decrease Percentage of Negative Log-likelihood with Probit Model of Missingness Mechanism

2.4.1.2 Evaluation in Terms of Imputation Accuracy

The theorems in Section 2.3.1.2 and simulation results in Section 2.4.1.1 show that our approach generates a greater likelihood value compared to the traditional imputation, which provide support to our semi-supervised imputation approach. In the subsequent simulation analysis, we show that our approach produces better imputation accuracy over traditional approach based on machine learning models. To

examine the imputation accuracy of our proposed method, we employ a relatively complex data generation process to reflect the complexity of real-world data. We simulate one incomplete variable and fifteen complete variables.

Data Generation

The incomplete variable z is generated by the following third-degree polynomial function:

$$z = {x_1}^3 + {x_2}^3 + {x_1}^2 x_2 + {x_2}^2 x_1 + {x_1}^2 + {x_2}^2 + x_1 x_2 + x_1 + x_2 + \epsilon\,,$$

wherein the two complete variables, x_1 and x_2, are randomly drawn from standard normal distribution, and the noise, ϵ, is drawn from normal distribution of $N(0,10)$. In addition, we simulate three complete variables, x_3, x_4, and x_5. These three variables are related to x_1 and/or x_2 following the equations below:

$$x_3 = x_1 + \epsilon,$$

$$x_4 = x_2 + \epsilon,$$

$$x_5 = x_1 x_2 + \epsilon,$$

where ϵ indicates that the noise is randomly drawn from a standard normal distribution. In addition, we generate ten variables $x_6 \sim x_{15}$ randomly drawn from standard normal distribution, which are uncorrelated with all other variables.

The missingness indicator s of variable z is generated by a logistic model:

$$p(s = 1|z, x_1, x_2; \boldsymbol{\psi}) = \frac{1}{1 + e^{-(\psi_0 + \psi_1 z + \psi_2 x_1 + \psi_3 x_2)}}.$$

Similar to the previous simulation setting, we vary the missingness mechanism by allowing the coefficient ψ_1 to take its value from $\{0, 1, \ldots, 10\}$. The other two coefficients ψ_2, ψ_3, are set as 1. We also vary ψ_0 so that missing value percentage is set as 20%, 30%, 40%, 50%, 60%, 70%, or 80%.

Imputation Methods

To evaluate the performance of our method with different machine learning algorithms, we employ different methods to model the conditional distribution,

$q(z|\boldsymbol{x}, \boldsymbol{\theta})$. These methods include (a) linear SVM, (b) SVM with radial basis function kernel, (c) two-layer neural network with sigmoid activation function, (d) CART decision tree, (e) random forest, and (f) gradient boosting machine (GBM) by Friedman (2001).[13] These machine learning algorithms can be used in data analytics tasks to impute missing values. In particular, we use complete observations to train the machine learning model that maps from complete variables to the incomplete variable, and then use the obtained machine learning model to impute the missing values of z. However, although machine learning algorithms such as neural network models or the gradient boosting machine are advanced in approximating complex relationship between the incomplete variable and other complete variables, the traditional way of using them does not incorporate the missingness mechanism and consequently implicitly assume MAR. The proposed imputation method aims at improving the imputation accuracy by incorporating the missingness mechanism.

For the proposed method, since the incomplete variable z is continuous, the residual term is discretized to fifty bins. For the missingness mechanism, all the variables, $x_1 \sim x_{15}$, are employed for classifying the missingness through the logistic model. In this sense, the missingness mechanism is correctly specified as the logit model but all variables are used (although only x_1, x_2, z are relevant). Results are qualitatively similar when the underlying missingness is generated by a probit model while the imputation process assumes a logit model representing the missingness mechanism.

[13] The software used for the machine learning algorithms is Matlab. For the linear SVM and radial SVM algorithms, the function is "fitrsvm" with the corresponding linear and radial kernel specification. The epsilon value is 1.23 according to the default rule for deciding epsilon and the default C (i.e., box constraint) is 1. For the neural network algorithm, the function being used is "fitnet"; the first hidden layer size is 5 and the second hidden layer size is 2. The training function is Levenberg-Marquardt by default. For the CART decision tree algorithm, the function being used is "fitrtree" and the quadratic error tolerance coefficient is 0.1 to control the tree splitting. For the random forest and gradient boosting, the function is "fitrensemble" with the corresponding ensembling methods being specified as 'bag' and 'LSBoost', respectively. The number of learners is set as 50.

Finally, to gain more insights on the property of the proposed method, we also evaluate the imputation accuracy of the proposed method when it is applied to a simple linear imputation model (i.e., estimating a linear model that maps from complete variables to the incomplete variable z using the complete observations). With respect to linear imputation, a state-of-the-art multiple imputation method for NMAR, implemented in the "miceMNAR" R package (Galimard et al. 2018), is also evaluated. This approach requires a Heckman's model for data generation. In particular, the error term in the imputation model is assumed to follow normal distribution, so that the NMAR mechanism is compatible with the Heckman's model. The multiple imputations are created from three steps (Galimard et al. 2018). First, maximum likelihood estimator gives estimates for parameters of the Heckman's model and their variance-covariance matrix. Second, parameters of the Heckman's model are drawn from their posterior distribution. Third, imputed values are drawn from their predictive distribution.

To obtain imputed values from "miceMNAR", we average the results of multiple imputations (with the number of imputations being fifty). Moreover, to maximize the compatibility of the Heckman's model, during the data generation, we set the underlying missingness mechanism to be a probit model, namely $p(s = 1|z, x_1, x_2; \boldsymbol{\psi}) = \Phi^{-1}(\psi_0 + \psi_1 z + \psi_2 x_1 + \psi_3 x_2)$ and Φ is the standard normal cumulative distribution function, other (relevant) factors being equal. For the proposed method, we implement it by using the probit model as the missingness mechanism.

Results of Imputation accuracy

The evaluation metrics for measuring imputation accuracy is the mean absolute error (MAE) of the imputed values normalized by the standard deviation of z. Results are based on the average MAE over 200 simulations. Table 2-4 lists the average MAE of each benchmark imputation method and the average MAE of the

proposed method building upon the corresponding benchmark when $\psi_1 = 0$, 5, or 10. The *t*-statistics for comparing the average MAE of the two methods across the two hundred replications of simulation are listed within the parentheses. It can be seen that when ψ_1 is 5 or 10 (namely NMAR), under different missing value percentages, the proposed method reduces MAE; the difference in MAE between the proposed method and benchmarks in Table 2-4 is statistically significant at the 5% level. Results are qualitatively similar under other values of ψ_1, as shown in Figure 2-3.

Figure 2-3 presents the reduction percentage of MAE by our proposed imputation method. The reduction percentage of MAE is calculated as the reduction in MAE scaled by the value of the benchmark. Each line is plotted with the corresponding 95% confidence interval of the mean decrease percentage of MAE over 200 simulations. Figure 2-3 shows that our method increases the imputation accuracy of traditional imputation models under different missing percentages of z. It is worth emphasizing that, in Figure 2-3 (or Table 2-4), when $\psi_1 = 0$, our method increases the MAE by 6.4% for SVM (radial kernel) and around 1% for the other five machine learning algorithms. This could be explained by the reason that, since the incomplete variable z is a continuous random variable, we adopt a nonparametric method in Section 2.3.1.3 to approximate its conditional distribution based on its residuals, which introduces approximation error. However, the proposed method largely reduces the MAE when ψ_1 is equal to or larger than one, which means that the benefit of incorporating the missingness mechanism largely exceeds the cost of approximation error in the nonparametric probability distribution estimation.

	Imputation Method		Missingness Percentage						
			20%	30%	40%	50%	60%	70%	80%
ψ_1 =0	SVM (Linear)	Benchmark	0.65	0.65	0.65	0.66	0.67	0.69	0.71
		Proposed	0.66	0.66	0.66	0.67	0.68	0.70	0.72
		t-statistic	(-2.08)	(-2.29)	(-2.54)	(-2.43)	(-3.03)	(-2.53)	(-2.04)
	SVM (Radial)	Benchmark	0.70	0.69	0.69	0.69	0.69	0.69	0.70
		Proposed	0.75	0.74	0.73	0.74	0.73	0.74	0.74
		t-statistic	(-11.48)	(-13.39)	(-13.39)	(-15.11)	(-15.90)	(-15.82)	(-13.08)
	Neural Network	Benchmark	0.70	0.70	0.69	0.69	0.70	0.71	0.75
		Proposed	0.71	0.71	0.69	0.69	0.70	0.70	0.73
		t-statistic	(-1.33)	(-1.28)	(-1.00)	(-0.61)	(0.75)	(1.18)	(1.84)
	Decision Tree	Benchmark	0.69	0.69	0.69	0.69	0.70	0.71	0.73
		Proposed	0.70	0.69	0.69	0.70	0.70	0.71	0.74
		t-statistic	(-1.29)	(-1.50)	(-1.36)	(-1.78)	(-1.22)	(-1.64)	(-1.78)
	Random Forest	Benchmark	0.70	0.69	0.69	0.69	0.68	0.69	0.70
		Proposed	0.71	0.70	0.69	0.69	0.69	0.69	0.70
		t-statistic	(-2.60)	(-2.18)	(-1.22)	(-2.78)	(-0.80)	(-0.87)	(0.18)
	Gradient Boosting	Benchmark	0.69	0.69	0.69	0.69	0.70	0.72	0.74
		Proposed	0.70	0.69	0.69	0.70	0.71	0.72	0.75
		t-statistic	(-1.55)	(-1.47)	(-1.92)	(-1.91)	(-1.74)	(-1.82)	(-1.95)
ψ_1 =5	SVM (Linear)	Benchmark	1.28	1.21	1.19	1.20	1.24	1.32	1.45
		Proposed	1.17	1.09	1.06	1.08	1.14	1.24	1.39
		t-statistic	(16.78)	(20.18)	(22.02)	(24.70)	(20.81)	(17.25)	(12.36)
	SVM (Radial)	Benchmark	1.40	1.33	1.31	1.33	1.38	1.47	1.62
		Proposed	1.06	0.99	0.97	1.00	1.07	1.17	1.32
		t-statistic	(66.96)	(70.32)	(76.03)	(76.98)	(77.02)	(72.84)	(67.35)
	Neural Network	Benchmark	1.44	1.35	1.35	1.34	1.39	1.48	1.65
		Proposed	1.23	1.06	1.05	1.04	1.12	1.22	1.40
		t-statistic	(19.71)	(35.13)	(17.86)	(41.74)	(40.87)	(33.19)	(28.15)
	Decision Tree	Benchmark	1.42	1.34	1.32	1.33	1.37	1.45	1.59
		Proposed	1.22	1.07	1.04	1.07	1.14	1.25	1.40
		t-statistic	(24.77)	(43.69)	(58.70)	(50.03)	(48.33)	(43.47)	(43.97)
	Random Forest	Benchmark	1.48	1.39	1.36	1.36	1.40	1.48	1.63
		Proposed	1.30	1.08	1.04	1.06	1.12	1.23	1.39
		t-statistic	(22.40)	(50.06)	(58.23)	(71.42)	(71.74)	(68.22)	(60.56)
	Gradient Boosting	Benchmark	1.39	1.32	1.29	1.29	1.33	1.41	1.53
		Proposed	1.22	1.11	1.09	1.12	1.19	1.31	1.47
		t-statistic	(24.37)	(35.98)	(40.13)	(36.61)	(31.78)	(21.35)	(11.45)
ψ_1 =10	SVM (Linear)	Benchmark	1.28	1.21	1.19	1.20	1.24	1.31	1.44
		Proposed	1.17	1.09	1.07	1.09	1.15	1.23	1.38
		t-statistic	(17.69)	(19.08)	(19.21)	(18.97)	(17.67)	(17.56)	(12.21)
	SVM (Radial)	Benchmark	1.40	1.32	1.31	1.33	1.38	1.47	1.62
		Proposed	1.06	0.98	0.97	1.00	1.06	1.17	1.32
		t-statistic	(65.92)	(68.35)	(71.68)	(74.89)	(77.91)	(77.83)	(72.89)
	Neural Network	Benchmark	1.45	1.36	1.35	1.35	1.39	1.49	1.66
		Proposed	1.26	1.05	1.02	1.04	1.11	1.22	1.40
		t-statistic	(17.45)	(35.67)	(39.90)	(43.98)	(40.88)	(33.07)	(27.84)
	Decision Tree	Benchmark	1.42	1.34	1.32	1.32	1.36	1.44	1.59
		Proposed	1.22	1.08	1.03	1.06	1.13	1.23	1.39
		t-statistic	(23.37)	(37.99)	(53.61)	(50.44)	(50.82)	(45.29)	(43.00)
	Random Forest	Benchmark	1.47	1.39	1.35	1.36	1.40	1.48	1.63
		Proposed	1.29	1.09	1.03	1.05	1.11	1.21	1.39
		t-statistic	(22.11)	(40.14)	(64.07)	(67.36)	(79.62)	(72.78)	(66.20)
	Gradient Boosting	Benchmark	1.39	1.32	1.29	1.29	1.33	1.40	1.53
		Proposed	1.22	1.11	1.08	1.12	1.19	1.30	1.46
		t-statistic	(22.67)	(32.87)	(36.37)	(32.40)	(28.66)	(21.43)	(12.64)

Table 2-4 Comparison of MAE under Different Missing Value Percentages

Moreover, the proposed method obtains relatively stable MAE decrease percentage when ψ_1 ranges from 1 to 10. Although the results might be subject to the

granularity of the values of ψ_1 (i.e., we increase the value of ψ_1 by 1 rather than a more refined magnitude such as 0.1), we observe a property of the proposed method that the error reduction is positive and tends to be stable even if ψ_1 becomes large.

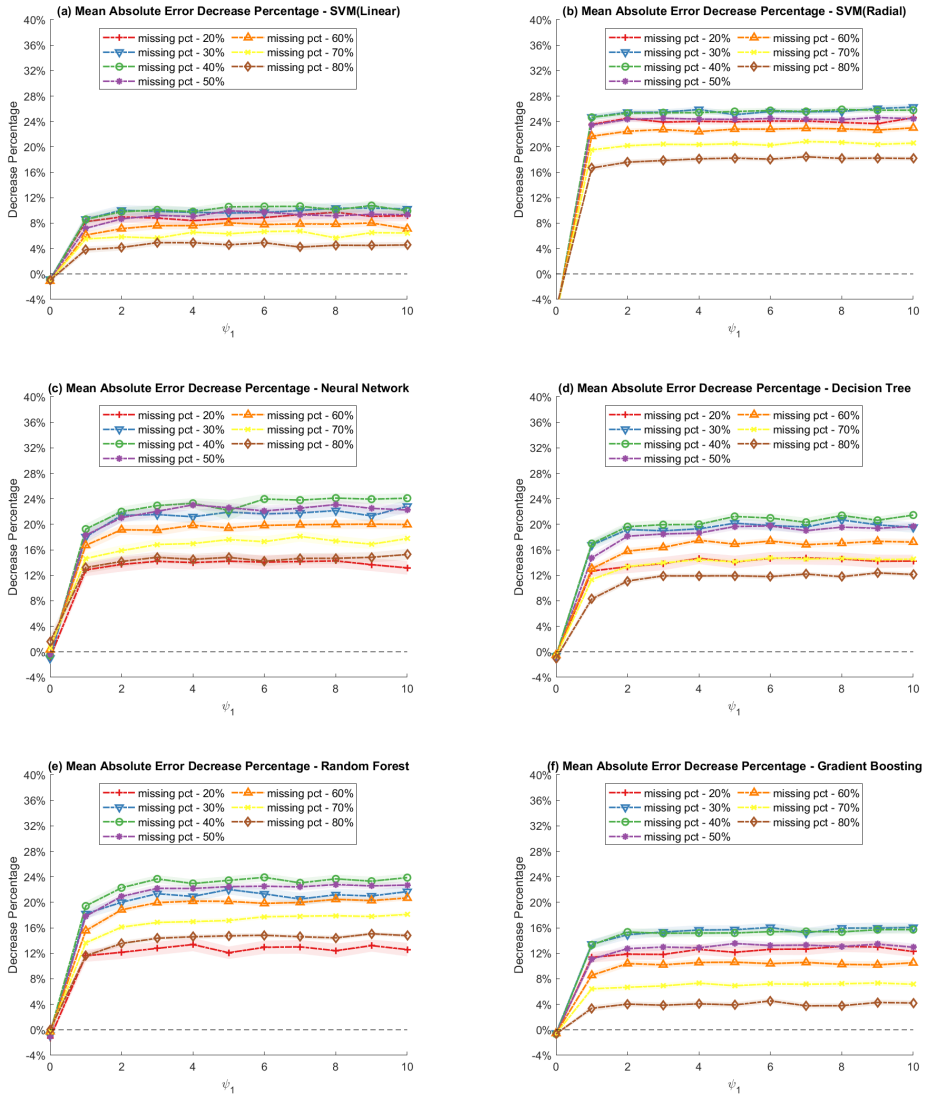


Figure 2-3 Decrease Percentage of MAE by Semi-Supervised Imputation Method

Finally, an interesting finding is that the effect of missing data percentage (e.g., 20%, 40%, 60%, 80%) on imputation accuracy is non-monotonic. We provide possible explanation by hypothesizing two extremes. When the missing data percentage is at the low extreme, the benchmark imputation model, even without considering the missingness mechanism, tends to approximate the underlying relationship between the incomplete variable and complete variables well, thus the

benefit of modeling the missingness mechanism tends to be limited. When the missing data percentage is at the high extreme, the data for estimating the conditional probability or distribution of the incomplete variable, which is required by the proposed method, is limited. Thus, the effectiveness of incorporating the missingness mechanism into imputation is also reduced. It would be a promising future research direction to investigate the influence of missing value percentage, in addition to the missingness mechanism, on the choice of missing value handling methods.

Table 2-5 reports MAE results of the linear imputation models: (1) simple linear imputation, (2) the proposed semi-supervised imputation approach, and (3) the miceMNAR approach with values of ψ_1 being 0, 2, 4, 6, 8, 10. The standard deviations of MAE over two hundred simulation replications are listed in parentheses. To facilitate the comparison of imputation accuracy, Figure 2-4 shows imputation accuracy of the linear imputation models: (a) the MAE of simple linear imputation, (b) the reduction percentage of MAE by the proposed semi-supervised imputation approach compared to simple linear imputation, (c) the reduction percentage of MAE by the miceMNAR approach compared to simple linear imputation, and (d) the MAE difference between miceMNAR and the proposed approach (i.e., MAE of miceMNAR minus that of the proposed approach).

From Figure 2-4(b) (or comparing results of the linear models in Table 2-5), we observe that the proposed method enhances imputation accuracy over all the non-zero values of ψ_1 and different missing value percentages (the MAE decrease percentage generally ranges from 10% to 20%). When $\psi_1 = 0$, it obtains close performance as the simple linear imputation model. From Figure 2-4(c), we observe that miceMNAR generally enhances imputation accuracy when $\psi_1 = 1, 2, \ldots$ and the missing value percentage is no more than 60% (the MAE decrease percentage generally ranges from 40% to 60%). However, when $\psi_1 = 0$, the imputation error largely increases. When the missing value percentage is as high as 70%, miceMNAR

reduces MAE by a limited extend. When the missing value percentage is 80%, it often results in increased MAE than the simple linear imputation model, meaning that it could be harmful to incorporate the missingness mechanism through miceMNAR when the underlying missingness mechanism tends to be MAR.

Imputation Method		Missingness Percentage						
		20%	30%	40%	50%	60%	70%	80%
$\psi_1 = 0$	Simple linear	0.66	0.68	0.71	0.74	0.79	0.87	1.09
		(0.03)	(0.02)	(0.03)	(0.03)	(0.04)	(0.06)	(0.09)
	Proposed	0.69	0.71	0.73	0.77	0.81	0.90	1.14
		(0.04)	(0.04)	(0.05)	(0.06)	(0.07)	(0.08)	(0.13)
	miceMNAR	0.86	1.03	1.18	1.37	1.60	1.94	2.62
		(0.05)	(0.07)	(0.07)	(0.08)	(0.11)	(0.18)	(0.24)
$\psi_1 = 2$	Simple linear	1.30	1.24	1.22	1.22	1.25	1.29	1.39
		(0.05)	(0.04)	(0.05)	(0.04)	(0.05)	(0.05)	(0.07)
	Proposed	1.09	1.05	1.04	1.05	1.08	1.15	1.25
		(0.10)	(0.08)	(0.09)	(0.08)	(0.07)	(0.08)	(0.11)
	miceMNAR	0.68	0.58	0.53	0.52	0.59	1.02	2.72
		(0.03)	(0.02)	(0.02)	(0.05)	(0.11)	(0.40)	(0.97)
$\psi_1 = 4$	Simple linear	1.31	1.25	1.23	1.23	1.25	1.30	1.40
		(0.05)	(0.04)	(0.04)	(0.04)	(0.04)	(0.04)	(0.06)
	Proposed	1.11	1.04	1.03	1.06	1.08	1.15	1.26
		(0.09)	(0.09)	(0.08)	(0.08)	(0.07)	(0.08)	(0.11)
	miceMNAR	0.69	0.59	0.54	0.53	0.62	1.17	2.15
		(0.03)	(0.02)	(0.02)	(0.06)	(0.19)	(0.47)	(0.85)
$\psi_1 = 6$	Simple linear	1.31	1.25	1.23	1.24	1.25	1.31	1.40
		(0.04)	(0.04)	(0.04)	(0.04)	(0.04)	(0.05)	(0.06)
	Proposed	1.09	1.02	1.02	1.05	1.09	1.17	1.27
		(0.09)	(0.08)	(0.08)	(0.08)	(0.07)	(0.08)	(0.09)
	miceMNAR	0.65	0.58	0.54	0.53	0.59	1.11	1.41
		(0.03)	(0.02)	(0.02)	(0.07)	(0.12)	(0.50)	(0.64)
$\psi_1 = 8$	Simple linear	1.31	1.25	1.23	1.23	1.26	1.31	1.41
		(0.05)	(0.05)	(0.04)	(0.04)	(0.05)	(0.05)	(0.07)
	Proposed	1.10	1.05	1.02	1.05	1.09	1.16	1.28
		(0.09)	(0.08)	(0.08)	(0.07)	(0.07)	(0.08)	(0.10)
	miceMNAR	0.63	0.59	0.53	0.53	0.64	0.92	0.95
		(0.03)	(0.02)	(0.02)	(0.06)	(0.16)	(0.37)	(0.43)
$\psi_1 = 10$	Simple linear	1.30	1.26	1.24	1.23	1.26	1.30	1.39
		(0.05)	(0.04)	(0.04)	(0.04)	(0.05)	(0.05)	(0.06)
	Proposed	1.07	1.05	1.04	1.05	1.09	1.16	1.25
		(0.09)	(0.08)	(0.08)	(0.07)	(0.08)	(0.08)	(0.10)
	miceMNAR	0.60	0.58	0.54	0.53	0.60	0.71	0.75
		(0.02)	(0.02)	(0.03)	(0.08)	(0.15)	(0.24)	(0.28)

Table 2-5 Imputation Accuracy (MAE) Using Linear Imputation Models

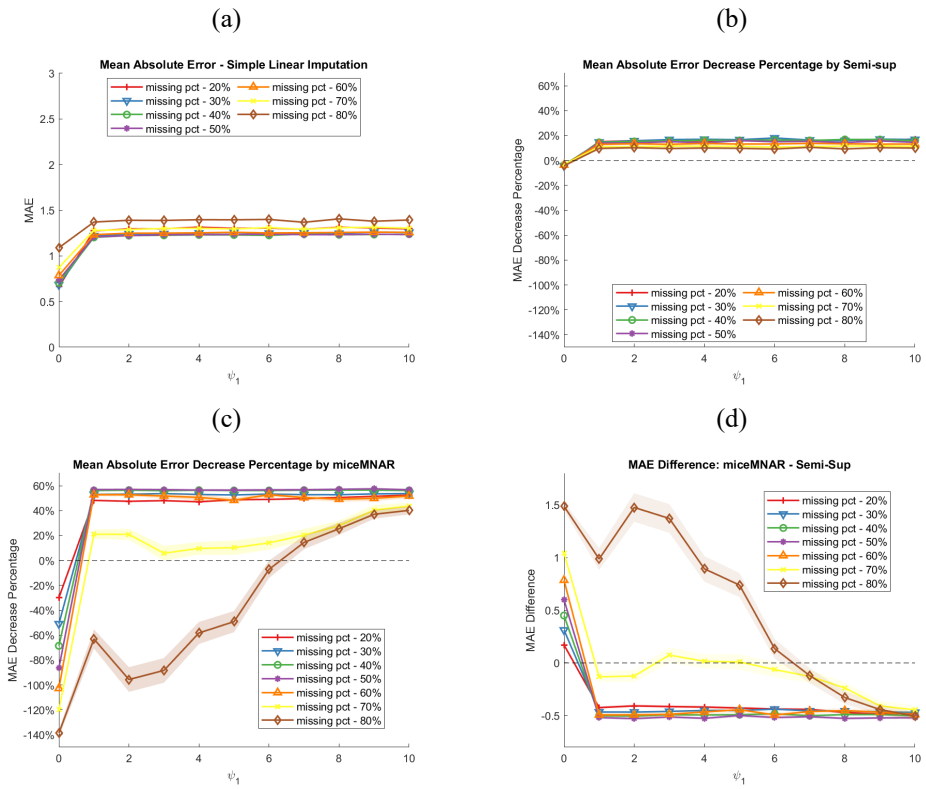


Figure 2-4 Comparison of Imputation Accuracy of Linear Imputation Models

The performance of miceMNAR over the proposed method when ψ_1 equals 1, 2, 3, …, with the missing value percentage being moderate can be explained by the maximum likelihood estimator for the Heckman model (step 1 of miceMNAR). Although the proposed method gains theoretical support in that it enhances the imputation accuracy compared to the benchmark of the simple linear imputation model, it does not necessarily obtain estimates of parameters that maximize the objective likelihood function, whereas the multiple imputations of miceMNAR are created based on the maximum likelihood estimator for the Heckman model. It can be shown that the Heckman model is equivalent to the model of missingness mechanism up to different parameterizations (Galimard et al. 2016).

When it comes to the MAR mechanism, namely $\psi_1 = 0$, there is little advantage in incorporating the missingness mechanism. Since miceMNAR requires normal distributional assumption for the error term, which is violated in our

simulation setting, it leads to increased imputation error when $\psi_1 = 0$. The increase in imputation error is exacerbated as the missing value percentage increases (e.g., at 80%). This can be explained by the increasing variance (uncertainty) of the maximum likelihood estimator for the Heckman model with fewer complete observations. Finally, estimating the Heckman model requires the exclusion-restriction criteria to avoid collinearity issues; that is, at least one covariate is included in the missingness mechanism and not in the imputation model. In practical implementation, researchers need to specify such covariate based on their own domain knowledge. Although such a covariate is not available in our simulation setting, both the proposed method and miceMNAR are evaluated with a level playing field.

Overall, by comparing the linear imputation models, we show that the miceMNAR method tends to achieve better imputation performance than the proposed method when the missing value percentage is low or moderate and when there is a strong reason to suspect that the missingness mechanism is NMAR. The proposed method is resilient to the non-normal distribution of the error term in the imputation model and does not harm imputation accuracy when the missingness mechanism tends to be MAR. Since researchers often do not have the prior knowledge on the extent of NMAR, it is necessary to employ an imputation method that is robust over the full spectrum from MAR to NMAR. Finally, with the generalizability of the proposed method to imputation model based on machine learning, the proposed method possesses important qualities for practical concerns.

2.4.2 Experimentation in Real-world Data Sets

The missing values problem is ubiquitous in real-world data analytics. We use two real-world data sets to examine the effectiveness of the proposed method. Both data sets may arguably satisfy the NMAR missing mechanism. Hence, we use the proposed imputation method to impute the most important variables for these two

prediction tasks. Since the underlying true values of the missing values are unknown, we cannot use imputation accuracy as the evaluation metric. Instead, we indirectly show the imputation accuracy by comparing the prediction error after the incomplete predictor is imputed by different imputation methods. Therefore, in this section, different imputation methods are evaluated according to the prediction accuracy of a machine learning model built upon the imputed datasets, which complements the evaluation procedure in Section 2.4.1. The main prediction model used in this section is the state-of-the-art GBM algorithm.[14] Since the initial ideas of gradient boosting (Friedman 2001), various extension models have been proposed, such as XGBoost (Chen and Guestrin 2016) and LightGBM (Ke et al. 2017). These extensions often achieve comparable prediction accuracy (Ke et al. 2017). By experimenting with the well-studied GBM algorithm, we expect that the results would hold in general. The imputation models are trained with the same six algorithms used in Section 2.4.1.

2.4.2.1 Credit Default Prediction

Data Description

The "Home Credit Default Risk" task from a recent Kaggle competition asked to predict whether each applicant will repay their loan according to the applicant's information provided by the Home Credit Group, an international consumer finance provider.

The records in the data set are at the loan application level. There are 307,511 records and 121 predictor variables, which cover various aspects of the applicants, such as credit scores from third-party agencies, real-estate ownership status, employment status, and loan amount. After applying one-hot encoding to transform

[14] We use the fitensemble function (www.mathworks.com/help/stats/fitensemble.html) of Matlab, for the GBM implementation of Friedman (2001).

the categorical variables, we obtained 239 predictor variables.[15] Based on the GBM importance score, Figure 2-5 lists the top ten important variables for predicting loan default (importance scores are normalized so that the importance scores of all the 239 variables add up to one).[16] The three most important variables are: *EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3*. These are the credit scores (continuous variables) provided by third-party agencies. Each of the three variables contains missing values, with the missing rate being 56.4%, 19.8%, and 0.2%, respectively. Given that variable *EXT_SOURCE_1* has the highest importance score and a severe missing values problem, we impute the missing values of this variable with our semi-supervised imputation method.

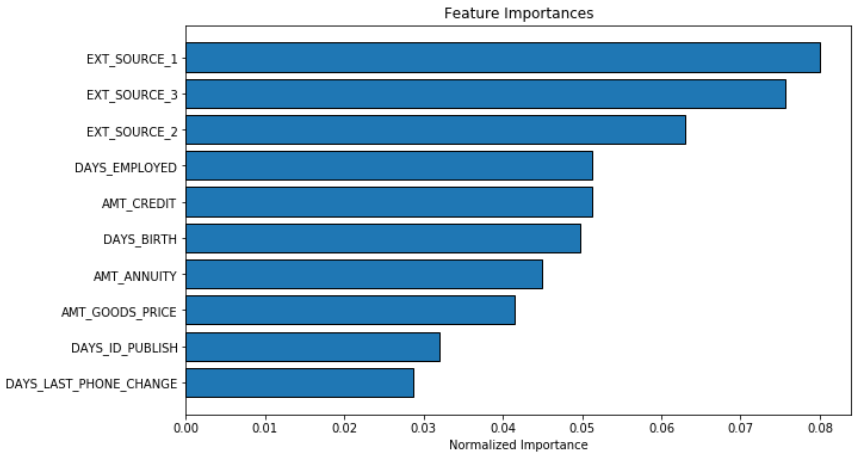


Figure 2-5 Top Ten Important Variables for Predicting Loan Default

In addition to the variable *EXT_SOURCE_1*, a large number of variables of the data set are subject to missing values. To isolate the effectiveness of our imputation method on the focal variable, we constructed the following sampling steps, which were summarized in Table 2-6. We first selected the top 100 important

[15] One-hot encoding is a common process by which categorical variables are converted into a numeric form that could be provided to ML algorithms. For instance, the variable WEEKDAY_APPR_ START indicates on which day of the week did the client apply for the loan. The value for that variable is of string type and comes from {Monday, Tuesday, …, Sunday}. To convert this categorical variable to a numeric one, we construct seven dummy variables indicating each of the weekday.

[16] The predictor importance is calculated by summing changes in the performance matric (e.g., mean squared error (MSE)) due to splits on every predictor and dividing the sum by the number of branch nodes. The change in performance associated with a split is computed as the difference between MSE for the parent node and the total MSE for the two children.

variables and then dropped the variables with a missing rate larger than 60%, which resulted in 11 variables being dropped. Among the remaining 89 variables, 12 of them were still incomplete. We then dropped rows with missing values for any of the 12 variables except for *EXT_SOURCE_1*, since the missing values of this variable will be imputed. Finally, we obtained 201,092 records, and the variable *EXT_SOURCE_1* is missing in 102,562 records (missing rate of 51.0%).[17]

Data Processing Steps	Number of Predictors	Number of Records
Raw data set	121	307,511
One hot encoding	239	307,511
Select top 100 important variables and drop the ones with missing value percentage larger than 60%	89	307,511
Drop records with missing values for any of the 11 incomplete variables	89	201,092

Table 2-6 Credit Default Prediction - Data Pre-processing and Sample Construction Process

Experimentation Results

After the sample construction process, only the predictor variable *EXT_SOURCE_1* contains missing values. This variable is imputed with all the other complete predictors using different imputation methods listed in Table 2-7, with the missingness mechanism being estimated with a logit model. Then the GBM is applied to the imputed data set (with no missing values) to develop the prediction model and to conduct the prediction. In the "Home Credit Default Risk" competition, since the target variable is unbalanced (with the default rate being 8.4%), the organizer employed AUC metrics to evaluate the prediction performance. In this experimentation, we report the averaged AUC by five-fold cross-validation tests. Table 2-7 summarizes the prediction performance of the GBM algorithm using the data set imputed by different methods. Results are qualitatively similar using different random sampling to form the five-fold tests . Columns 2 and 3 report the prediction

[17] Imputing all incomplete variables would be computationally costly and even bring much noise into the data. Therefore, we choose to impute *EXT_SOURCE_1*, the variable with the highest importance ranking in making prediction and a high percentage of missing values.

performance after the incomplete variable is imputed by the traditional method and new method, respectively. Standard deviations of AUC over the five-fold tests are reported in the corresponding parentheses. The improved prediction performance (percentage increase of AUC) is reported in Columns 4.

Benchmark Imputation	AUC of Benchmark	AUC of Proposed	Proposed vs. Benchmark
SVM (Linear)	0.7030 (0.0170)	**0.7222** (0.0298)	2.73%
SVM (Radial)	0.6905 (0.0130)	**0.7133** (0.0175)	3.30%
Neural-2	0.7068 (0.0134)	**0.7089** (0.0120)	0.30%
Decision Tree	**0.7109** (0.0169)	0.7031 (0.0139)	-1.10%
Random Forest	0.7013 (0.0236)	**0.7102** (0.0191)	1.27%
Gradient Boosting	0.7073 (0.0193)	**0.7112** (0.0260)	0.55%

Table 2-7 Credit Default Prediction - AUC Using Different Missing Value Handling Methods

The results in Table 2-7 show that the proposed method outperforms the benchmark imputation method, including SVM, neural network, random forest, and gradient boosting. Compared to SVM (linear) imputation, the proposed method improves the prediction accuracy by 2.73%. When compared to the state-of-the-art gradient boosting algorithm for imputation, our method still improves the prediction accuracy by 0.55%. Without a pre-processing step of imputing missing values, GBM algorithm has an embedded way of handling missing values through the surrogate variable method, achieving an AUC of 0.7046. This is lower than the AUC using benchmark imputations of neural network, decision tree, and gradient boosting.[18]

[18] A surrogate is a substitute for the primary splitter of a node. A good surrogate splits the data in similar way as the primary split. Namely, we are looking for a variable that closely approximate the behavior of the primary split. This technology is invented in the Classification and Regression Trees, CART (Breiman et al. 1984). Moreover, to exclude the possibility that results may be driven by the sample construction process described in Table 2-6, we conduct the experimentation under the situation that no observations or variables are dropped. Under such situation, the GBM using the default surrogate method achieves an accuracy of 0.7063. Compared with the AUC after the sample selection process, which is 0.7046, the difference 0.2% is minimal.

It is worth noting that the proposed method does not generate better prediction performance when the baseline imputation method is the decision tree. A possible reason is that our method is theoretically demonstrated to outperform the benchmark imputation method when the benchmark imputation method is parametric. However, the theoretical property is not proved under the nonparametric situation. Although so far there is no clear boundary for dividing machine learning algorithms into parametric and nonparametric ones, decision tree is sometimes viewed as a nonparametric model (Stekhoven and Bühlmann 2012).[19] In practical applications, we recommend using the proposed method to increase the benchmark imputation methods, such as linear SVM, neural network, random forest, and gradient boosting to achieve relatively robust results as shown in the simulations and real-world experimentations.

2.4.2.2 Earnings Prediction

Data Description

The second data set is used to predict the quarterly earnings of US public firms based on financial statements and analyst consensus forecasts. This data set is another good example that shows the importance of taking into account the NMAR mechanism for missing value imputation. The extant literature shows that analyst forecast data is often not available for small firms and financially distressed firms (Diether et al. 2002), which suggests that the analyst consensus forecasts are likely to be NMAR.

The records in this data set are at the firm-quarter level. For each record, there are 362 variables from firms' quarterly financial statements, which are provided

[19] An omnipotent definition of nonparametric model is still under-developed. According to the definitions by Russell and Norvig (2016, Chapter 10.8), "…. A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model…. A nonparametric model is one that cannot be characterized by a bounded set of parameters." In this chapter, only K-nearest neighbor is acknowledged as a nonparametric method.

by the Compustat database. The analyst consensus forecast is provided by the I/B/E/S database. In total, 363 variables are used for building the prediction model by GBM. The data set contains 120 quarters starting from 1987 Q1 to 2016 Q4. On average, there are 3,164 firms in each quarter. We conducted the prediction on quarterly earnings in 107 consecutive quarters from 1990 Q1 to 2016 Q3. The last quarter of earnings prediction is 2016 Q3 because the next quarter's earnings are required to measure the accuracy of the current quarter's prediction.

During the forecast in each quarter, we used the same quarter data from the past three years to construct the prediction model. We illustrate the construction of the training and test datasets in Figure 2-6.

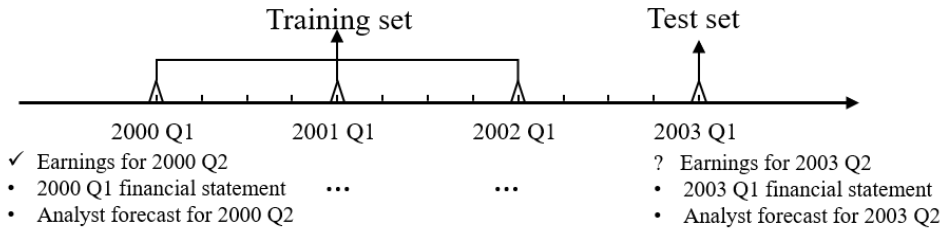


Figure 2-6 Illustration of Training and Test Data Sets Construction

For fiscal quarter 2003 Q1, we used the financial statement data items of 2003 Q1 and the consensus forecast for 2003 Q2, 363 variables in total, to make predictions for the next quarter's earnings (of 2003 Q2). The training data set for building the prediction model consists of the corresponding predictors and the dependent variable in three fiscal quarters, 2002 Q1, 2001 Q1, and 2000 Q1.

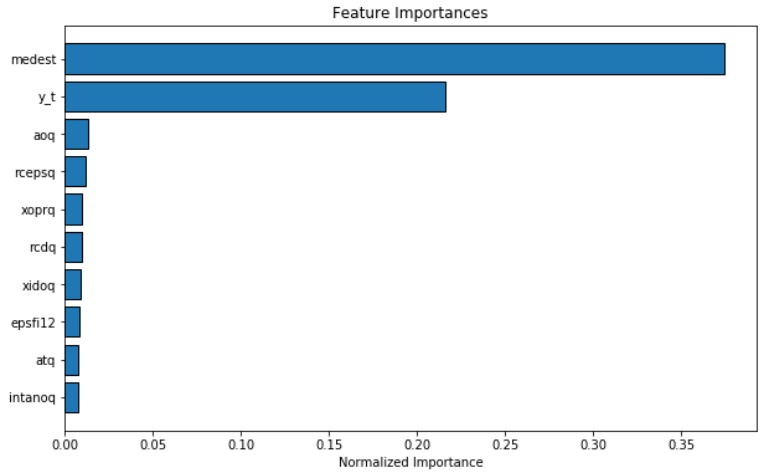


Figure 2-7 Top Ten Important Variables for Predicting Earnings

We conducted a preliminary analysis to find the important variables using GBM. Figure 2-7 lists the top ten important variables for predicting the next quarter's earnings. It is shown that, *medest* (analyst consensus forecast measured by the median of analysts' earnings estimation) and *y_t* (actual earnings of the current quarter) are the two most important variables, and these two variables have much higher importance scores than all other variables. As for the missing values, *y_t* is a complete variable and *medest* is missing for 40.7% of the records. Therefore, in the earnings prediction task, we impute the important predictor, *medest*, aiming at enhancing the prediction accuracy.

Missing Value Percentage	Number of Predictors	Number of Predictors (in Percentage)
0 ≤ percentage <20%	170	46.83%
20% ≤ percentage <40%	76	20.94%
40% ≤ percentage <60%	22	6.06%
60% ≤ percentage <80%	13	3.58%
80% ≤ percentage ≤100%	82	22.59%
Total	363	100.00%

Table 2-8 Earnings Prediction - Missing Value Percentage of Predictors

In addition to the variable *medest*, the variables from Compustat include missing values to different extents. Table 2-8 lists the missing value percentage in this data set.

Imputing the missing values for all the variables would be time consuming and computationally taxing. A common practice in accounting research is to replace the missing values in financial statements with 0 (Koh and Reeb 2015; Li and Mohanram 2014). This missing value handling method is reasonable, since firms do not report data items when the amount incurred is immaterial. We handle the missing variables from Compustat following this practice. Admittedly, our choice is subject to some limitations, since the underlying assumption is that missing values in financial statements are negligible. However, this pre-processing step does not seem to favor our method, and therefore it also serves as the robustness check in that our method can outperform traditional methods when the other missing values are handled differently, as mentioned in Section 2.4.2.1.

Experimentation Results

Since quarterly earnings is a continuous variable, its prediction is a regression problem. We therefore use the absolute prediction error of raw earnings scaled by firms' market capitalization as the evaluation metrics following the earnings prediction literature (Hou et al. 2012; Li and Mohanram 2014).

Benchmark imputation	MAE of Benchmark	MAE of Proposed	Proposed vs. Benchmark
SVM (Linear)	0.1295	**0.1230**	5.28%
SVM (Radial)	0.1005	**0.1002**	0.30%
Neural-2	0.1083	**0.1061**	2.07%
Decision Tree	0.1029	**0.1015**	1.38%
Random Forest	0.1063	**0.1061**	0.19%
Gradient Boosting	0.1129	**0.1107**	1.99%

Table 2-9 Earnings Prediction - MAE Using Different Missing Value Handling Methods

Table 2-9 summarizes the prediction performance using the data set imputed by different methods. The results of the scaled mean absolute error (MAE) in Table 2-9 are averaged across all firm-quarter pairs (330,672 rows in total). Columns 2 and 3 report the prediction performance after the incomplete variable is imputed by the benchmark method and the proposed method, respectively. The percentage decrease

of MAE for earnings prediction is reported in Columns 4. Results demonstrate that our method outperforms the various benchmark imputation methods. Using simple mean imputation, the prediction error is 0.1154, higher than that of benchmark imputation methods using SVM (radial), neural network, decision tree, and random forest. Without the machine learning prediction model GBM, using the simple moving average (MA) model gives prediction errors of 0.1227 (MA(4)).

2.5 Evaluation of Monte Carlo Likelihood Estimation of Regression Coefficients

In this section, we experiment commonly used approaches such as listwise deletion, and the well-established likelihood-based approach, maximum likelihood estimation assuming MAR (denoted with ML-MAR thereafter). We first discuss how the problem emerges that the endogenous sample selection interacts with the NMAR mechanism, then we illustrate the operating characteristics of different missing value handling methods, including our proposed Monte Carlo-based approach.

2.5.1 Simulation Setting

We perform a simulation for the estimation of regression coefficients in the following model:

$$y = \beta_0 + \beta_1 x + \beta_2 z + \varepsilon,$$

where $\binom{x}{z} \sim N(\mu, \sigma^2)$, $\mu = \binom{0}{0}$, $\sigma^2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$, $\varepsilon \sim N(0,1)$. The values of coefficients are set to $\beta_0 = 0$, $\beta_1 = 1$ and $\beta_2 = -1$.

One thousand data samples of the values of (x, z, y) are drawn from the above data generating process.[20] Missing values are imposed on variable z according to the missingness mechanism represented by the following logistic model:

[20] Results are qualitatively similar for sample size of 500 and 200. We did not experiment with small sample sizes such as 50 since we expect that researchers in today's big data environment often have a large sample size but are likely to face nontrivial amount of missing values.

$$p(s = 1|x, y, z; \boldsymbol{\psi}) = \frac{1}{1 + e^{-(\psi_0 + \psi_x x + \psi_y y + \psi_z z)}}$$

where parameter ψ_x is fixed at zero. We vary the missingness mechanism by allowing coefficients ψ_z and ψ_y to take values in $\{0, 2, 4, 6\}$. The missing value percentage is set to 10%, 20%, and 30% by solving the intercept term ψ_0.

In general regression analysis, the missing values on independent variables or dependent variables are not fundamentally different. So long as the missingness (of the independent or the dependent variable), does not depend on the dependent variable, dropping the incomplete observations (i.e., applying listwise deletion) will yield approximately unbiased estimates of regression coefficients (Little 1992; Schafer and Graham 2002). Therefore, the bias occurs when the missingness depends on the dependent variable, which leads to the endogenous sample selection.

When ψ_z is non-zero, the missingness mechanism becomes NMAR, which makes ML-MAR invalid. When ψ_y is non-zero, the coefficient estimation, if dropping incomplete observations, is subject to the endogenous sample selection problem. In other words, ML-MAR and listwise deletion generate approximately unbiased estimation only under the special case of $\psi_z = 0$ and $\psi_y = 0$. This observation is illustrated in Figure 2-8 below.

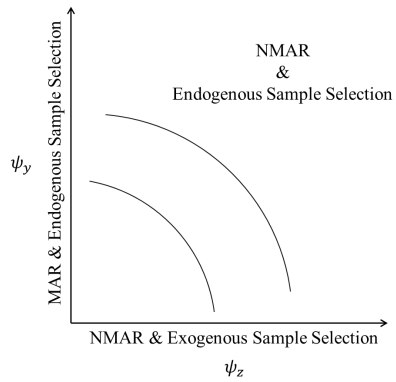


Figure 2-8 Handling Missing Values in Regression Analysis

The key point here is that, in general situations that $\psi_z \neq 0$ and $\psi_y \neq 0$, neither listwise deletion nor ML-MAR are valid approaches to handling missing

values. To handle the more general situation, we need to enhance the ML-MAR to make it robust to the NMAR mechanism, that is to employ the Monte Carlo maximum likelihood estimation.

2.5.2 Simulation Results

We present the results by listwise deletion and ML-MAR under different values of ψ_z and ψ_y in Figures 2-9 and 2-10, respectively. The spectrum from white to black represents increasing absolute bias averaged across the three regression coefficients.[21] Results are based on 400 replications of the simulation.[22]

Figure 2-9 shows that, under listwise deletion, the bias is approximately zero for ψ_y= 0 despite the value of ψ_z. This corresponds to the horizontal axis in Figure 2-8. It is also well-known that estimations of coefficients are unbiased under listwise deletion if the missingness results in exogenous sample selection. However, the potentially detrimental effects of listwise deletion are evident when ψ_y becomes non-zero. Moreover, the bias gets more severe as the missing value percentage increases. A rule of thumb is that problematic levels of bias occur when the absolute value of the bias is greater than about one half of the estimate's standard error (Schafer and Graham 2002). Given the data generating process in our simulation setting, standard errors of coefficients, β_0, β_1 and β_2 are around 0.03 if there were no missing values. Therefore, even under a moderate 10% missing value percentage, listwise deletion could result in problematic bias in coefficient estimation.

Figure 2-10 shows that, departures from MAR cause the performance of ML-MAR estimation to also be degraded. Under ML-MAR, bias is approximately zero for

[21] Results of other commonly used missing values handling methods in empirical studies (e.g., conditional mean imputation and zero/mean substation, and multiple imputation) are presented in Appendix 1.4.2. These methods generally cause bias under different values of ψ_z, ψ_y and missing value percentage.

[22] We conducted a pilot analysis to investigate the number of replications of the simulation that is necessary to obtain stable measures of bias. Results show that, the averaged performance over 200 or more rounds of simulation are stabilized. To be conservative, in our experimentation, we run 400 replications for each simulation setting.

ψ_z = 0 despite the value of ψ_y. This corresponds to the vertical axis in Figure 2-8. However, this property of ML-MAR does not hold when $\psi_z \neq 0$. In addition, when we compare Figures 2-9 and 2-10, the bias of ML-MAR estimation under the NMAR mechanism tends to be less than the bias of listwise deletion under endogenous sample selection. Although results depend on the parameter setting of the simulation, there has been much evidence that, principled methods such as maximum likelihood and multiple imputation, tend to perform better than ad hoc methods (Rubin 1996; Schafer and Olsen 1998; Schafer and Graham 2002; Buhi et al. 2008; Newman 2014).

Overall, we emphasize two considerations in determining whether missing values is problematic. The first is missing value percentage – as the proportion of missing values becomes greater, the chosen method will exert a higher degree of influence over the results, and differences among competing methods will be magnified. The second consideration is the missingness mechanism. A stronger MNAR mechanism makes the ML-MAR approach generate biased estimation; endogenous sample selection makes the listwise deletion invalid.

Figure 2-11 shows the coefficient estimation results using our Monte Carlo likelihood estimation method that incorporates the missingness mechanism. Results shows that the estimations of regression coefficients are approximately unbiased across different combinations of ψ_z, $\psi_y$and missing value percentages.

a. 10% missing b. 20% missing c. 30% missing

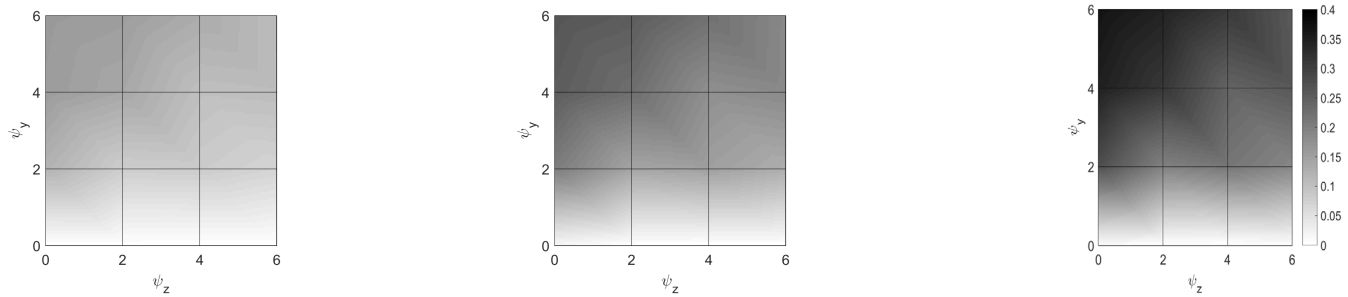


Figure 2-9 Bias of Regression Coefficients Using Listwise Deletion

a. 10% missing b. 20% missing c. 30% missing

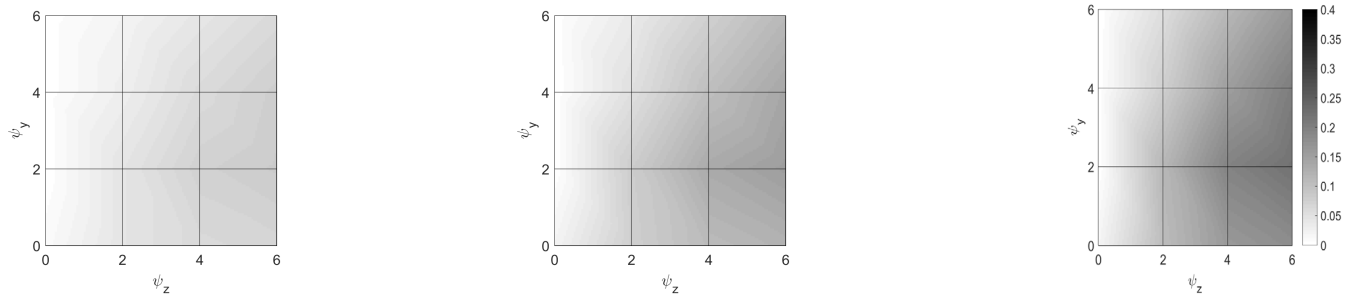


Figure 2-10 Bias of Regression Coefficients Using ML-MAR

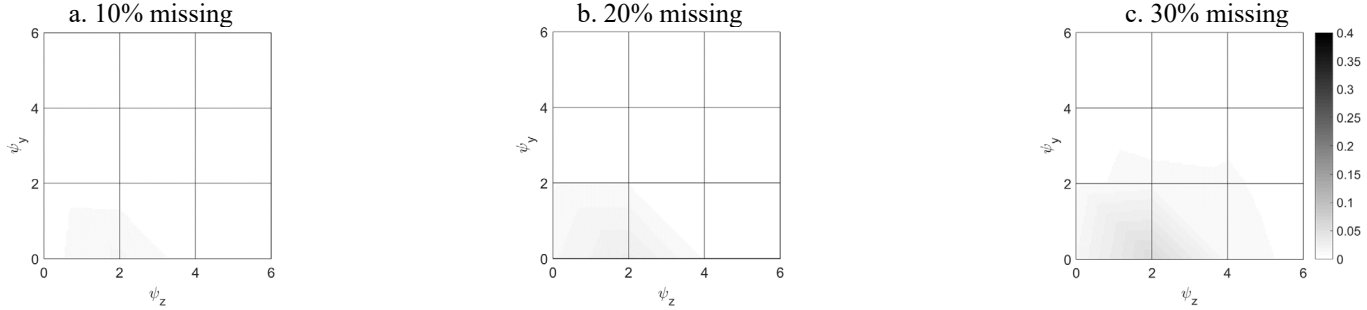


Figure 2-11 Bias of Regression Coefficients Using Monte Carlo Likelihood Estimation

		β_0				β_1				β_2			
Method	ψ_z \ ψ_y	0	2	4	6	0	2	4	6	0	2	4	6
Listwise Deletion	0	0.001	0.369***	0.437***	0.454***	0.005	-0.229***	-0.303***	-0.324***	-0.019**	0.229***	0.302***	0.323***
	2	0.001	0.366***	0.445***	0.457***	0.003	-0.226***	-0.324***	-0.346***	-0.004	0.001	0.161***	0.229***
	4	0.002	0.332***	0.432***	0.464***	0.000	-0.157***	-0.301***	-0.345***	0.000	-0.155***	0.001	0.115***
	6	0.002	0.280***	0.416***	0.451***	0.006***	-0.097***	-0.247***	-0.320***	-0.005*	-0.191***	-0.121***	0.003
ML-MAR	0	-0.001	0.000	0.002	0.000	0.020***	0.000	-0.002	0.000	-0.032***	-0.001	0.000	-0.001
	2	0.204***	0.191***	0.138***	0.097***	-0.045***	-0.082***	-0.054***	-0.037***	-0.072***	-0.070***	-0.039***	-0.026***
	4	0.286***	0.294***	0.241***	0.193***	-0.081***	-0.128***	-0.122***	-0.089***	-0.112***	-0.160***	-0.091***	-0.056***
	6	0.314***	0.335***	0.307***	0.258***	-0.084***	-0.133***	-0.149***	-0.134***	-0.136***	-0.196***	-0.152***	-0.095***
ML-Monte Carlo	0	-0.009	-0.002	-0.002	-0.001	0.010*	-0.005**	-0.003	-0.001	-0.019***	0.008***	0.004**	0.002
	2	0.103***	0.015***	0.001	-0.002	-0.018***	-0.009***	0.000	-0.001	-0.044***	-0.002	0.001	0.000
	4	0.015***	0.008***	-0.002	0.002	-0.003	-0.005**	-0.004*	0.000	-0.009***	-0.009***	0.004	0.002
	6	0.004*	0.002	0.001	0.000	0.005**	-0.003	0.000	-0.003	-0.005**	0.000	0.001	0.005**

Note: 1%, 5%, and 10% statistical significance are shaded and indicated with ***, **, and *, respectively.

Table 2-10 Estimation of Beta Coefficients (Missing Value Percentage = 30%)

Table 2-10 details the results of each of the three regression coefficients with 30% missing value percentages.[23] The values in the table are the mean bias of coefficient estimation over 400 replications of the simulation. The significance levels are for testing that the mean bias of coefficient estimation equals to zero. The results show that listwise deletion generates approximately unbiased parameter estimation when ψ_y= 0, while maximum likelihood assuming MAR (ML-MAR) is approximately unbiased when ψ_z= 0. When $\psi_z \neq 0$ or $\psi_y \neq 0$, the bias becomes significant and severe for listwise deletion and ML-MAR. With the Monte Carlo likelihood estimation, we incorporated the missingness mechanism into the likelihood model which makes it robust to the NMAR mechanism (ML-NMAR). Estimation results are approximately unbiased under different values of $\psi_y$and ψ_z (with fewer cells having significantly biased parameter estimation and overall lower magnitude in the bias[24]). Since the standard errors of the coefficients, β_0, β_1 and β_2 are around 0.03, the few biased cells are unlikely to result in problematic estimates based on the benchmark of one half of the standard error (Schafer and Graham 2002).

	Missingness Mechanism		
Missing Value Handling Methods	MCAR	MAR	NMAR
Listwise deletion	**Unbiased**	Biased	Biased*
ML ignoring missingness mechanism	**Unbiased**	**Unbiased**	Biased
ML incorporating missingness mechanism	**Unbiased**	**Unbiased**	**Unbiased**

Note: * A special case where listwise deletion generates unbiased estimation for regression coefficients under the NMAR mechanism is when ψ_y= 0 and $\psi_z \neq 0$.

Table 2-11 Comparing Frequently Used Missing Value Handling Methods

Table 2-11 summarizes the implication of missingness mechanism on the validity of different missing value handling methods based on the simulation analysis, which is largely consistent with the discussion by Newman (2014).[25] Given that $\psi_x$is

[23] Appendix 1.4.1 presents results for different missing value percentages from 10% to 40%.
[24] Although the bias is significant for certain cells of the table, the magnitude is generally around or less than 0.01, which is much less than that of listwise deletion or ML-MAR.
[25] Results of zero substitution, mean substitution, and conditional mean imputation in Appendix 1.4.2 show that the bias exists regardless the missingness mechanism. Multiple imputation behaves qualitatively the same way with maximum likelihood estimation that ignores the missingness mechanism.

fixed at zero, the MCAR mechanism corresponds to the setting where $\psi_y = \psi_z = 0$; the MAR mechanism when $\psi_y \neq 0$ and $\psi_z = 0$; and NMAR to the more general situation that $\psi_z \neq 0$. Listwise deletion is valid under the MCAR situation. Moreover, a special case for listwise deletion being unbiased under NMAR is that the missing of z only depends on z itself (i.e., $\psi_y = 0$ and $\psi_z \neq 0$). In this sense, although whether the NMAR mechanism is at play is not directly testable, the endogeneity of sample selection can be investigated by testing whether the response indicator is associated with the dependent variable (Schlomer et al. 2010). However, this is only feasible when the missing values occur in the independent variable. When the dependent variable is missing and the missingness depends on the variable itself, the problem becomes a case of the Heckman selection model as mentioned earlier. Maximum likelihood estimation ignoring the missingness mechanism eliminates the bias under the MCAR and MAR mechanisms but leads to biased estimation under the NMAR mechanism. The method which is robust to different missingness mechanisms is maximum likelihood estimation incorporating the missingness mechanism, wherein parameters are estimated using the Monte Carlo likelihood approach.

We further experimented the proposed approach in two alternative simulation scenarios to show its robustness to the mis-specification of the missingness mechanism and its generalizability. First, we let the underlying missingness mechanism to be represented with a probit model whereas specify it to be a logit model during the parameter estimation process. Second, we extend the linear regression model to a generalized linear form where the relationship between the dependent variable and independent variables is represented with a logit model. Our proposed approach generally produces unbiased coefficients estimates. Results are presented in Appendix 1.4.3. Our computational approach can be extended to handle the situation where both dependent and independent variables are subject to missing values. We present these results in Appendix 1.4.4.

2.6 Discussion

Missing values are a ubiquitous problem faced by data scientists. The statistics and data analytics literatures have attempted to tackle the missing values problem from different perspectives. Our approaches to this problem are to integrate the missingness mechanism into the traditional imputation process to improve the robustness to different missingness mechanisms including NMAR.

This proposed semi-supervised missing value imputation approach distinguishes itself from existing methods in two aspects. First, current statistics models under the broad maximum likelihood or multiple imputation approaches rarely consider the NMAR mechanism (Schafer and Graham 2002). Moreover, compared to the existing statistical methods, our non-parametric implementation does not assume joint distribution—often multivariate normal distribution—of variables in the data matrix (Allison 2009). Second, although data mining studies have explored and evaluated different machine learning algorithms for enhancing missing value imputation accuracy (e.g., Luengo et al. (2012) and Saar-Tsechansky and Provost (2007)), to the best of our knowledge, there does not exist a unique machine learning algorithm that universally outperforms all other algorithms. Moreover, the extant literature does not consider the missingness mechanism and implicitly assume MCAR or MAR. We go beyond the choice of machine learning algorithms for missing value imputation. Instead, we leverage additional information from incomplete records and further improve the imputation accuracy of traditional imputation methods using different machine learning algorithms. The traditional imputation method can be viewed as a special case of our semi-supervised imputation approach, but if the underlying missingness mechanism is NMAR, then our approach is better positioned to build a more accurate imputation model.

Our Monte Carlo based approach for handling missing values makes significant contribution to enhance the scientific validity of empirical studies. The

proposed estimation process archives robustness to NMAR, the most general missingness mechanism, by jointly modeling the distribution of the incomplete variable and the missingness mechanism. This estimation process generates approximately unbiased estimation under both MAR and NMAR mechanisms. Generating unbiased parameter is necessary and critical for both economic interpretation and statistical inference. The magnitude of coefficients (i.e., the economic significance) often have profound practical implication in empirical studies. Without incorporating the missingness mechanism or simply dropping the incomplete observations, bias could be severe.

This study contributes to the information systems literature on data quality and the capability of big data analytics. Starting from the early accounting and financial-driven databases to big data analytics nowadays, missing value has always been one of the data quality challenges that add complexity to the effective use of information systems (Ballou et al. 2003; Cappiello et al. 2003; Grover et al. 2018). We answer the call for enhancing the data quality of information systems by developing a semi-supervised missing value imputation approach.

Moreover, the problem of missing data is also critically important to the IS discipline as IS researchers are at the forefront in terms of leveraging big data from a variety of domains such as e-commerce, healthcare, among others, where IS researchers have been able to draw high-impact insights (Chen et al. 2012; Chiang et al. 2018). We contribute to research practice by proposing and demonstrating the superior performance of a Monte Carlo likelihood approach in correcting bias in parameter estimation. Our simulation study suggests that research validity can be enhanced through reasoned adoption of missing value handling method and missing value reporting practice.

Overall, the implication of our research on understanding the missing value mechanism will motivate more rigorous data collecting and analyzing process to

reduce the occurrence of missing values, or to obtain more information on the mechanisms of missing values when missing values is often inevitable. Although our research makes important contributions in multiple aspects, the limitations of this study must be acknowledged.

Although we demonstrate that, compared to traditional methods, the proposed method increases the objective likelihood function incorporating the missingness mechanism, the theoretical analysis is conducted when the benchmark imputation method is parametric. The theoretical property is not guaranteed under a nonparametric situation for our semi-parametric imputation approach. Extending the theoretical properties to nonparametric situations could be a promising direction for future work. In practical applications, we recommend using the proposed method to augment benchmark imputation methods, such as linear SVM, neural network, random forest, and gradient boosting to achieve relatively robust results, as demonstrated in the simulations and real-world experimentations.

In addition, we acknowledge the limitations of the simulation analysis where we did not exhaustively explore all possible settings. For instance, in demonstrating the bias correction in regression analysis, simulation settings would further involve varying the correlations among the variables of interest, the magnitude of the beta coefficients of the regression model, and the explanatory power (e.g., R^2) of the regression model, etc. Although the simulation setting would determine the magnitude of the bias, we expect that the results will be qualitatively similar given the theoretical properties of the missing values problem.

CHAPTER 3 TRANSFER LEARNING IN DYNAMIC BUSINESS ENVIRONMENTS: TRADE-OFFS IN RESPONSE TO CHANGES

3.1 Introduction

Advanced business analytics such as machine learning have been successfully applied in a variety of business applications such as in predicting defaults in consumer credit loans (Khandani et al. 2010), recommending products based on review ratings (Chen et al. 2012), among many others. However, applying machine learning in a real-world business context brings about many challenges. One of the challenges arises from dynamically changing data environments (Grover et al. 2018; Saboo et al. 2016; Yang and Wu 2006). In practice, empirical evidence shows that forecasting in dynamic business environments is difficult even for experts (Makridakis et al. 2009). Traditional supervised machine learning methods use historical data as a training set to construct a prediction model, and then applies the built model to current (test) data to make predictions of future events or of variables of interest. One important assumption is that the historical training data and current test data exhibit the same underlying pattern (Pan and Yang 2010). In dynamic data environments, this assumption may not always hold. For instance, in predicting firms' future earnings during recession periods, the distribution of predictors and the functional relationship between predictors and the dependent variable may change.

A simple solution to applying machine learning in such dynamic data environments is to re-train the machine learning model using re-collected current data. However, current data is often scarce, thus it could be beneficial to also leverage some aspects of the historical data in addition to the current data. This is essentially a transfer learning perspective. Transfer learning is defined as extracting knowledge from a *source* data set and applying this knowledge to a *target* task (Pan and Yang 2010). In this study, we examine the question *whether and how we can make use of all of the source data (including same-distribution recent source data and the*

remaining diff-distribution past source data)[26] *to achieve better prediction accuracy for a target task when there is only a small amount of source data that exhibit the target data pattern.*

In particular, we identify and investigate two important trade-offs faced by data analysts in dynamic data environments. First, the same-distribution source data is often scarce. Thus, the first trade-off is between two alternative strategies – 1) re-training a model using a small but more relevant same-distribution source data set or 2) using transfer learning (i.e., training a model using large but potentially less relevant data sets consisting of both same-distribution and diff-distribution source data). Moreover, since the fundamental challenge of adapting to the change originates from the scarcity of the same-distribution source data, data analysts can naturally consider waiting for a time period and to collect and incorporate more same-distribution source data to train a more accurate model for the target task, but at the cost of deteriorating prediction performance before the adjustment is made. Therefore, the second trade-off faced by data analysts is with the time dimension – whether to make the adjustment 1) immediately or 2) at a later time point when more same-distribution source data has become available.

Extant research has long been focused on detecting changes and making adjustments to the prediction model in changing data environments. Several algorithms have been proposed for monitoring errors of machine learning models, such as the drift detection method (DDM; Gama et al. 2004), adaptive windowing algorithm (ADWIN; Bifet and Gavalda 2007), among others. However, even if a change in data is detected, challenges regarding how to adapt to the changes given scarce same-distribution data still persist.

[26] The source data records used to train a machine learning model can be divided to two parts, same-distribution data and diff-distribution data. The same-distribution data exhibit the target data pattern while the diff-distribution data exhibit different pattern to the target data. In dynamic data environments, we can specify the same-distribution data as the records that are collected in the current data regime.

To overcome the problem of scarcity of same-distribution data, transfer learning generally employs a weighting scheme to jointly use the same-distribution and the diff-distribution source data sets. Intuitively, the same-distribution source data records are assigned higher weights while the diff-distribution source data records are assigned lower weights (Dai et al. 2007). There also exist a theoretically motivated instance weighting approach based on variables' distributions (Zadrozny 2004). However, these studies often focus on the situation where the distribution of predictors changes across the source data and target data while the relationship between predictors and the variable to be predicted is fixed, a situation known as *transductive* transfer learning or as the covariate drift problem (Huang et al. 2007; Zadrozny 2004). To the best of our knowledge, there exists little research on *inductive* transfer learning settings where both the distribution of predictors and the functional relationship between predictors and the dependent variable change across the source and target data.

Moreover, although different algorithms have demonstrated successful implementation in changing data environments (Ganin et al. 2016; Pan et al. 2008), we still do not have a clear understanding or explanation on when and to what extent transfer learning works. This is due to confounding factors in empirical data experimentations and the variety of design mechanisms for transfer learning algorithms. In this study, we aim to gain theoretical insights on transfer learning by proposing a transfer learning framework from the sample selection perspective and investigate the trade-offs of whether and how to conduct transfer learning through a systematic Monte Carlo study.

In our proposed framework of transfer learning, the change in the data pattern is represented by a sample selection model. The sample selection model generates the probability that a data point represents the different data pattern to the target data given its values of predictors and the value to be predicted. Assuming a model that

fits both the source data and target data exists, sample selection would result in different model estimations if the model is fitted using source data and target data separately. To adjust the prediction model trained on the source data to let it fit the target data, we derive a weighting approach based on the sample selection probability. The proposed method is theoretically driven by the empirical risk minimization (ERM) for the target data distribution, which is also reflected in the instance weighting approach by Zadrozny (2004) and Kim and Yu (2011).

We further analyze transfer learning effectiveness under the proposed framework. Based on our conceptual analysis, it is expected that the number of predictors being used, and the extent of underlying changes will have an effect on the effectiveness of transfer learning. To clearly depict the overall trade-offs, we design a simulation study to examine the effectiveness of transfer learning in changing data environments. Results are consistent with our expectations. In general, transfer learning outperforms retraining using only the same-distribution data when the same-distribution data is scarce. In addition, the benefits of transfer learning are more prominent under a larger number of predictors and when the extent of change is smaller. Regarding when the model should be adapted, our simulation results show that retraining using the same-distribution data greatly benefit from adjusting the model at a later time point until more same-distribution data can be incorporated. When the number of same-distribution data is large enough, transfer learning that leverages the diff-distribution data tends to be less accurate than retraining a model using same-distribution data.

Our study provides important theoretical and managerial insights from multiple aspects. First, this study investigates two important trade-offs in response to changes in data patterns and sheds light onto the understanding of the bias-variance and exploration-exploitation trade-offs, respectively (Shmueli and Koppius 2011). Regarding the first trade-off between applying a transfer learning strategy vs. re-

training the prediction model using only the same-distribution source data, our simulation results show that using historical source data with uniform weight introduces bias in model prediction although at the same time reduces the variance of prediction error. However, the inherent bias-variance trade-off can be alleviated by strategically leveraging the source data, i.e., applying transfer learning based on sample selection probability which corrects the model trained using source data towards the target data pattern. When the same-distribution source data is sparse, both theoretical analysis and simulation results point to the advantage of transfer learning based on sample selection probability. The exploration-exploitation trade-off is reflected in the timing of adjusting the prediction model. The tension of this trade-off differs among the two alternative strategies. The advantage of exploration of additional data samples significantly outweighs exploitation of sparse same-distribution source data in the re-training strategy, whereas this trade-off is not apparent for the transfer learning strategy. Built upon our theoretical analysis on the effectiveness of transfer learning, data analysts can incorporate their prior knowledge or conduct customized simulation analysis in deciding whether or not to use transfer learning and when to adjust the prediction model.

Second, our study provides practical managerial implications for predictive analytics in changing data environments. In predictive modeling, a large historical training data set is often viewed as beneficial to improve the reliability of prediction models and also allows data analysts to develop complex models to closely approximate reality. However, the inherent uncertainty and dynamism in the business environment prompt us to reconsider the information value of historical data records and the effectiveness of transfer learning in utilizing them. Although transfer learning would effectively improve prediction performance compared to re-training a new prediction model, as the number of same-distribution data grows, evidence from our simulations shows that simply re-training the model using same distribution data

could actually be superior. Therefore, the choice of the optimal strategic response to data pattern change depends on the feasibility of actively collecting additional data. In some domains, delaying action until additional data examples are available may result in substantial loss. In such situations, an immediate response to the change needs to be made which favors the transfer learning strategy.

Third, we contribute to the transfer learning literature by developing a theoretical framework from a sample selection perspective. Built upon empirical risk minimization, we further derive a probabilistic weighting scheme which minimizes bias caused by source data. Based on the proposed framework, we decompose the approximation error of transfer learning, which guides further insights into the effectiveness of transfer learning. Compared to extant transfer learning methods which often rely on pre-specified commonalities among different data patterns, our perspective focuses on modeling the sample selection process that distinguishes different data sets, and then correcting prediction models towards the target data patterns by adjusting the weights of source data samples. Our method minimizes heuristic pre-specifications thus potentially enhances the robustness of transfer learning in data environments without much unexpected uncertainty.

3.2 Related Works

Learning in dynamic environments is an important topic. In this section, we will review related research streams on change detection and transfer learning that contribute to this challenge. Detecting change is an important part of enhancing learning quality in dynamic environments. Transfer learning provides a direction to adjust the prediction model when the data pattern changes. By synthesizing the related research streams, we discuss the research gaps and motivate our research method.

3.2.1 Change Detection

Detecting changes in a data stream is an important area of research with many applications. A common approach to change detection is to monitor the error rate of the current prediction model, which, under a stable data pattern, should remain stabilized. The prediction model that maps the predictors to the variable to be predicted does not need to change if the underlying relationship is stable. When the error rate increases significantly, change is declared, and it is invoked that the prediction model should be revised or rebuilt with new data.

Existent approaches have been successfully implemented to detect changes in the data pattern based on error streams. These approaches generally compare two sub-sets of a data sequence; if the difference is sufficiently significant (i.e., surpasses a certain threshold), change is declared (Harel et al. 2014). This threshold is often based on a heuristic statistical model. For example, the drift detection method (DDM) by Gama et al. (2004), monitors the error rate p_i and the standard deviation s_i at instance i and registers two values p_{min} and s_{min}. Every time a new instance i is processed, p_{min} and s_{min} are updated when $p_i + s_i$ is lower than $p_{min} + s_{min}$. The condition for detecting change is $p_i + s_i \geq p_{min} + \alpha \times s_{min}$ with $\alpha = 3$ as recommended by the authors. Another popular algorithm is the adaptive windowing algorithm (ADWIN) by Bifet and Gavalda (2007). It exclusively compares two sub-windows of a data sequence; whenever the averages of two sub-windows are different (i.e., exceed a certain threshold ϵ_{cut}), it concludes that the expected values are different across the two sub-windows. The calculation of the threshold is as follows, which in general involves a user-input δ parameter as an upper bound of the false positive rate.

$$m = \frac{1}{1/n_0 + 1/n_1},\ \delta' = \frac{\delta}{n}, \epsilon_{cut} = \sqrt{\frac{1}{2m} \cdot \ln\left(\frac{4}{\delta'}\right)}.$$

where n_0 and n_1 are lengths of the two sub-windows and n is the length of the whole window ($n = n_0 + n_1$).

However, even if data change can be effectively detected, the challenge in adapting the prediction model persists due to the scarcity of new data. Unlike change detection, which is based on the prediction errors of instances, adjusting the prediction model typically involves numerous predictors along with the variable to be predicted. Hence, a large sample is generally required to retrain a reliable model.

3.2.2 Transfer Learning

Transfer learning is defined as extracting knowledge from a source data set and applying this knowledge to a target task. Using source data may be beneficial in improving prediction performance when new data or target data is scarce. In this section, we introduce concepts and developments in transfer learning. The changes in data pattern across source data and target data may involve both the distribution of predictors $\boldsymbol{x}$, namely $\Pr(\boldsymbol{x})$, as well as the relationship between $\boldsymbol{x}$ and the variable y to be predicted, represented with a probabilistic conditional distribution $\Pr(y \mid \boldsymbol{x}; \boldsymbol{\theta})$. The former is within the domain space and the latter is within the task space. Depending on how the source data and target data are different from one another, the two broad categories of transfer learning are transductive transfer learning and inductive transfer learning, as summarized in Table 3-1, where the superscript S (T) indicates that the variable or parameter is from source (target) data.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Task space: $\Pr(y \mid \boldsymbol{x}; \boldsymbol{\theta})$</td></tr>
<tr><td>$\Pr^T(y|\boldsymbol{x};\boldsymbol{\theta}) = \Pr^S(y|\boldsymbol{x};\boldsymbol{\theta})$</td><td>$\Pr^T(y|\boldsymbol{x};\boldsymbol{\theta}) \neq \Pr^S(y|\boldsymbol{x};\boldsymbol{\theta})$</td></tr>
<tr><td rowspan="2">Domain space: $\Pr(\boldsymbol{x})$</td><td>$\Pr^T(\boldsymbol{x}) = \Pr^S(\boldsymbol{x})$</td><td>Traditional Machine Learning</td><td rowspan="2">Inductive Transfer Learning</td></tr>
<tr><td>$\Pr^T(\boldsymbol{x}) \neq \Pr^S(\boldsymbol{x})$</td><td>Transductive Transfer Learning</td></tr>
</table>

Table 3-1 Categories of Transfer Learning in Supervised Machine Learning

Transductive Transfer Learning

The transductive transfer learning setting is the situation where (1) the conditional probability $\Pr(y|\boldsymbol{x};\boldsymbol{\theta})$ is fixed while the change of data environments is represented by the different distribution of $\boldsymbol{x}$ across the source and target data; and (2) the $\boldsymbol{x}$ of the target data set needs to be observed when the machine learning model is being constructed so that the information about the change of $\Pr(\boldsymbol{x})$ can be obtained. Within machine learning topics, this sub-problem is also known as covariate shift (Sugiyama et al. 2008) since predictors $\boldsymbol{x}$ are also named covariates.

With structured data, transductive transfer learning methods are dominated by instance-transfer through importance weighting. To illustrate the idea of importance weighting, we introduce the notations for source data set and target data set. Let $\boldsymbol{D_S} = \{(\boldsymbol{x}_1^S, y_1^S),\ (\boldsymbol{x}_2^S, y_2^S),\ \dots,\ (\boldsymbol{x}_m^S, y_m^S)\}$ be the source data set, and the target data set is denoted with $\boldsymbol{D_T} = \{\boldsymbol{x}_1^T,\ \boldsymbol{x}_2^T,\ \dots, \boldsymbol{x}_n^T\}$ where the corresponding $y_1^T, y_2^T, \dots, y_n^T$ are unknown and to be predicted. The sample size of source data and target data is *m* and *n*, respectively.

The estimated parameter $\boldsymbol{\theta}^*$ in $\Pr(y|\boldsymbol{x};\boldsymbol{\theta})$ should minimize the expected risk,

$$\boldsymbol{\theta}^* = \arg_{\boldsymbol{\theta}}\min \mathbf{E}_{(\boldsymbol{x},y)\sim \Pr(\boldsymbol{x},y)}[l(\boldsymbol{x}, y; \boldsymbol{\theta})], \tag{3.1}$$

where $l(\boldsymbol{x}, y; \boldsymbol{\theta})$ is a loss function. For instance, the loss function can be the negative log-likelihood $-\ln\big(\Pr(y|\boldsymbol{x};\boldsymbol{\theta})\big)$ from a maximum likelihood estimation perspective. The expected loss is calculated over the distribution of $(\boldsymbol{x}, y)$, denoted with $\Pr(\boldsymbol{x}, y)$. However, since we only observe source data examples $\boldsymbol{D_S} = \{(\boldsymbol{x}_1^S, y_1^S),\ (\boldsymbol{x}_2^S, y_2^S),\ \dots,\ (\boldsymbol{x}_m^S, y_m^S)\}$ drawn from $\Pr(\boldsymbol{x}, y)$, we have to resort to estimating parameter $\boldsymbol{\theta}^*$ through empirical risk minimization (ERM; Vapnik 1995), namely,

$$\boldsymbol{\theta}^* = \arg_{\boldsymbol{\theta}}\min \frac{1}{m}\sum_{(\boldsymbol{x},y)\in \boldsymbol{D_S}}[l(\boldsymbol{x}, y; \boldsymbol{\theta})]. \tag{3.2}$$

In general transfer learning settings (inductive or transductive), since the distribution of $(\boldsymbol{x}, y)$ is different across target and source data sets, we have $\Pr^T(\boldsymbol{x}, y) \neq \Pr^S(\boldsymbol{x}, y)$. In this case, we want to learn an optimal model for the target data set by minimizing the following expected risk:

$$\boldsymbol{\theta}^* = \arg_{\boldsymbol{\theta}}\min \mathbf{E}_{(\boldsymbol{x},y)\sim \Pr^T(\boldsymbol{x},y)}[l(\boldsymbol{x}, y; \boldsymbol{\theta})], \tag{3.3}$$

which can be transformed to $\boldsymbol{\theta}^* = \arg_{\boldsymbol{\theta}}\min \mathbf{E}_{(\boldsymbol{x},y)\sim \Pr^S(\boldsymbol{x},y)} \left[\frac{\Pr^T(\boldsymbol{x},y)}{\Pr^S(\boldsymbol{x},y)} l(\boldsymbol{x}, y; \boldsymbol{\theta})\right]$.

Therefore, the ERM to estimate $\boldsymbol{\theta}^*$ in the general transfer learning setting is:

$$\boldsymbol{\theta}^* = \arg_{\boldsymbol{\theta}}\min \frac{1}{m}\sum_{(\boldsymbol{x},y)\in \boldsymbol{D_S}} \left[\frac{\Pr^T(\boldsymbol{x},y)}{\Pr^S(\boldsymbol{x},y)} l(\boldsymbol{x}, y; \boldsymbol{\theta})\right], \tag{3.4}$$

Intuitively, Equation (3.4) indicates that, when $\Pr^T(\boldsymbol{x}, y) \neq \Pr^S(\boldsymbol{x}, y)$, we need to assign weight $\Pr^T(\boldsymbol{x}_i^S, y_i^S)/\Pr^S(\boldsymbol{x}_i^S, y_i^S)$ to the source data record $(\boldsymbol{x}_i^S, y_i^S)$, *i*=1,…, *m*.

In the transductive transfer learning setting, $\Pr(y|\boldsymbol{x}; \boldsymbol{\theta})$ is assumed to be fixed across the source and target data sets, thus we have $\Pr^T(\boldsymbol{x}_i^S, y_i^S)/\Pr^S(\boldsymbol{x}_i^S, y_i^S) = \Pr^T(\boldsymbol{x}_i^S)/\Pr^S(\boldsymbol{x}_i^S)$. Estimating $\Pr^T(\boldsymbol{x}_i^S)$ and $\Pr^S(\boldsymbol{x}_i^S)$ only for the purpose of calculating $\Pr^T(\boldsymbol{x}_i^S)/\Pr^S(\boldsymbol{x}_i^S)$ may lead to much unnecessary computational workload. For instance, given that the number of predictors is usually large, it would be expensive and even infeasible to estimate the joint distribution of all predictors (Sugiyama et al. 2008). Zadrozny (2004) propose a weighting scheme based on a sample selection model without estimating $\Pr^T(\boldsymbol{x}_i^S)$ or $\Pr^S(\boldsymbol{x}_i^S)$. The sample selection model describes the probability that a data record is "selected" to be a source data record (rather than a target data record). In their selection model, this probability is assumed to only depend on predictors $\boldsymbol{x}$ but not y. This sample selection idea is related to our proposed method. However, our method aims at solving the problem of inductive transfer learning as introduced below.

Inductive Transfer Learning

Inductive transfer learning is a more general learning setting and aims to investigate not only the possible change in $\boldsymbol{x}$, but also the change in the conditional probability $\Pr(y|\boldsymbol{x};\boldsymbol{\theta})$. However, if $\Pr^T(y|\boldsymbol{x};\boldsymbol{\theta})$ is arbitrarily different to $\Pr^S(y|\boldsymbol{x};\boldsymbol{\theta})$, there is no way we could infer a good estimator based on the source data. More specifically, in the above importance weighting scheme, we cannot even theoretically estimate the weight $\Pr^T(\boldsymbol{x}_i^S, y_i^S)/\Pr^S(\boldsymbol{x}_i^S, y_i^S)$ if $\Pr^T(y|\boldsymbol{x};\boldsymbol{\theta})$ is completely unknown in the target data. Therefore, some prior information is required to infer the change of $\Pr(y|\boldsymbol{x};\boldsymbol{\theta})$.

One type of prior information is the prior distribution of parameters. Methods employing this type of information are viewed as parameter-transfer methods since it is the parameters (prior of parameters) that are transferred across source and target data sets. For instance, Lawrence and Platt (2004) and Kumagai and Iwata (2018) assume that the $\boldsymbol{\theta}$ of $\Pr(y|\boldsymbol{x};\boldsymbol{\theta})$ follows a Gaussian process. Saboo et al. (2016) employ a time-varying effect analysis to model the regression coefficients as a smooth function of time.

Another type of prior information is the specification of some "good" source data that exhibit the same data pattern as the target data (called same-distribution source data). Methods employing this type of information are instance-transfer methods and like transductive learning, adjust the training of the machine learning model by importance weighting. The same-distribution source data can be specified by human experts based on their domain knowledge. As illustrated in Figure 3-1 below, in changing data environments, it is natural to identify data records that are collected in the current data regime as the same-distribution source data since they reflect the most up-to-date data pattern. The historical data records collected during the previous data regime are diff-distribution source data. In the simulation analysis

of Section 3.4, we detailed the specification of same-distribution and diff-distribution source data in the context of change detection in dynamic data environments.

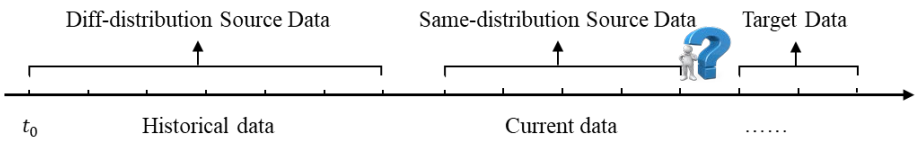


Figure 3-1 Illustration of Same-Distribution and Diff-Distribution Source Data in Changing Data Environments

Compared to parameter-transfer, instance-transfer does not require a GP-type view of the dynamic environment, but rather represents a data-driven approach which adjusts the training of the machine learning algorithm in a desirable direction implied by the same-distribution source data records. Moreover, parameter-transfer assumes pattern change of the whole data generating process over a long-time horizon while instance-transfer involves short memory and is triggered by the changes across source and target data. In the context of business cycle dynamics, policy changes and structural breaks, the short memory view may be more realistic and easily understood (Chen and Niu 2014).

However, in spite of this salient advantage, existing solutions on instance-transfer for inductive learning are rare and have non-trivial limitations. To exploit the same-distribution source data, an important approach in the literature is to assign higher weights to the same-distribution source data and lower weights to the remaining source data. However, quantifying an appropriate weighting scheme is a non-trivial issue. Jiang and Zhai (2007) discuss an approach in which human experts learn from the same-distribution source data to identify and delete the "misleading" source data records. However, this approach would require intensive human intervention thus could be practically infeasible. Dai et al. (2007) develop a TrAdaBoost classifier to exploit the same-distribution source data. TrAdaBoost adjusts the iterative process of the original AdaBoost, by increasing the weights of

same-distribution source data and decreasing the weights of the remaining diff-distribution source data (called diff-distribution source data). Although the weighting scheme of TrAdaBoost is intuitively appropriate, however, as pointed out by the authors, TrAdaBoost does not guarantee to always improve AdaBoost, since the quality of diff-distribution source data is uncertain.

In summary, the extant transfer learning literature has primarily been method oriented and less focused on the question of when and to what extent transfer learning works well in the broader context of changing data environments. Theoretical guidance in conducting inductive transfer learning is also in dire need.

3.3 Developing a Transfer Learning Framework for Dynamic Data Environments

To account for changing data patterns, transfer learning typically requires strategically weighting the diff-distribution source data. In this section, we first formalize the transfer learning problem as a sample selection setting and propose a probabilistic weighting method. Then, we explore factors that would influence the effectiveness of transfer learning compared to only using the same-distribution source data.

3.3.1 Transfer Learning from a Sample Selection Perspective

Transfer learning can be conceptualized as sample selection bias (Heckman 1979). Transductive transfer learning (a.k.a. covariate shift) has been explored empirically from a sample selection perspective (e.g., Bickel et al. 2007; Zadrozny 2004). Here, we develop a more general framework for the transfer learning setting where the change across data pattern may involve both predictors and the functional relationship between predictors and the variable to be predicted.

From the perspective of sample selection, an underlying sample selection model can be used to represent the probability that a data point represents a different

data pattern from the target data given its values of predictors and the value to be predicted. The probability is realized in a way that we observed some data points in the source data set and other data points in the target data set. The sample selection model is widely used in improving causality identification (Heckman 1979). Although data analytics for prediction do not aim at establishing causality (Shmueli and Koppius 2011), the sample selection perspective has been preliminarily adopted in transductive transfer learning studies (e.g., Zadrozny 2004) and helps to derive a more robust model when there is potential heterogeneity in different data sets.

Let r=1 if the data record follows different distribution to the target data (i.e., diff-distribution source data records), and r=0 if the data record follows the same distribution to the target data (i.e., same-distribution source data records). The proposed model involves approximating $\Pr(r=1 \mid \boldsymbol{x}, y)$, the probability that a data point exhibits the source data pattern (or equivalently in approximating $\Pr(r=0 \mid \boldsymbol{x}, y)$, the probability that a data point exhibits the target data pattern).[27] It can be shown through the following derivation that, if the underlying probability $\Pr(r=1 \mid \boldsymbol{x}, y)$ of the source data is known, the target data pattern can be estimated through empirical risk minimization (ERM) combining same-distribution source data with weighted diff-distribution source data. More specifically, the challenge is to obtain an estimate for $\boldsymbol{\theta}$ that minimizes the expected loss in target data, namely $E(l(\boldsymbol{x}, y; \boldsymbol{\theta}) | r = 0)$. It can be shown that the weights of diff-distribution source data records can be theoretically derived for minimizing the risk of target data.

Theorem 3.1 Let $\Pr(r = 0|\boldsymbol{x}, y)$ be the probability that the data record $(\boldsymbol{x}, y)$ does not follow the target data pattern, wherein $r = 1$ indicates that the data record follows a data pattern different from the target data; and $r = 0$ if the data record follows the

[27] When the selection process does not depend on y, namely $\Pr(r=1 \mid \boldsymbol{x}, y) = \Pr(r=1 \mid \boldsymbol{x})$, it is a typical scenario of exogenous selection which generally does not result in a change in the relationship between $\boldsymbol{x}$ and y in regression analysis. This exogenous selection process is investigated in transductive transfer learning setting (Zadrozny, 2004).

target data pattern. Then we have

$$E(l(\boldsymbol{x}, y; \boldsymbol{\theta})|r = 0) = E(w(\boldsymbol{x}, y)l(\boldsymbol{x}, y; \boldsymbol{\theta})|\, r = 1), \tag{3.5}$$

wherein $w(\boldsymbol{x}, y)$ is defined as the following:

$$w(\boldsymbol{x}, y) = \frac{\Pr(r=0|\boldsymbol{x},y)}{\Pr(r=1|\boldsymbol{x},y)} \frac{\Pr(r=1)}{\Pr(r=0)}, \tag{3.6}$$

$w(\boldsymbol{x}, y)$ indicates the odds that the data record exhibiting the target data pattern given the value $(\boldsymbol{x}, y)$ multiplying a constant $c = \frac{\Pr(r=1)}{\Pr(r=0)}$ which is the overall odds of observing diff-distribution data.

Proof sketch:

$$\begin{aligned}
E(l(\boldsymbol{x}, y; \boldsymbol{\theta})|r = 0) &= \int l(\boldsymbol{x}, y; \boldsymbol{\theta}) \Pr(\boldsymbol{x}, y|r = 0) dxdy && \text{(i)} \\
&= \int l(\boldsymbol{x}, y; \boldsymbol{\theta}) \frac{\Pr(\boldsymbol{x},y,r=0)}{\Pr(r=0)} dxdy && \text{(ii)} \\
&= \int l(\boldsymbol{x}, y; \boldsymbol{\theta}) \frac{\Pr(\boldsymbol{x},y,r=1)}{\Pr(r=1)} \frac{\Pr(r=1)}{\Pr(\boldsymbol{x},y,r=1)} \frac{\Pr(\boldsymbol{x},y,r=0)}{\Pr(r=0)} dxdy && \text{(iii)} \\
&= \int l(\boldsymbol{x}, y; \boldsymbol{\theta}) \frac{\Pr(\boldsymbol{x},y,r=0)}{\Pr(\boldsymbol{x},y,r=1)} \frac{\Pr(r=1)}{\Pr(r=0)} \frac{\Pr(\boldsymbol{x},y,r=1)}{\Pr(r=1)} dxdy && \text{(iv)} \\
&= \int l(\boldsymbol{x}, y; \boldsymbol{\theta}) \frac{\Pr(\boldsymbol{x},y,r=0)}{\Pr(\boldsymbol{x},y,r=1)} \frac{\Pr(r=1)}{\Pr(r=0)} \Pr(\boldsymbol{x}, y|r = 1) dxdy && \text{(v)} \\
&= \int w(\boldsymbol{x}, y) l(\boldsymbol{x}, y; \boldsymbol{\theta}) \Pr(\boldsymbol{x}, y|r = 1) dxdy && \text{(vi)} \\
&= E(w(\boldsymbol{x}, y) l(\boldsymbol{x}, y; \boldsymbol{\theta})|r = 1) && \text{(vii)}
\end{aligned}$$

wherein the equality (ii) above is because the conditional probability $\Pr(\boldsymbol{x}, y|r) = Pr(\boldsymbol{x}, y, r)/Pr(r)$, and the equality (iii) above is obtained by multiplying $1 = Pr(\boldsymbol{x}, y, r = 1)/Pr(r = 1) \times Pr(r = 1)/Pr(\boldsymbol{x}, y, r = 1)$. The weight $w(\boldsymbol{x}, y) = \frac{\Pr(\boldsymbol{x},y,r=0)}{\Pr(\boldsymbol{x},y,r=1)} \frac{Pr(r=1)}{Pr(r=0)} = \frac{\Pr(r=0|\boldsymbol{x},y)}{\Pr(r=1|\boldsymbol{x},y)} \frac{\Pr(r=1)}{\Pr(r=0)}$.

Q.E.D.

Equation (3.5) suggests that, to estimate $\boldsymbol{\theta}$ in $E(l(\boldsymbol{x}, y; \boldsymbol{\theta})|r = 0)$, we can alternatively minimize the expected loss $E(w(\boldsymbol{x}, y)l(\boldsymbol{x}, y; \boldsymbol{\theta})|r = 1)$. This line of probabilistic weighting approach is also employed for handling non-ignorable missing values (Kim and Yu 2011). The left-hand side expectation in Equation (3.5) can be approximated by averaging the loss of data points that follow the target data

pattern (conditional on r=0), while the right-hand side expectation can be approximated by averaging the weighted loss of data points following different data pattern to target data (conditional on r=1). Since same-distribution source data is often scarce, it would be beneficial if we can leverage a relatively large number of diff-distribution source data and minimize the empirical loss for the right-hand side expectation. Table 3-2 presents the algorithm of the proposed transfer learning method based on sample selection.

Algorithm 3.1: Transfer learning based on sample selection

Data:

Source data set $\boldsymbol{D_S} = \{(\boldsymbol{x}_1^S, y_1^S),\ (\boldsymbol{x}_2^S, y_2^S),\ \dots,\ (\boldsymbol{x}_m^S, y_m^S)\}$. $\boldsymbol{D_S}$ is splitted to two components of size p and q, $m = p + q$, respectively:

// Diff-distribution source data: $\boldsymbol{D_{S-D}} = \{(\boldsymbol{x}_1^{S-D}, y_1^{S-D}),\ \dots,\ (\boldsymbol{x}_p^{S-D}, y_p^{S-D})\}$

// Same-distribution source data: $\boldsymbol{D_{S-S}} = \{(\boldsymbol{x}_1^{S-S}, y_1^{S-S}),\ \dots,\ (\boldsymbol{x}_q^{S-S}, y_q^{S-S})\}$

$\boldsymbol{r}$, selection indicator vector of length m

// $r_i = 1$ if the i-th source data record is a diff-distribution source record, $r_i = 0$ otherwise, $i = 1, 2, \dots, m$.

Target data set $\boldsymbol{D_T} = \{\boldsymbol{x}_1^T,\ \boldsymbol{x}_2^T,\ \dots,\ \boldsymbol{x}_n^T\}$.

Input:

ω, a list of weights for source data records

PreMdl, user specified prediction model of the relationship between y and $\boldsymbol{x}$

SelMdl, user specified classification model to classify the same-distribution and the diff-distribution source data sets

Output:

$\boldsymbol{y}^T$, prediction outcome for target data records

1 train classification model *SelMdl* to classify same-distribution and diff-distribution source data;
2 **for** each i in $\{1, 2, \dots, p\}$
3 calculate the weight of the diff-distribution source data record $(\boldsymbol{x}_i^{S-D}, y_i^{S-D})$ using Equation (3.6), and store the corresponding weight to the list ω;
4 **end for**
5 **for** each i in $\{1, 2, \dots, q\}$
6 assign weight one to the same-distribution source data record $(\boldsymbol{x}_i^{S-S}, y_i^{S-S})$, and store the corresponding weight to the list ω;
7 **end for**
8 train prediction model *PreMdl* using the same-distribution and diff-distribution source data with weights in ω;
9 apply *PreMdl* to the target data $\boldsymbol{D_T}$ and generate prediction $\boldsymbol{y}^T$;
10 **return** $\boldsymbol{y}^T$

Table 3-2 Algorithm of Transfer Learning Based on Sample Selection

In practice, after splitting the overall source data to diff-distribution and same-distribution components, with the sample size being p and q, respectively, the empirical loss for estimating $\boldsymbol{\theta}$ consists of two terms:

$$L_{transfer} = \frac{1}{p+q}\{\sum_{i=1}^{p}[w(\boldsymbol{x}_i^{S-D}, y_i^{S-D})l(\boldsymbol{x}_i^{S-D}, y_i^{S-D}; \boldsymbol{\theta})] + \sum_{i=1}^{q}[l(\boldsymbol{x}_i^{S-S}, y_i^{S-S}; \boldsymbol{\theta})]\}, \quad (3.7)$$

where the first term corresponds to the empirical weighted loss of diff-distribution source data and the second term corresponds to the empirical loss of same-distribution source data. Up to now, we assumed that the selection probability $\Pr(r = 1|\boldsymbol{x}, y)$ is known. In practice, we would have to estimate it from the data. Using the diff-distribution source data labelled with r=1 and same-distribution source data labelled with r=0, we can train a classifier that generates the 0/1 label and the estimated probability $\widehat{\Pr}(r = 1|\boldsymbol{x}, y)$.

3.3.2 The Effectiveness of Transfer Learning

In the transductive transfer learning paradigm (which models the change in the distribution of predictors $\boldsymbol{x}$), theoretical analysis of error bounds shows that the distribution divergence of predictors' distribution between same-distribution and diff-distribution source data and the complexity of function classes measured with Vapnik–Chervonenkis (VC) dimension are important factors determining whether we should use the diff-distribution source data (Ben-David et al. 2010). However, the understanding of the inductive transfer learning paradigm (where the relationship between $\boldsymbol{x}$ and y changes) is still in a vacuum. In this section, we discuss two factors that would potentially influence the effectiveness of transfer learning – 1) the number of predictors used in the prediction model and 2) extent of change across the source and the target data sets.

To proceed with our analysis, we leverage the Hoeffding's inequality, a commonly used inequality in statistical learning theory. The Hoeffding's inequality gives Lemma 3.1 and 3.2 as follows.

Lemma 3.1 The empirical average loss of same-distribution source data examples is expressed as $\frac{1}{q}\sum_{i=1}^{q}\left[l\left(\boldsymbol{x}_i^{S-S}, y_i^{S-S}; \boldsymbol{\theta}\right)\right]$, denoted with L_{S-S}. Let the loss of any data example be upper bounded: $l(\boldsymbol{x}, y; \boldsymbol{\theta}) \leq b$. By Hoeffding's inequality, we have, for any $\delta > 0$:

$$\Pr\left(|L_{S-S} - E^{r=0}| \geq \frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)}\right) \leq \delta,$$

wherein b is the upper bound of the loss of data examples, q is the sample size of same-distribution source data.

Lemma 3.2 The empirical average weighted loss of diff-distribution source data examples is expressed as $\frac{1}{p}\sum_{i=1}^{p}\left[w_i l\left(\boldsymbol{x}_i^{D-S}, y_i^{D-S}; \boldsymbol{\theta}\right)\right]$, denoted with L_{D-S}. Let the weighted loss of any data example be upper bounded: $wl(\boldsymbol{x}, y; \boldsymbol{\theta}) \leq a$. By Hoeffding's inequality, we have, for any $\delta > 0$:

$$\Pr\left(|L_{D-S} - E^{r=0}| \geq \frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)}\right) \leq \delta,$$

wherein a is the upper bound of the weighted loss of data examples, and p is the sample size of diff-distribution source data.

In Lemma 3.1, we have that, with probability at least δ, the distance between empirical average loss of same-distribution data examples and $E^{r=0}$ is bounded with $\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)}$. Analogously, Lemma 3.2 shows that, with probability at least δ, the distance between empirical average weighted loss of diff-distribution data examples and $E^{r=0}$ is bounded with $\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)}$. Under the transfer learning strategy, both the same-distribution and the diff-distribution data records are used. Based on Lemma 3.1

and Lemma 3.2, we obtain probabilistic bound for the distance between L_T and $E^{r=0}$, as presented in Theorem 3.2.

Theorem 3.2 According to Lemma 3.1 and Lemma 3.2, for any $\delta > 0$, we have:

$$\Pr\left(|L_T - E^{r=0}| \geq \frac{p}{p+q}\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)} + \frac{q}{p+q}\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}\right) \leq \delta.$$

According to Theorem 3.2, with probability at least $1-\delta$, the distance between the empirical loss using transfer learning and the expected loss of target data is bounded with $\zeta \equiv \frac{p}{p+q}\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)} + \frac{q}{p+q}\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}$. Proof of Theorem 3.2 is presented in Appendix 2.1.

The results we obtain so far allow us to analyse the potential benefit of transfer learning. In particular, we denote the difference between the two probabilistic upper bounds for the distance of transfer learning/non-transfer learning to $E^{r=0}$ as Δd, namely $\Delta d = \frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)} - \left(\frac{p}{p+q}\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)} + \frac{q}{p+q}\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}\right)$. By rearranging the terms, we have:

$$\Delta d = \frac{b}{\sqrt{q}}\left[\frac{p}{p+q}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}\left(1 - \frac{a\sqrt{q}}{b\sqrt{p}}\right) - \left(\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)} - \sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)}\right)\right].$$

Assuming that we have a sufficiently large number of diff-distribution source data (i.e., $p \gg q$ with a and b being fixed), then we have $\frac{p}{p+q} \approx 1$. Moreover, for a fixed confidence level δ, $\sqrt{\frac{1}{2}\ln\left(\frac{2}{\delta}\right)}$ and $\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}$ are constants, denoted with c_1 and c_2, respectively. Therefore, we approximate Δd as follows:

$$\Delta d \approx \frac{b}{\sqrt{q}}\left[c_1 - \frac{a\sqrt{q}}{b\sqrt{p}}c_2\right].$$

From our analysis, we gain theoretical insights on whether and to what extent transfer learning is more beneficial than re-training a prediction model. First, the amount of same-distribution source data influences transfer learning effectiveness. When we have a sufficiently large number of diff-distribution source data (i.e., $p \gg q$

with a and b being fixed), we have $1 - \frac{a\sqrt{q}}{b\sqrt{p}} > 0$, meaning that transfer learning reduces the error bound. In response to detected changes, the relative magnitude between p and q varies depending on the timing of when the prediction model is adjusted (i.e., re-trained). As the number of same-distribution source data points increases, transfer learning would retain less of an advantage.

Second, the term $\frac{b}{\sqrt{q}}$ indicates the upper bound of approximation error when only using same-distribution data. Given that transfer learning is beneficial in reducing the error bound, we expect the effectiveness of transfer learning to be more significant when the approximation error of only using the same-distribution target data is large. In a linear regression setting, given a fixed number of observations, increasing the number of predictors would result in less reliable estimation which would make it more beneficial to conduct transfer learning. The expected influence of the number of predictors on transfer learning effectiveness is consistent with the conclusion drawn by Ben-David et al. (2010) that increasing complexity of function classes measured with the VC dimension would favor using diff-distribution source data.

Third, the effectiveness of transfer learning depends on the relative magnitude of a and b which in turn depends on the underlying data pattern change. When the data pattern remains unchanged, all the data points are selected to source or target data completely by random; thus, $w_i = 1$ and $a = b$; whereas when there is a systematic sample selection process, we will observe a variation of w_i which will likely increase the upper bound a.[28] Although the underlying true weight w_i is unknown, one observation following this analysis is that the advantage of transfer

[28] A more nuanced reasoning is that, when all the data points are equally selected to source or target data, we have $\Pr(r = 0|\boldsymbol{x}, y) = \Pr(r = 1|\boldsymbol{x}, y) = 0.5$, meaning that the formation of source data and target data is as random as a coin toss. Then according to equation (3.6), the weight of any source data point w_i becomes a constant. Without loss of generality, we assume that the constant equals to one thus $a = b$. When different data points are systematically selected with the probability $\Pr(r = 0|\boldsymbol{x}, y)$ varying from zero to one, we need a greater upper bound of w_i.

learning over re-training tends to diminish when there exists changes in the data patterns. Intuitively, when the data pattern undergoes a large change, using the diff-distribution data could be sub-optimal (Ben-David et al., 2010). In practice, the extent of changes is difficult to measure ex-anti since same-distribution source data is scarce, thus it would be difficult to use the extent of change as a factor to judge transfer learning effectiveness. That said, in our subsequent simulation analysis, we vary the underlying sample selection mechanism to gain theoretical insights.

Although it is difficult to exactly quantify the approximation error of prediction models without restrictive assumptions such as variables' distribution, the nature of the underlying changes, and the specification of prediction models, our theoretical analysis suggests several directions for better appreciating the effectiveness of transfer learning. In the following section, we examine the trade-offs inherent in responding to changes in a change detection setting.

3.4 Data Experimentations

When data pattern changes, data analysts face the trade-off of whether to apply transfer learning in the first place. Moreover, they also face a trade-off with the time dimension – whether to make an adjustment immediately or at a later time point (to incorporate more same-distribution source data to train a more accurate model for the target task). Finally, implementing transfer learning requires splitting the source data to the same-distribution and diff-distribution components a-priori. In practical applications, data analysts have to distinguish them based on observed information and their own judgement.

To depict a full picture on these practical issues, we design a simulation study to systematically evaluate the effectiveness of transfer learning. The changing data pattern occurs under the sample selection framework and the trade-offs regarding transfer learning are triggered by the detected change in data environments. We

investigate the trade-offs under different simulation settings by systematically varying the number of predictors being used and extents of change in the data pattern.

3.4.1 Simulation of Changing Data Patterns

The prediction model we focus on is a linear regression model. As discussed previously, the complexity of a linear regression model is largely determined by the number of predictors – as the number of predictors increases, a growing number of observations is required to derive a reliable prediction model. As such, transfer learning would be particularly beneficial to high-dimensionality prediction problems.

In our simulation study, we use a linear model of the form $y = \boldsymbol{x} \times \boldsymbol{\beta} + \varepsilon$ with the number of predictors being k, where k takes value from $\{10, 20, \dots, 50, 60\}$. The predictors $\boldsymbol{x} = (x_1, x_2, \dots, x_k)$ are taken from a *k*-dimensional normal distribution as follows:

$$\boldsymbol{x} \sim \mathcal{N}_k(\boldsymbol{\mu}, \Sigma),$$

with mean $\boldsymbol{\mu}$ being a zero vector and elements (σ_{ij}) of the covariance matrix being 0.5 for $i \neq j$ and 1 for $i = j$. Coefficient parameters $\beta_1, \beta_2, \dots, \beta_k$ are set as ones. The error term ε is drawn from a normal distribution. As the number of predictors increases, the variance of $\boldsymbol{x} \times \boldsymbol{\beta}$ increases. Since the number of predictors ranges from 10 to 60, to fix the model fit (R^2) at a moderate level, we set the variance of the error term as follows:

$$Var(\boldsymbol{x} \times \boldsymbol{\beta}) \times \frac{1 - R^2}{R^2}.$$

Furthermore, we let R^2 to be 0.6. Results are broadly similar when R^2 equals 0.2 or 0.8. We use a sample selection model to separate the simulated data to two data sets that exhibit different relationships between predictors $\boldsymbol{x}$ and the outcome variable y. The model we employ is a probit model as follows:

$$\Pr(r = 0 | \boldsymbol{x}, y) = \Phi\left(\psi_y y + \boldsymbol{\psi}_{\boldsymbol{x}} \boldsymbol{x}\right).$$

$\Pr(r = 0|\boldsymbol{x}, y)$ of an observation $(\boldsymbol{x}, y)$ is the probability that it exhibits the different data pattern as the target data. Separation of the simulated data is realized through a Bernoulli trial. To change the relationship between $\boldsymbol{x}$ and y across the two data sets ($r = 0$ versus $r = 1$), it is important to set the coefficient parameter ψ_y to be non-zero. Otherwise, it becomes the well-known exogenous sample selection case and does not result in different coefficient parameters that capture the relationship between $\boldsymbol{x}$ and y. We vary the magnitude of ψ_y to vary the extent of changes. In particular, to let the variation of ψ_y successfully controls the extent of changes, we set $\boldsymbol{\psi_x}$ as an all-ones vector and let ψ_y take value from {0.3, 0.5, …, 1.5}. Figure 3-2 shows that the extent of change can indeed be manipulated by varying ψ_y. The change across two data sets are measured with the Euclidean distance between the two vectors of coefficient parameters. For different number of predictors in use, increasing ψ_y leads to increased Euclidean distances between coefficient vectors of the two data sets.

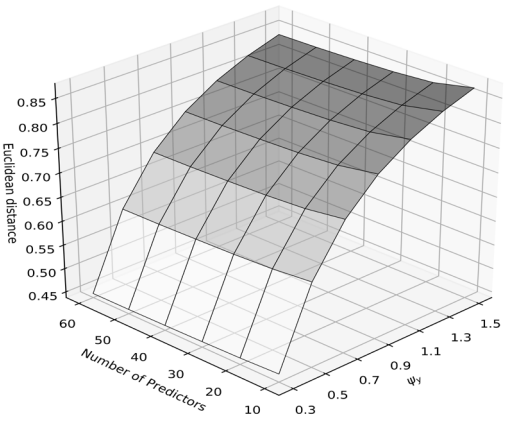


Figure 3-2 Simulation of Changing Data Patterns through Sample Selection

3.4.2 Detecting Changes in Data Environments

When data patterns undergo changes, it becomes necessary to adjust the prediction model to avoid deteriorating prediction accuracy. Transfer learning combines the same-distribution data and weighted diff-distribution data to generate an adjusted

prediction model. The point in time where the change takes place can then be the cutoff point between same-distribution and diff-distribution data sets.

In practical applications, however, it is true that a data analyst would not have the foresight on future changes. We argue then that data analysts should monitor the performance of the prediction model and signals of deteriorating prediction accuracy such that it may be necessary to adjust the model. In the data mining literature, several algorithms have been proposed for monitoring errors of machine learning models, such as the adaptive windowing algorithm (ADWIN; Bifet and Gavalda 2007) among others.

In Figure 3-3, we illustrate the change detection process. Suppose a change takes place at t^*_{change} but the time of actual change point is unbeknownst to the data analyst. An error monitoring process would be able to detect the presence of the change at time t_{detect} and estimate that the prediction model most likely started to produce less accurate predictions since time $\hat{t}_{change}$. Given the inferred changing point $\hat{t}_{change}$, data records that appear after it can be viewed as *same-distribution source data* while the ones before that point can be viewed as *diff-distribution source data*. Diff-distribution source data consists mainly of historical source data that is less relevant, while same-distribution source data consists mainly of timely-relevant data to the target data pattern.

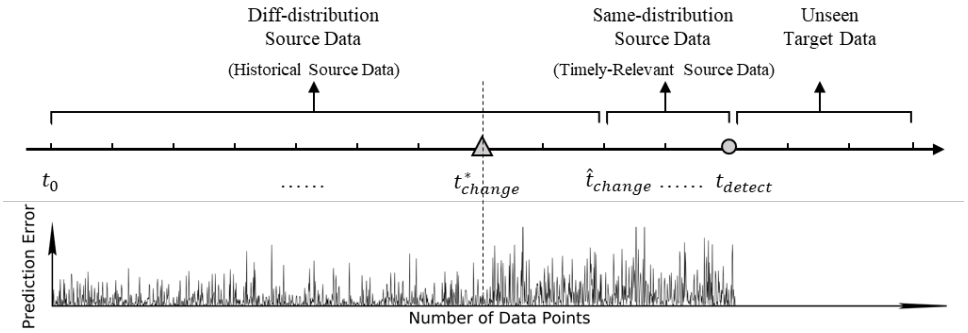


Figure 3-3 Change Detection in Dynamic Data Environments

In our simulations, the information for detecting changes is the out-of-sample prediction error of a prediction model trained using 10,000 source data records. These 10,000 source data records can be viewed as historical data within which no significant change of data pattern takes place. The out-of-sample prediction is first conducted on 1,000 data points following source data pattern and then 1,000 data points following target data pattern. Therefore, the underlying change of data pattern takes place at time t=1,000 if we view the first data point for conducting out-of-sample prediction as t=0. Note that here t does not indicate the time instance but the order of observing data points over time. In practical application, data can be collected at irregular time intervals.

Change detection is conducted based on the sequence of out-of-sample prediction error. In particular, for every record of prediction error, a test of change is conducted using the ADWIN algorithm (Bifet and Gavalda 2007).[29] The idea of ADWIN is intuitive as introduced earlier. It requires a user-input δ parameter as an upper bound of the false positive rate. During our experimentations, the change detection time point generally varies a little given different δ within the interval (0, 1). This is consistent with the observations of Ang et al. (2012). We set the confidence value to be 0.3 and results are qualitatively similar under different values of δ at 0.1 or 0.9.

In Figure 3-4, we show (a) the delay in detecting the change, calculated as $t_{detect} - t^*_{change}$, and (b) the number of same-distribution source data used when the change is detected, calculated as $t_{detect} - \hat{t}_{change}$, under different simulation settings of extent of change (ψ_y) and number of predictors. A larger extent of change can be more promptly detected. However, we also note that after the change is successfully detected, challenges will still remain. The size of same-distribution source data

[29] We use the adwin function provided by scikit-multiflow package: https://scikit-multiflow.github.io/scikit-multiflow/skmultiflow.drift_detection.adwin.html.

(generally 20~50 data points) is often insufficient to re-train the prediction model for the target data.

(a) Delay in Detection (b) Same-distribution Source Data

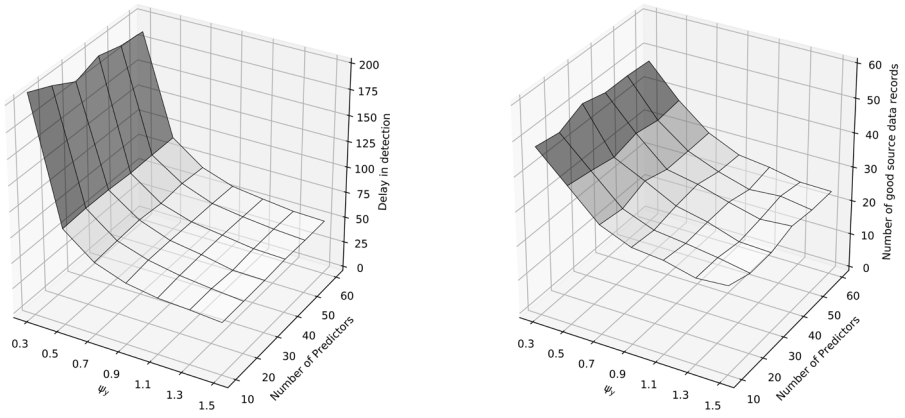


Figure 3-4 Change Detection Results

3.4.3 Trade-offs in Response to Changes

Due to the scarcity of the same-distribution source data, we face a trade-off between transfer learning using both same-distribution and diff-distribution source data, and retraining a model only using the same-distribution source data set – i.e., the trade-off between bias and efficiency. The out-of-sample prediction error of a linear model depends on both the bias and efficiency in parameter estimation (e.g., for mean squared error, its expectation can be decomposed to the bias and variance components: $E\left(y - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\right)^2 = \left[E\left(y - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\right)\right]^2 + Var\left(y - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\right)$. Using only the same-distribution source data set favors the minimization of the bias but will result in larger variance in parameter estimation.

In addition, another trade-off emerges simultaneously when the change is detected – i.e., when the prediction model should be retrained. It is within expectation that making the adjustment at a later point in time allows the data analyst to incorporate a larger same-distribution source data set to generate a more accurate model for the target task. However, what is unknown is what is the extent of the

benefit and whether this benefit outweighs the cost of deteriorating prediction performance before the retraining is conducted. Moreover, this question needs to be answered for different response strategies (i.e., transfer learning vs. simply retraining using same-distribution source data).

3.5 Results

3.5.1 The Trade-off on Whether and How to Implement Transfer Learning

When the same distribution source data is scarce, transfer learning augments the (re)training dataset by leveraging the diff-distribution data. Furthermore, regarding how to conduct transfer learning, there are alternative approaches. First, a naïve approach is to expand the sample size for training a prediction model by uniformly weighting all source data examples. This approach incorporates relevant information from same-distribution data to a limited extent as the training sample is dominated by diff-distribution data. Second, the proposed probabilistic weighting approach is shown to generate unbiased estimation of the expected loss. Hence the second transfer learning approach is to train a prediction model using uniformly weighted (assigned with a weight of ones) same-distribution data and probabilistic weighted diff-distribution data. To obtain weights according to Equation (3.6), we need to estimate for each diff-distribution data record the probability that it exhibits different data pattern from the target data. This probability can be estimated with a probit model. The dependent variable is the label for the source data (ones for diff-distribution source data and zeros for same-distribution source data) and the explanatory variables include all the predictors x and the variable to be predicted y. Third, based on the estimated probability that a source data record exhibits the same or different data pattern to the target data, one can choose to remove the diff-distribution data that are unlikely to exhibit the target data pattern and train a model using uniformly weighted same-distribution data and remaining diff-distribution data.

In our subsequent analyses, we will present the trade-off between only using the same-distribution source data (named as 'Dropping' for short) and three alternative transfer learning methods: 1) equally weighting all source data, named as 'Transfer - equal weight', 2) combining same-distribution data (assigned with weight ones) and probabilistic weighted diff-distribution data, named as 'Transfer - weighting', and 3) combining same-distribution data with filtered diff-distribution data (all assigned with weight ones), named as 'Transfer - filtering'.

As shown in Figure 3-4, when the change is detected, the number of same-distribution source data can be even less than the number of predictors, making the prediction model unidentifiable. In such cases, we evaluate the performance of difference response methods when the number of same-distribution source data is at least the number of predictors, which is consistent with the experimentation of Ben-David et al. (2010).[30] The outcomes of interest are mean squared error (MSE), squared mean bias ($Bias^2$), and variance of the error (Variance). Since the variance of the variable to be predicted increases with the number of predictors being used, all the errors are normalized by the variance of the variable to be predicted. Results presented are based on four hundred replications of the simulation.

Figure 3-5 shows the pairwise comparisons among the four methods based on MSE of 1,000 data points following the target data pattern (Appendix 2.2 provides tabulated results of Figure 3-5). The *z*-axis indicates the difference of MSE between a pair of methods (the MSE of the former method minus the MSE of the latter method). Figure 3-5a shows the relative performance between 'Dropping' and 'Transfer-equal weight'. These two approaches can be viewed as two extremes in their response to

[30] When the number of same-distribution source data is equal to the number of predictors, the linear model under 'Dropping' is just identifiable, but the prediction performance fluctuates largely. We monitor the performance of 'Dropping' until it achieves a relatively stable prediction performance - that is the number of same-distribution source data being the number of predictors plus ten. Results are qualitatively similar using slightly increasing number of same-distribution source data (e.g., 1.1 times of the number of predictors plus ten) and are presented in Appendix 2.2.

changes. ‘Dropping’ implicitly assumes that the previously collected diff-distribution data is completely irrelevant thus the benefit of using transfer learning (reduced variance) does not compensate the large bias caused by the diff-distribution data. ‘Transfer-equal weight’ implicitly assumes a persistent data pattern overtime thus the same-distribution and diff-distribution data are assigned with uniform (i.e., equal) weights. Results show that in changing data environments, ‘Dropping’ results in lower MSE than ‘Transfer-equal weight’ (cf., see negative difference of MSE between ‘Dropping’ and ‘Transfer-equal weight’) when there is a relatively large extent of change and a smaller number of predictors – the MSE difference is 1.49 under ψ_y=0.3 with 60 predictors, and -0.60 under ψ_y=1.5 with 10 predictors.

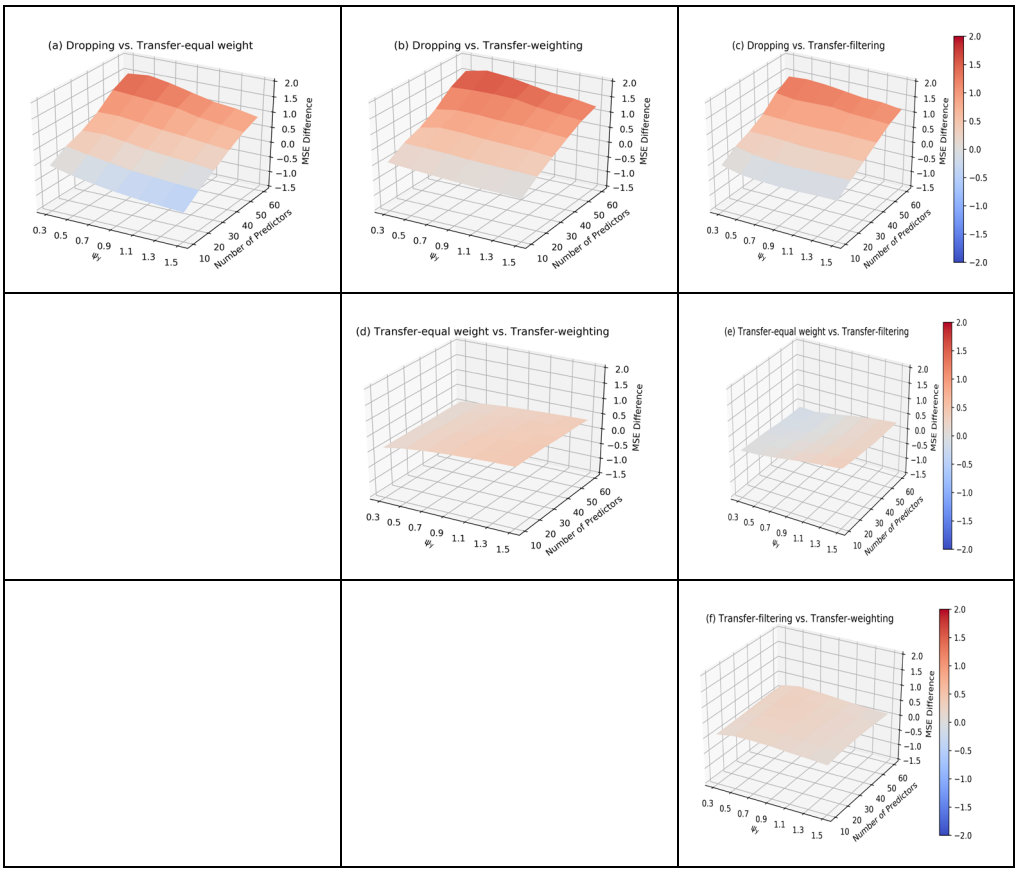


Figure 3-5 Pairwise Comparison of the Four Methods in Response to Changes

This is consistent with our expectation regarding the bias-variance decomposition, as shown in Figure 3-6 (tabulated results are provided in Appendix 2.3). Figures 3-6a and 3-6b show the difference in squared bias and variance,

respectively, between ‘Dropping’ and ‘Transfer-equal weight’ approaches. Using the diff-distribution data results in greater bias compared to ‘Dropping’ (cf., see negative values in the difference of Bias^2 between ‘Dropping’ and ‘Transfer-equal weight’ in Figure 3-6a) and the increase in bias is more prominent when the extent of change increases. With respect to variance, ‘Dropping’ increases error variance (cf., positive values in the difference of variance between ‘Dropping’ and ‘Transfer-equal weight’ in Figure 3-6b) and the increase in variance is exacerbated when the number of predictors increases.

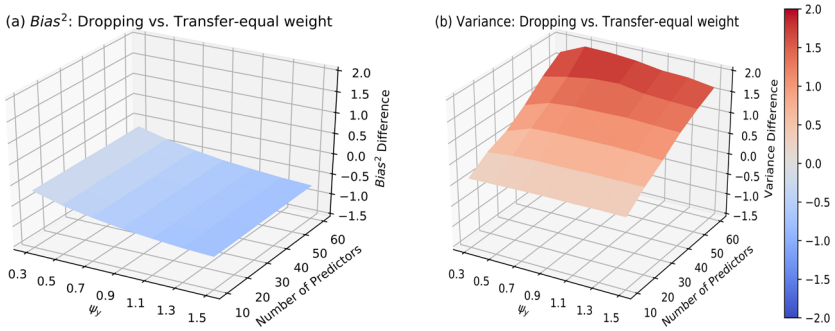


Figure 3-6 Bias-Variance Trade-off in Responding to Changes

Among the transfer learning approaches (see Figures 3-5d, 3-5e, 3-5f), ‘Transfer-weighting’ outperforms both ‘Transfer-equal weight’ and ‘Transfer-filtering’. Between ‘Transfer-weighting’ and ‘Transfer-equal weight’ (see Figure 3-5d), the probabilistic weighting approach adjusts the prediction model towards the target data pattern which alleviates the bias caused by diff-distribution source data. Between ‘Transfer-weighting’ and ‘Transfer- filtering’ (see Figure 3-5f), the probability weighting provides more information than filtering the diff-distribution data (which can be viewed as discretizing the real-value weighting to a 0/1 weighting scheme).

Finally, ‘Transfer-weighting’ outperforms ‘Dropping’ (see Figure 3-5b). The MSE difference between ‘Dropping’ and ‘Transfer-weighting’ increases when there is a relatively smaller extent of change and a larger number of predictors – the MSE difference is 1.59 under ψ_y=0.3 with 60 predictors, and -0.16 under ψ_y=1.5 with 10

predictors. This is again consistent with our expectation. As the number of predictors increases, more observations are required to generate a reliable prediction model, which disadvantages the 'Dropping' approach. Under a relatively small change, the data pattern of the diff-distribution data is relatively close to that of the target data thus it is more beneficial to leverage on these data points. Moreover, even if the extent of change increases, 'Transfer-weighting' still achieves better performance than 'Dropping'. This is because the probability weighting approach generates an unbiased estimate to the expected loss in the target data, as discussed in Equation (3.5).

The detection of change may be triggered not only due to the dynamic relationship between predictors and the outcome variable, but also due to the decreased predictability of the outcome variable. In our robustness tests, we fix the relationship between predictors and the outcome variable but double the variance of the error term in the prediction model for the target data points. Under this scenario, an optimal way of developing the prediction model in response to "changes" is to use the source data set with uniform weights across all examples. Results show that 'Transfer-weighting' obtains similar results as 'Transfer-equal weight' and the difference in MSE between the two methods is less than 0.02 under different number of predictors. 'Dropping' results in a much greater MSE than the two transfer learning approaches.

Overall, at the time when change is detected, the strategic transfer learning based on sample selection probability generates higher prediction accuracy compared to discarding the diff-distribution source data or equally weighting all the source data records.

3.5.2 The Trade-off on When to Retrain the Prediction Model

In the previous subsection, the trade-off faced by data analysts when a change is detected is due to the scarcity of same-distribution data. Under such a situation, data

analysts could consider waiting for some time to allow for more same-distribution source data to be collected for use in re-training a new model for future target data. An increasing number of same-distribution source data will generate more accurate predictions, for both transfer learning methods and for 'Dropping'. However, a notable downside is the inflated prediction error in the data before the model is re-trained. Therefore, it is important to examine to what extent the benefits of incorporating more same-distribution data outweighs the opportunity costs arising from later adjustment.

Here, we focus on comparing 'Dropping' and 'Transfer-weighting'. We choose 'Transfer-weighting' out of the three transfer learning methods since it predominantly outperforms the other two under most simulation settings. Table 3-3 illustrates the trade-off with the time dimension under 2×2 combinations of simulation settings – i.e., the number of predictors being 20 and 50, and $\psi_y$being 0.5 and 1.3. Results are based on four hundred replications of simulations. Other simulation settings with different number of predictors and values of ψ_y exhibit qualitatively similar patterns.

Among the figures presented in Table 3-3, the *x*-axis, ranging from 0 to 1000, indicates the number of data points being evaluated) since the actual change in the data pattern. The *y*-axis indicates the accumulated squared error (ASE) of the data points being evaluated. The trade-off in the time dimension is illustrated by the ASE of three lines (solid blue line, orange dashed line, and green dotted line) that use different number of same-distribution source data records (1/3/5 times the number of predictors) to adjust the prediction model. For the lines depicted in each figure, the slope represents the model prediction accuracy MSE, thus their turning points happen at around 1/3/5 times the number of predictors.[31] The first column shows the results when the prediction model is adjusted by 'Transfer-weighting', the second column

[31] The turning points are not exactly located at 1/3/5 times of the number of predictors due to the delay in change detection.

shows results of ‘Dropping’, and the third column shows the difference in ASE between ‘Dropping’ and ‘Transfer-weighting’.

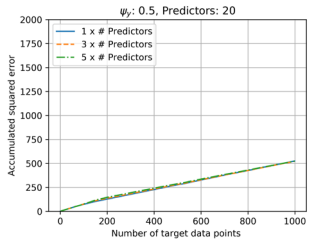
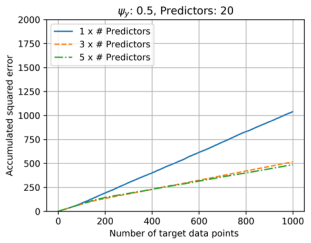
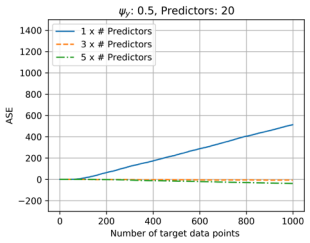
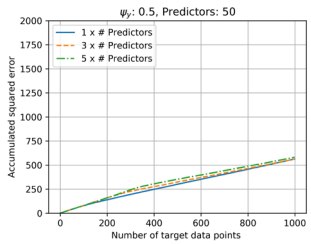
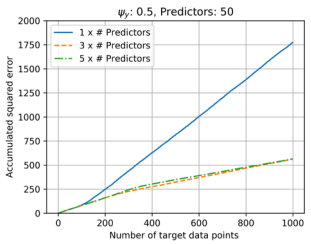
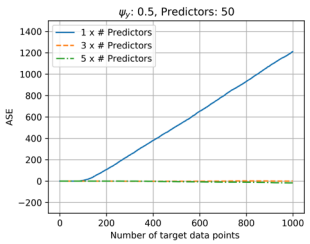
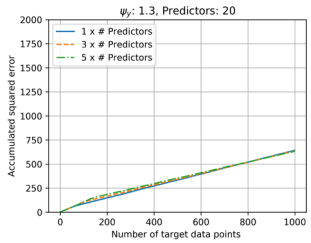
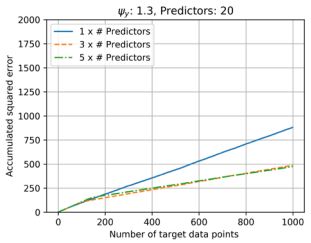
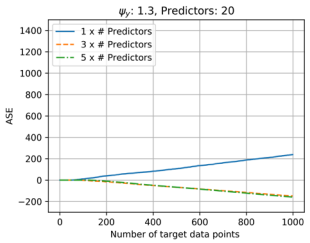
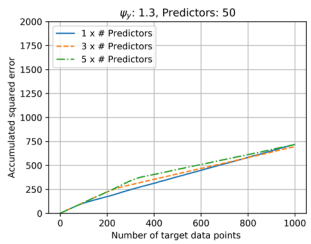
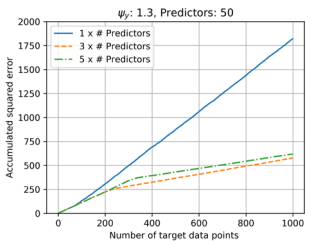
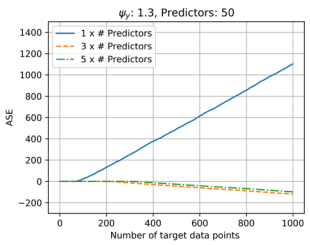
Transfer-weighting	Dropping	Dropping vs. Transfer-Weighting
		
		
		
		

Table 3-3 Illustration of the Trade-off of When to Adjust the Prediction Model

The performance of ‘Transfer-weighting’ is relatively insensitive to the number of same-distribution data points being used, especially for low-dimensionality prediction problems (i.e., 20 predictors). For instance, comparing the situation of 20 predictors versus that of 50 predictors under ψ_y=0.5, the slop of the lines after adjusting the prediction model are closer to each other for 20 predictors than for 50 predictors. A similar pattern can be found by comparing 20 predictors versus 50

predictors under ψ_y=1.3. Therefore, waiting for more same-distribution data to incorporate in the re-training is more beneficial for higher dimension prediction problems. However, the benefit is minimal for 'Transfer-weighting', since we observe relatively stable performance across different number of same-distribution data.

'Dropping' is much more sensitive compared to 'Transfer-weighting' to the number of same-distribution data points being used. This can be shown by the largely flattened ASE slope comparing 3 times and 1 time the number of predictors, especially for high-dimensionality prediction problems (i.e., 50 predictors). For the 'Dropping' method, the ASE slopes under 3 times and 5 times the number of predictors are close to each other, which is due to decreasing marginal benefits of same-distribution source data points in improving the model's prediction performance.

Comparing 'Transfer-weighting' and 'Dropping' in the third column, we observe an obvious benefit of transfer learning in adjusting the model right at the time that changes are detected, as shown by the positive slopes of the blue lines (i.e., compared to 'Transfer-weighting', 'Dropping' has greater accumulated error). However, if data analysts decide to adjust the prediction model after more same-distribution data are collected, 'Dropping' generally outperforms transfer learning, as shown by the negatively sloped orange dashed and green dotted lines. Intuitively, with the availability of more same-distribution data, the data scarcity is no longer a problem and there is less reason to use diff-distribution data. In addition, the noise in specifying the diff-distribution data would influence the accuracy of estimating the selection probability.

To investigate the general trends of prediction performance at different timing of adjusting the prediction model, we gradually evaluate the prediction performance since the detection of changes at the interval of 0.1 times the number of

predictors. In Figure 3-7, we present the prediction performance MAE over the 1,000 target data points when the prediction model is adjusted at 1, 1.1, 1.2, …, 5.8, 5.9, 6 times the number of predictors. Results of 2×2 combinations of simulation settings are presented – i.e., the number of predictors being 20 and 50, and ψ_y being 0.5 and 1.3. Full results for 6×7 simulation settings are presented in Appendix 2.4. It is shown that in changing data environments, 'Transfer-equal weight' is always inferior to both 'Dropping' and 'Transfer-weighting'. When we compare 'Dropping' with 'Transfer-weighting', at the early stages after change detection, 'Transfer-weighting' largely outperforms 'Dropping'. However, as the number of predictors increases, the two approaches tend to converge, but 'Dropping' achieves slightly lower MSE compared to 'Transfer-weighting' when the extent of change is larger (ψ_y=1.3). Finally, at the setting of ψ_y=1.3 and 50 predictors, we observe a slight increasing trend of MSE for 'Dropping' as the timing of adjusting the prediction model is further delayed. This is driven by the fact that the improvement in prediction accuracy for subsequent target data points does not compensate the inflated prediction error of target data points before prediction model is adjusted.

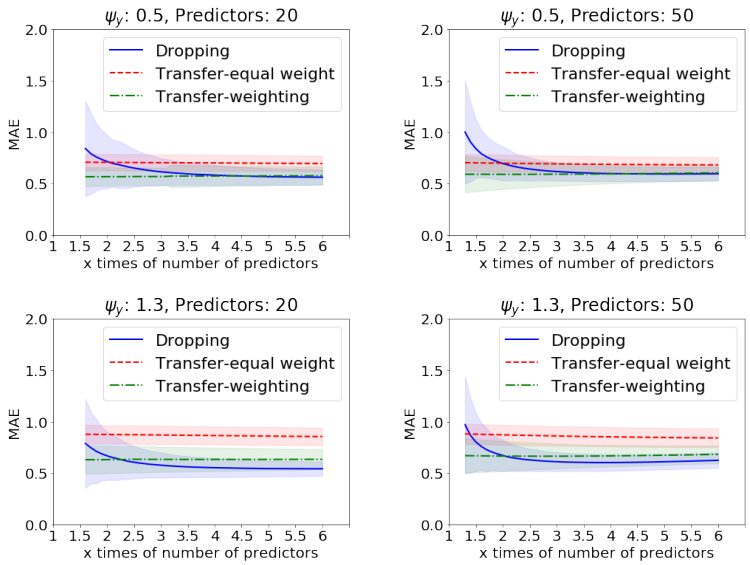


Figure 3-7 Prediction Performance and the Timing of Adjusting the Prediction Model

Overall, since the two decisions of using which strategy to adapt to the change and when the model should be adapted have to be made simultaneously, our

simulation results show that if the data analysts are able to collect more same-distribution data, 'Dropping' and re-training the prediction model at a later time point would be a simpler and superior way of responding to the changes. However, sometimes the target data do not come in a sequential pattern but are instead in bulk. In such a case, a response to the change needs to be made immediately which will favor the transfer learning strategy.

3.6 Conclusions

Improving decision outcomes in dynamic data environments is a common goal in many decision-making settings (Meyer et al. 2014). In dynamic data environments, the traditional supervised learning approach to assisting decision making may results in suboptimal outcomes since the historical pattern captured by the machine learning model may not repeat in the current period. To be well prepared for underlying changes in data environments, consistently monitoring the prediction performance and re-considering the fitness of the prediction model are necessary. However, the challenge is how to adjust the prediction model given very little information that represents the target data pattern.

In this study, we investigate transfer learning which leverages and balances the use of the same-distribution source data and diff-distribution source data jointly. Transfer learning algorithms have been widely used to make predictions in changing data environments (Ganin et al. 2016; Pan et al. 2008). However, due to the existence of confounding factors in the empirical data experimentations and varying design mechanisms of transfer learning algorithms, we lack a clear understanding of when and to what extent transfer learning works.

Motivated by the research gap in inductive transfer learning and the lack of theoretical guidelines, we first propose a transfer learning framework from the sample selection perspective. Compared to existing heuristic instance-transfer methods, the proposed method is built upon ERM theory with the objective of minimizing the

expected prediction error of target data using weighted source data. Moreover, we conduct a conceptual analysis on the effectiveness of transfer learning by decomposing the approximation error and derive insights on the effectiveness of transfer learning. This relates to the practical trade-off of whether we should be using transfer learning or should simply retrain a model using the same distribution data in the first place. Moreover, due to the scarcity of same-distribution data, another trade-off we identify is whether we should retrain the prediction model immediately when the change is detected or at a later time point when more same-distribution data has become available to train a more accurate prediction model for the target data. We conduct simulation analyses to investigate the overall trade-offs in the context of a dynamic data environment where these two trade-offs are invoked by the change detected.

Overall, our study contributes to the transfer learning literature by developing a theoretical framework for understanding transfer learning from a sample selection perspective. Our simulation results offer a comprehensive depiction on the two trade-offs and provide practical implications for data analytics in dynamic data environments. Transfer learning is generally robust to dynamic data environments and also overcomes the scarcity problem in adjusting the prediction model. However, as the number of same-distribution data grows, retraining the model is more efficient than transfer learning. Therefore, the choice of the optimal strategic response to data pattern changes depends on the practical routine of data collection and data analysis. If the data analyst can expand the same-distribution data set, simply retraining the prediction model would be superior to transfer learning. However, sometimes the delay in adjusting the prediction model may result in substantial loss, or the collection of additional same-distribution data could be costly. In such situations, an immediate response to the change need to be made which supports the transfer learning strategy.

CHAPTER 4 CONCLUSIONS

Data analytics has become a popular and promising research area that attracts substantive attention from practitioners and academia. However, to make successful applications of data analytics in real-world practice, we need to understand the theory and assumptions of appropriate applications. As discussed in the Introduction Chapter, one important fundamental challenge of applying data analytics is the heterogeneity in data patterns. The first study investigates the heterogeneity between observed values and unobserved values, while the second study investigates the heterogeneity between source data (e.g., historical data) and target data (e.g., recent data generated in the new data regime). For both problems, augmenting or identifying the information that explains how the data patterns differ to each other, and making proper use of this information are critical issues.

In the first essay, built upon the missing value literature, I employ the missing value mechanism as a device to represent systematic difference between observed and unobserved values. Under the NMAR mechanism where a variable's missingness depends directly on its value, ignoring the missingness mechanism often results in invalid statistical estimations. Motivated with research gaps in the literature, I focus on missing values imputation under the general NMAR mechanism. We use a classification model, such as logistic regression or machine learning model to represent the missing value mechanism, which is in turn incorporated in the process of imputing missing values. Since missing values are unknown in the first place, we do not have sufficient information to directly estimate the unknown parameters in the missingness mechanism. In the proposed semi-supervised missing value imputation approach, to augment the missing information, I estimate the imputation model and missing value mechanism with a semi-supervised learning approach by randomly drawing missing values based on their conditional distribution. In the proposed Monto Carlo based maximum likelihood estimation approach, I correct the bias in

coefficient estimation by jointly estimating the relationship between variables and the missingness mechanism.

In the second essay, I focus on the problem of statistical learning in dynamic data environments. Traditional machine learning uses historical data as the source data to obtain a prediction model, and then apply the model to predict future information. When the data pattern undergoes a significant change, for instance, with different interdependencies between predictors and the variable to be predicted, an inductive transfer learning problem, prediction performance may drop dramatically. A simple solution to implement machine learning in such dynamic data environments is to re-train a machine learning model using re-collected current data. However, current data is often scarce, thus it would be optimal to leverage both historical data and current data. To obtain a theoretical understanding on how to conduct transfer learning, I propose to approximate the underlying changes by distinguishing the same-distribution and diff-distribution data sets through a sample selection model, which guides the training of machine learning algorithms using same- and diff-distribution data sets in a proper direction to fit the target data pattern.

Moreover, a challenge in solving the problems examined in both studies lies in the scarcity of usable information to identify the heterogeneity in data patterns. The missing values are almost unknown without an additional follow up data collection process. However, uncovering missing values by interventions would be expensive or even infeasible, such as in the healthcare practice where medical resources are very limited (Zhang et al. 2005). Therefore, analysts need to consider cost and benefit in augmenting usable information. A tradeoff on exploration and exploitation is examined in the second essay through simulation analysis. In dynamic environments, this tradeoff happens at the time dimension – when faced with changes, whether we should adjust the model based on scarce data exhibiting the target pattern, or waiting for a certain period and to incorporate more same-distribution source data to train

more accurate model for the target task.

In summary, theoretical understanding of how to develop models that are resilient to heterogeneity in complex data environment has important implications in promoting appropriate usage of data analytics in today's complex data environment. The proposed methods in this dissertation have practical implications in handling missing values and developing prediction models that are robust to changes. The illustration of tradeoff between enhancing the model performance by learning from more information and elapsing opportunities of correcting decision outcomes provides insights on developing strategic response to changes.

Built upon the dissertation work, there are several promising future research directions. For the missing value study, another important future work is to evaluate different missing value handling methods with respect to the unbiasedness of parameter estimates in empirical IS research questions. In survey-based empirical research, respondents may refuse to answer certain fields of the questionnaire (Sivo et al. 2006). In empirical studies on the interaction effect between research and development (R&D) expenses and IT investment, researchers have to deal with the large number of missing values of R&D (Banker et al. 2011; Bardhan et al. 2013; Gomez et al. 2017; Havakhor et al. 2019; Kleis et al. 2012; Mithas et al. 2017; Ravichandran et al. 2017). Firms may not disclose R&D expenses to avoid releasing proprietary information (Koh and Reeb 2015). Whether missing values in these studies significantly bias the estimates of regression coefficients raise grave concerns among empirical researchers. This direction would make significant contribution to enhance the scientific validity of empirical studies.

In dynamic data environment, making response to the changes not only involves mechanical applications of prediction models, but also the input of human domain knowledge. Human intelligence is characterized by the ability to learn and adapt efficiently to new environments (Collins 2019). Domain knowledge from

human experts would play an important role in supplementing scarce information in the new environment (Liao and Ji 2009). For instance, to detect changes, the threshold of significant change is determined by a human expert who may hold aggressive or conservative opinions in claiming the change. Thus, the overall performance of prediction model considering the response under possible false alarm will be determined by the validity of domain knowledge. In the simulation analysis examining the exploration and exploitation tradeoff, the exploitation side did not account for additional domain knowledge on change from experts. As data analytics is being integrated into the business world, human knowledge would interact with the design of algorithms more frequently. In future, I intend to examine how domain knowledge, which is often external above the prediction model, can be incorporated in the construction of prediction models and complement the models' performance in changing data environments.

REFERENCES

Abdou, H. A., and Pointon, J. 2011. "Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature," *Intelligent Systems in Accounting, Finance and Management* (18:2-3), pp. 59-88.

Allison, P. D. 2009. "Missing Data," in: *The SAGE Handbook of Quantitative Methods in Psychology,* R.E. Millsap and A. Maydeu-Olivares (eds.). London: SAGE Publications, pp. 73-90.

Ang, H. H., Gopalkrishnan, V., Zliobaite, I., Pechenizkiy, M., and Hoi, S. C. 2012. "Predictive Handling of Asynchronous Concept Drifts in Distributed Environments," *IEEE Transactions on Knowledge and Data Engineering* (25:10), pp. 2343-2355.

Athey, S., Tibshirani, J., and Wager, S. 2019. "Generalized Random Forests," *Annals of Statistics* (47:2), pp. 1148-1178.

Baird, A., Davidson, E., and Mathiassen, L. 2017. "Reflective Technology Assimilation: Facilitating Electronic Health Record Assimilation in Small Physician Practices," *Journal of Management Information Systems* (34:3), pp. 664-694.

Balcan, M.-F., and Blum, A. 2010. "An Augmented Pac Model for Semi-Supervised Learning," in *Semi-Supervised Learning,* O. Chapelle, B. Schlkopf and A. Zien (eds.). Cambridge, MA: MIT Press, pp. 397-419.

Ballou, D., Madnick, S., and Wang, R. 2003. "Special Section: Assuring Information Quality," *Journal of Management Information Systems* (20:3), pp. 9-11.

Banholzer, N., Feuerriegel, S., and Tschernutter, D. 2018. "The Misty Crystal Ball: Efficient Concealment of Privacy-Sensitive Attributes in Predictive Analytics," in: *Proceedings of the 13th Pre-ICIS Workshop on Information Security and Privacy*. San Francisco, CA, Paper 38.

Banker, R. D., Wattal, S., and Plehn-Dujowich, J. M. 2011. "R&D Versus Acquisitions: Role of Diversification in the Choice of Innovation Strategy by Information Technology Firms," *Journal of Management Information Systems* (28:2), pp. 109-144.

Bardhan, I., Krishnan, V., and Lin, S. 2013. "Business Value of Information Technology: Testing the Interaction Effect of IT and R&D on Tobin's Q," *Information Systems Research* (24:4), pp. 1147-1161.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. 2010. "A Theory of Learning from Different Domains," *Machine Learning* (79:1-2), pp. 151-175.

Bickel, S., Brückner, M., and Scheffer, T. 2007. "Discriminative Learning for Differing Training and Test Distributions," in: *Proceedings of the 24th International Conference on Machine Learning,* Z. Ghahramani (ed.), Corvallis, OR: ACM, pp. 81-88.

Bifet, A., and Gavalda, R. 2007. "Learning from Time-Changing Data with Adaptive Windowing," in: *Proceedings of the 2007 SIAM International Conference on Data Mining,* C. Apte, D. Skillicorn, B. Liu and S. Parthasarathy (eds.), Minneapolis, MN: SIAM, pp. 443-448.

Booth, J.G., and Hobert, J.P. 1999. "Maximizing Generalized Linear Mixed Model Likelihoods with an Automated Monte Carlo EM Algorithm," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61(1): 265-285.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. 1984. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks.

Cameron, A. C., Cameron, A. C., and Trivedi, P. K. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

Cappiello, C., Francalanci, C., and Pernici, B. 2003. "Time-Related Factors of Data Quality in Multichannel Information Systems," *Journal of Management Information Systems* (20:3), pp. 71-92.

Chen, H., Chiang, R. H., and Storey, V. C. 2012. "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Quarterly* (36:4), pp. 1165-1188.

Chen, T., and Guestrin, C. 2016. "XGBoost: A Scalable Tree Boosting System," in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA: ACM, pp. 785-794.

Chen, Y., and Niu, L. 2014. "Adaptive Dynamic Nelson–Siegel Term Structure Model with Applications," *Journal of Econometrics* (180:1), pp. 98-115.

Chiang, R. H. L., Grover, V., Liang, T. P., and Zhang, D. S. 2018. "Special Issue: Strategic Value of Big Data and Business Analytics," *Journal of Management Information Systems* (35:2), pp. 383-387.

Clark, J., and Provost, F. 2016. "Matrix-Factorization-Based Dimensionality Reduction in the Predictive Modeling Process: A Design Science Perspective (September 2016)," *NYU Working Paper* (No. CBA-16-01. Available at SSRN: https://ssrn.com/abstract=2845851).

Collins, A. G. E. 2019. "Reinforcement Learning: Bringing Together Computation and Cognition," *Current Opinion in Behavioral Sciences* (29), pp. 63-68.

Crammer, K., Kearns, M., and Wortman, J. 2008. "Learning from Multiple Sources," *Journal of Machine Learning Research* (9:Aug), pp. 1757-1774.

Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. 2007. "Boosting for Transfer Learning," in: *Proceedings of the 24th International Conference on Machine Learning,* Z. Ghahramani (ed.), Corvallis, OR: ACM, pp. 193-200.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (39:1), pp. 1-38.

Diether, K. B., Malloy, C. J., and Scherbina, A. 2002. "Differences of Opinion and the Cross Section of Stock Returns," *Journal of Finance* (57:5), pp. 2113-2141.

Ding, Y., and Simonoff, J. S. 2010. "An Investigation of Missing Data Methods for Classification Trees Applied to Binary Response Data," *Journal of Machine Learning Research* (11:Jan), pp. 131-170.

Farhangfar, A., Kurgan, L., and Dy, J. 2008. "Impact of Imputation of Missing Values on Classification Error for Discrete Data," *Pattern Recognition* (41:12), pp. 3692-3705.

Friedman, J. H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics* (29:5), pp. 1189-1232.

Galimard, J.-E., Chevret, S., Curis, E., and Resche-Rigon, M. 2018. "Heckman Imputation Models for Binary or Continuous MNAR Outcomes and Mar Predictors," *BMC medical research methodology* (18:1), pp. 90-13.

Galimard, J. E., Chevret, S., Protopopescu, C., and Resche-Rigon, M. 2016. "A Multiple Imputation Approach for MNAR Mechanisms Compatible with Heckman's Model," *Statistics in Medicine* (35:17), pp. 2907-2920.

Gama, J., Medas, P., Castillo, G., and Rodrigues, P. 2004. "Learning with Drift Detection," *Brazilian Symposium on Artificial Intelligence,* A.L.C. Bazzan and S. Labidi (eds.), Berlin Heidelberg, Germany: Springer-Verlag, pp. 286-295.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. 2016. "Domain-Adversarial Training of Neural Networks," *Journal of Machine Learning Research* (17:1), pp. 2096-2130.

García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. 2010. "Pattern Classification with Missing Data: A Review," *Neural Computing and Applications* (19:2), pp. 263-282.

Glynn, R. J., Laird, N. M., and Rubin, D. B. 1993. "Multiple Imputation in Mixture Models for Nonignorable Nonresponse with Follow-Ups," *Journal of the American Statistical Association* (88:423), pp. 984-993.

Gomez, J., Salazar, I., and Vargas, P. 2017. "Does Information Technology Improve Open Innovation Performance? An Examination of Manufacturers in Spain," *Information Systems Research* (28:3), pp. 661-675.

Grover, V., Chiang, R. H., Liang, T.-P., and Zhang, D. 2018. "Creating Strategic Business Value from Big Data Analytics: A Research Framework," *Journal of Management Information Systems* (35:2), pp. 388-423.

Hall, B. L., Hirbe, M., Yan, Y., Khuri, S. F., Henderson, W. G., and Hamilton, B. H. 2007. "Thyroid and Parathyroid Operations in Veterans Affairs and Selected University Medical Centers: Results of the Patient Safety in Surgery Study," *Journal of the American College of Surgeons* (204:6), pp. 1222-1234.

Harel, M., Mannor, S., El-Yaniv, R., and Crammer, K. 2014. "Concept Drift Detection through Resampling," in: *Proceedings of the 31st International Conference on Machine Learning,* E.P. Xing and T. Jebara (eds.), Beijing, China: ACM, pp. 1009-1017.

Hartley, H. O. 1958. "Maximum Likelihood Estimation from Incomplete Data," *Biometrics* (14:2), pp. 174-194.

Havakhor, T., Sabherwal, R., Steelman, Z. R., and Sabherwal, S. 2019. "Relationships between Information Technology and Other Investments: A Contingent Interaction Model," *Information Systems Research* (30:1), pp. 291-305.

Heckman, J. J. 1979. "Sample Selection Bias as a Specification Error," *Econometrica* (47:1), pp. 153–161.

Hosmer Jr, D. W. 1973. "A Comparison of Iterative Maximum Likelihood Estimates of the Parameters of a Mixture of Two Normal Distributions under Three Different Types of Sample," *Biometrics* (29:4), pp. 761-770.

Hou, K., Van Dijk, M. A., and Zhang, Y. 2012. "The Implied Cost of Capital: A New Approach," *Journal of Accounting and Economics* (53:3), pp. 504-526.

Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B., and Smola, A. J. 2007. "Correcting Sample Selection Bias by Unlabeled Data," *Advances in Neural Information Processing Systems,* J.C. Platt, D. Koller, Y. Singer and S.T. Roweis (eds.), Vancouver, BC, Canada, pp. 601-608.

Ibrahim, J. G., Lipsitz, S. R., and Chen, M. H. 1999. "Missing Covariates in Generalized Linear Models When the Missing Data Mechanism is Non-Ignorable," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (61:1), pp. 173-190.

Jiang, J., and Zhai, C. 2007. "Instance Weighting for Domain Adaptation in NLP," in: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics,* A. Zaenen and A.v.d. Bosch (eds.), Prague, Czech Republic, pp. 264-271.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. 2017. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Advances in Neural Information Processing Systems*, Long Beach, CA, pp. 3146-3154.

Khandani, A. E., Kim, A. J., and Lo, A. W. 2010. "Consumer Credit-Risk Models via Machine-Learning Algorithms," *Journal of Banking & Finance* (34:11), pp. 2767-2787.

Kim, J. K., and Yu, C. L. 2011. "A Semiparametric Estimation of Mean Functionals with Nonignorable Missing Data," *Journal of the American Statistical Association* (106:493), pp. 157-165.

King, G., and Zeng, L. 2001. "Logistic Regression in Rare Events Data," *Political Analysis* (9:2), pp. 137-163.

Kleis, L., Chwelos, P., Ramirez, R. V., and Cockburn, I. 2012. "Information Technology and Intangible Output: The Impact of IT Investment on Innovation Productivity," *Information Systems Research* (23:1), pp. 42-59.

Koh, P.-S., and Reeb, D. M. 2015. "Missing R&D," *Journal of Accounting and Economics* (60:1), pp. 73-94.

Kossinets, G. 2006. "Effects of Missing Data in Social Networks," *Social Networks* (28:3), pp. 247-268.

Kumagai, A., and Iwata, T. 2018. "Learning Dynamics of Decision Boundaries without Additional Labeled Data," in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* Y. Guo and F. Farooq (eds.), London, United Kingdom: ACM, pp. 1627-1636.

Lawrence, N. D., and Platt, J. C. 2004. "Learning to Learn with the Informative Vector Machine," in: *Proceedings of the 21st International Conference on Machine learning*, Banff, Alberta, CA: ACM, pp. 65-72.

Li, K. K., and Mohanram, P. 2014. "Evaluating Cross-Sectional Forecasting Models for Implied Cost of Capital," *Review of Accounting Studies* (19:3), pp. 1152-1185.

Li, X.-B. 2009. "A Bayesian Approach for Estimating and Replacing Missing Categorical Data," *Journal of Data and Information Quality* (1:1), pp. 1-11.

Li, X.-B., and Sarkar, S. 2011. "Protecting Privacy against Record Linkage Disclosure: A Bounded Swapping Approach for Numeric Data," *Information Systems Research* (22:4), pp. 774-789.

Li, X., and Hitt, L. M. 2008. "Self-Selection and Information Role of Online Product Reviews," *Information Systems Research* (19:4), pp. 456-474.

Liao, W., and Ji, Q. 2009. "Learning Bayesian Network Parameters under Incomplete Data with Domain Knowledge," *Pattern Recognition* (42:11), pp. 3046-3056.

Little, R.J. 1992. “Regression with Missing X’s: A Review,” *Journal of the American Statistical Association* 87(420): 1227-1237.

Little, R. J., and Rubin, D. B. 2014. *Statistical Analysis with Missing Data*. Hoboken, NJ: John Wiley & Sons.

Luengo, J., García, S., and Herrera, F. 2012. “On the Choice of the Best Imputation Methods for Missing Values Considering Three Groups of Classification Methods,” *Knowledge and Information Systems* (32:1), pp. 77-108.

Makridakis, S., Hogarth, R. M., and Gaba, A. 2009. “Forecasting and Uncertainty in the Economic and Business World,” *International Journal of Forecasting* (25:4), pp. 794-812.

McLachlan, G. J. 1977. “Estimating the Linear Discriminant Function from Initial Samples Containing a Small Number of Unclassified Observations,” *Journal of the American Statistical Association* (72:358), pp. 403-406.

Melville, N., and McQuaid, M. 2012. “Research Note: Generating Shareable Statistical Databases for Business Value: Multiple Imputation with Multimodal Perturbation,” *Information Systems Research* (23:2), pp. 559-574.

Miao, W., Ding, P., and Geng, Z. 2016. “Identifiability of Normal and Normal Mixture Models with Nonignorable Missing Data,” *Journal of the American Statistical Association* 111(516): 1673-1683.

Mithas, S., Whitaker, J., and Tafti, A. 2017. “Information Technology, Revenues, and Profits: Exploring the Role of Foreign and Domestic Operations,” *Information Systems Research* (28:2), pp. 430-444.

Neath, R. C. 2013. “On Convergence Properties of the Monte Carlo EM Algorithm,” in *Advances in Modern Statistical Theory and Applications: A Festschrift in Honor of Morris L. Eaton*. Beachwood: OH: Institute of Mathematical Statistics, pp. 43-62.

Newman, D. A. 2014. “Missing Data: Five Practical Guidelines,” *Organizational Research Methods* (17:4), pp. 372-411.

Pan, S. J., and Yang, Q. 2010. “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering* (22:10), pp. 1345-1359.

Pan, S. J., Zheng, V. W., Yang, Q., and Hu, D. H. 2008. “Transfer Learning for Wifi-Based Indoor Localization,” *Association for the Advancement of Artificial Intelligence (AAAI) Workshop*, Chicago, IL.

Pardoe, D., and Stone, P. 2010. “Boosting for Regression Transfer,” in: *Proceedings of the 27th International Conference on Machine Learning,* J. Fürnkranz and T. Joachims (eds.), Madison, WI: Omnipress, pp. 863-870.

Ravichandran, T., Han, S., and Mithas, S. 2017. “Mitigating Diminishing Returns to R&D: The Role of Information Technology in Innovation,” *Information Systems Research* (28:4), pp. 812-827.

Rotnitzky, A., Robins, J. M., and Scharfstein, D. O. 1998. “Semiparametric Regression for Repeated Outcomes with Nonignorable Nonresponse,” *Journal of the American Statistical Association* (93:444), pp. 1321-1339.

Rubin, D. B. 1976. “Inference and Missing Data,” *Biometrika* (63:3), pp. 581-592.

Rubin, D. B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley & Sons.

Russell, S. J., and Norvig, P. 2016. *Artificial Intelligence: A Modern Approach*, (3rd Global ed.). Upper Saddle River, NJ: Pearson.

Saar-Tsechansky, M., and Provost, F. 2007. “Handling Missing Values When Applying Classification Models,” *Journal of Machine Learning Research* (8:Jul), pp. 1623-1657.

Saboo, A. R., Kumar, V., and Park, I. 2016. “Using Big Data to Model Time-Varying Effects for Marketing Resource (Re)Allocation,” *MIS Quarterly* (40:4), pp. 911-939.

Schafer, J. L. 1997. *Analysis of Incomplete Multivariate Data*. New York: CRC press.

Schafer, J. L., and Graham, J. W. 2002. "Missing Data: Our View of the State of the Art," *Psychological Methods* (7:2), pp. 147-177.

Schölkopf, B. 2017. "Invited Talk - Causal Learning," in: *34th International Conference on Machine Learning*. Sydney, Australia.

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O., and Hemingway, H. 2014. "Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data Using Mice: A Caliber Study," *American Journal of Epidemiology* (179:6), pp. 764-774.

Shi, D., Guan, J., Zurada, J., and Manikas, A. 2017. "A Data-Mining Approach to Identification of Risk Factors in Safety Management Systems," *Journal of Management Information Systems* (34:4), pp. 1054-1081.

Shmueli, G., and Koppius, O. R. 2011. "Predictive Analytics in Information Systems Research," *MIS Quarterly* (35:3), pp. 553-572.

Sivo, S. A., Saunders, C., Chang, Q., and Jiang, J. J. 2006. "How Low Should You Go? Low Response Rates and the Validity of Inference in IS Questionnaire Research," *Journal of the Association for Information Systems* (7:6), pp. 351-414.

Stata. 2013. *Stata Multiple-Imputation Reference Manual Release 13.* College Station, Texas.

Stekhoven, D. J., and Bühlmann, P. 2012. "MissForest—Non-Parametric Missing Value Imputation for Mixed-Type Data," *Bioinformatics* (28:1), pp. 112-118.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. 2008. "Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation," *Advances in Neural Information Processing Systems,* D. Koller, D. Schuurmans, Y. Bengio and L. Bottou (eds.), Vancouver, BC, Canada, pp. 1433-1440.

Sung, Y.J., and Geyer, C.J. 2007. "Monte Carlo Likelihood Inference for Missing Data Models," *Annals of Statistics* 35(3): 990-1011.

Van Buuren, S., and Oudshoorn, K. 2000. *Multivariate Imputation by Chained Equations: MICE V1.0 User's Manual*. Volume Pg/Vgz/00.038. TNO Prevention and Health, Leiden.

Van Buuren, S., and Groothuis-Oudshoorn, K. 2011. "MICE: Multivariate Imputation by Chained Equations in R," *Journal of Statistical Software* (45:3), p. 67.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. New York: Springer Science & Business Media.

Verstraeten, G., and Van den Poel, D. 2005. "The Impact of Sample Bias on Consumer Credit Scoring Performance and Profitability," *Journal of the Operational Research Society* (56:8), pp. 981-992.

Wang, S., Shao, J., and Kim, J.K. 2014. "An Instrumental Variable Approach for Identification and Estimation with Nonignorable Nonresponse," *Statistica Sinica* 24(3): 1097-1116.

Wei, G. C. G., and Tanner, M. A. 1990. "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms," *Journal of the American Statistical Association* 85(411): 699-704.

Yang, Q., and Wu, X. 2006. "10 Challenging Problems in Data Mining Research," *International Journal of Information Technology & Decision Making* (5:4), pp. 597-604.

Ying, Y., Feinberg, F., and Wedel, M. 2006. "Leveraging Missing Ratings to Improve Online Recommendation Systems," *Journal of Marketing Research* (43:3), pp. 355-365.

Zadrozny, B. 2004. "Learning and Evaluating Classifiers under Sample Selection Bias," in: *Proceedings of the 21st International Conference on Machine learning*, Banff, Alberta, Canada: ACM, pp. 114-121.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. 2013. "Domain Adaptation under Target and Conditional Shift," in: *Proceedings of the 30th*

International Conference on Machine Learning, Atlanta, GA: ACM, pp. 819-827.

Zhang, K., Schölkopf, B., Spirtes, P., and Glymour, C. 2017. "Learning Causality and Causality-Related Learning: Some Recent Progress," *National Science Review* (5:1), pp. 26-29.

Zhang, S., Qin, Z., Ling, C. X., and Sheng, S. 2005. "Missing Is Useful: Missing Values in Cost-Sensitive Decision Trees," *IEEE Transactions on Knowledge and Data Engineering* (17:12), pp. 1689-1693.

Zhu, X., and Goldberg, A. B. 2009. "Introduction to Semi-Supervised Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning* (3:1), pp. 1-130.

APPENDIX 1 SUPPLEMENTARY MATERIALS FOR CHAPTER 2

Appendix 1.1 Detailed Review of Statistical Models for Handling Missing Values

This appendix provides detailed descriptions for the state-of-the-art statistical methods under the MAR mechanism so that this thesis is self-contained, including (1) maximum likelihood estimation with EM (information source: Sections 8.3 and 8.4 of Little and Rubin (2014)), (2) multiple imputation (information source: Stata (2013) and Rubin (1987)), and (3) Multivariate Imputation by Chained Equations (MICE, information source: Van Buuren and Oudshoorn (2000)). Each method is presented with an independent set of notations so that they are consistent with the respective information sources as much as possible.

Appendix 1.1.1 Maximum Likelihood Estimation with EM

This sub-section describes the implementation of expectation maximization for maximum likelihood estimation. The information source is the content of Sections 8.3 and 8.4 of Little and Rubin (2014).

Different variables, x_1, x_2, x_3, ..., are not distinguished. Instead, the whole data matrix is denoted with X. X_{obs} and X_{mis} denote the observed part and missing part of this data matrix respectively. In this sense, multiple variables may be subject to missing values, as illustrated in the following figure.

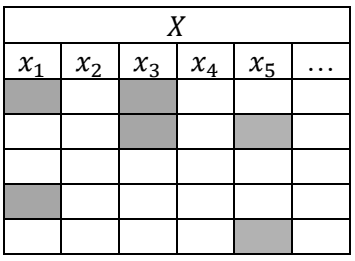


Figure A-1 Illustration of Incomplete Data Matrix

Note: Figure A-1 is adapted from Figure 1.1 of Little and Rubin 2014 (Sect. 1.2).

In Figure A-1, X_{obs} indicates the un-shadowed data entries while X_{mis}

indicates the shadowed (missing) data entries. Such notation system largely simplifies the presentation of theories. For instance, let $f(X|\theta) = f(X_{obs}, X_{mis}|\theta)$ to denote the density function of the joint distribution of X_{obs} and X_{mis}.

Let $f(X|\theta) = f(X_{obs}, X_{mis}|\theta)$ denote the density of the joint distribution of X_{obs} and X_{mis}. θ denotes the parameters of interest that determine the underlying data generation process (DGP) of the data X. The objective is to estimate θ using the incomplete data matrix X. For instance, if assume that X follows a multivariate normal distribution, then θ contains the mean and the variance-covariance matrix.

The marginal probability density of X_{obs} is obtained by integrating out the missing data X_{mis}: $f(X_{obs}|\theta) = \int f(X_{obs}, X_{mis}\,|\theta)\, dX_{mis}$. The likelihood of θ based on data X_{obs} ignoring the missing-data mechanism (assuming MAR) is defined to be any function of θ proportional to $f(X_{obs}|\theta)$:

$$L_{ign}(\theta|X_{obs}) \propto f(X_{obs}|\theta) = \int f(X_{obs}, X_{mis}\,|\theta)\, dX_{mis}, \theta \in \Omega_\theta, \qquad \text{(A.1)}$$

wherein Ω_θ indicates the parameter space of θ. This $L_{ign}(\theta|X_{obs})$ is the objective function of the EM optimization method.

Definition of the EM method is as the following (Little and Rubin 2014, Section 8.3). Let the $\theta^{(t)}$ be the current estimate of the paremeter θ.

The E step of EM finds the expected complete-data log-likelihood over the probability of X_{mis} as if the true θ were $\theta^{(t)}$:

$$Q\left(\theta|\theta^{(t)}\right) = \int f(X_{obs}, X_{mis}\,|\theta) f\left(X_{mis}\,|\theta^{(t)}, X_{obs}\right) dX_{mis}. \qquad \text{(A.2)}$$

The M step of EM determines the $\theta^{(t+1)}$ by increasing this expected complete-data log-likelihood:

$$Q\left(\theta^{(t+1)}|\theta^{(t)}\right) \geq Q\left(\theta|\theta^{(t)}\right),\ for\ all\ \theta \in \Omega_\theta. \qquad \text{(A.3)}$$

Theorem A.1. shows the monotone property of EM. That is, EM algorithm increases $L_{ign}(\theta|X_{obs})$ at each iteration.

Theorem A.1. (Little and Rubin 2014, Sect. 8.4) Every EM algorithm increases $L_{ign}(\theta|X_{obs})$ at each iteration, that is,

$$L_{ign}\left(\theta^{(t+1)}|X_{obs}\right) \geq L_{ign}\left(\theta^{(t)}|X_{obs}\right), \tag{A.4}$$

with equality if and only if

$$Q\left(\theta^{(t+1)}|\theta^{(t)}\right) = Q\left(\theta^{(t)}|\theta^{(t)}\right). \tag{A.5}$$

Appendix 1.1.2 Multiple Imputation

Multiple imputation shares with single imputation two basic advantages, namely, the ability to use complete-data methods of analysis and the ability to incorporate the data collector's knowledge. For the second advantage of single imputation, readers can refer to Rubin (1987). Moreover, multiple imputation overcomes an obvious and important disadvantage of single imputation: the single value being imputed can reflect neither sampling variability about the actual value when one model for the missing data mechanism is being considered nor additional uncertainty when more than one model is being entertained (Rubin 1987, Section 1.4, 1.5). Rubin (1987) formally defined the proper multiple imputation in a general and conceptual way without imposing assumptions on the research sampling scheme and missing data mechanism. However, due to the difficulties in implementing the whole theory, in practice, softwares for multiple imputation only focus on the missing data issue and assume the MAR mechanism.

In the following description of multiple imputation, the information source for the implementation of multiple imputation is the section on the methods' description of mi impute mvn in Stata (2013), and the information source for combining estimates of multiple imputation is Rubin (1987).

Implementation of Multiple Imputation

Under the MAR assumption, for multivariate imputation with arbitrary missing-data patterns, Schafer (1997) proposed a Joint Model approach to implement the multiple imputation. This approach assumes a multivariate distribution for all imputation variables and imputes missing values as draws from the resulting posterior predictive distribution of the missing data given the observed data. The predictive distribution is often difficult to draw from directly, so the imputed values are often obtained by approximating this distribution using data augmentation MCMC method.

Let $x_1^C, x_2^C, \ldots, x_n^C$ (note: x in lower case) be a random sample from a p-variate normal distribution recording values of p complete variables. The sample size is n. Let $x_1^{IC}, x_2^{IC}, \ldots, x_n^{IC}$ be a random sample from a q-variate normal distribution recording values of q incomplete variables. Consider a multivariate normal regression

$$x_i^{IC} = x_i^C \Theta + \epsilon_i, \qquad i = 1, \ldots, n, \tag{A.6}$$

where Θ is a $p \times q$ matrix of regression coefficients, and ϵ_i is a $1 \times q$ vector of random errors from a q-variate normal distribution with a zero mean vector and a $q \times q$ positive-definite covariance matrix Σ. Let Θ and Σ be model parameters of interest. Consider the partition $x_i^{IC} = (x_{i(m)}^{IC}; x_{i(o)}^{IC})$ corresponding to missing and observed values of incomplete variables in observation i for $i = 1,2, \ldots, n$.

The data augmentation MCMC method aims to impute the missing values in x_i independently for each observation $i = 1,2, \ldots, n$. Data augmentation consists of two steps, an I step (imputation step) and a P step (posterior step), performed at each iteration $t = 0,1, \ldots T$. At iteration t of the I step, missing values in x_i are replaced with draws from the conditional posterior distribution of $x_{i(m)}$ given observed data and current values of model parameters independently for each $i = 1,2, \ldots, n$. During the P step, new values of model parameters are drawn from their conditional posterior

distribution given the observed data and the data imputed in the previous I step. Mathematically, this process can be described as follows:

I step:

$$x_{i(m)}^{IC(t+1)} \sim P\left(x_{i(m)}^{IC} \middle| x_i^C, x_{i(o)}^{IC}, \Theta^{(t)}, \Sigma^{(t)}\right), \quad i = 1, \ldots, n. \tag{A.7}$$

P step:

$$\Sigma^{(t+1)} \sim P\left(\Sigma \middle| x_i^C, x_{i(o)}^{IC}, \Theta^{(t)}, x_{i(m)}^{IC(t+1)}\right), \tag{A.8}$$

$$\Theta^{(t+1)} \sim P\left(\Theta \middle| x_i^C, x_{i(o)}^{IC}, x_{i(m)}^{IC(t+1)}, \Sigma^{(t+1)}\right). \tag{A.9}$$

The above steps are repeated until the specified number of iterations, T, is reached. The total number of iterations, T, is determined by the length of the initial MCMC burn-in period, b, and the number of MCMC iterations between imputations, k. Specifically, $T = b + m \times k$, where m is the number of imputations to be created. b must be large enough so that the above chain converges to the stationary distribution $P\left(x_{i(m)}^{IC}, \Sigma, \Theta \middle| x_i^C, x_{i(o)}^{IC}\right)$ by iteration $t = b$. k must be large enough so that random draws for different imputations are approximately independent.

Combining Estimates of Multiple Imputation

Assume that with complete data, inferences for θ would be based on the statement that

$$(\theta - \hat{\theta}) \sim N(0, U), \tag{A.10}$$

where $\hat{\theta}$ is a statistic estimating θ, U is a statistic providing the variance of ($\hat{\theta}$ - θ), and $N(0, U)$ is the normal distribution with mean 0 and variance U.

Suppose that under a specified Bayesian model, m sets of repeated imputations have been drawn and used to construct m completed data sets. Let $\hat{\theta}_{*1}$,

$\hat{\theta}_{*2}$,…, $\hat{\theta}_{*m}$ be the value of statistics $\hat{\theta}$ for each of these data sets. Let U_{*1}, U_{*2},…, U_{*m} be the value of statistics U for each of these data sets.

The m repeated complete-data estimates and associated complete-data variances for θ under a model for nonresponse can be combined as follows (Rubin 1987, p. 76). Let

$$\bar{\theta}_m = \frac{\sum_{l=1}^{m} \hat{\theta}_{*l}}{m}, \tag{A.11}$$

be the average of the m complete-data estimates,

$$\bar{U}_m = \frac{\sum_{l=1}^{m} U_{*l}}{m}, \tag{A.12}$$

be the average of the m complete-data variances, and

$$B_m = \frac{\sum_{l=1}^{m} (\hat{\theta}_{*l} - \bar{\theta}_m)' (\hat{\theta}_{*l} - \bar{\theta}_m)}{m-1}, \tag{A.13}$$

be the variance between the m complete-data estimates, where the superscript in $(\hat{\theta}_{*l} - \bar{\theta}_m)'$ indicates transpose when θ is a vector. The $\bar{\theta}_m$ is the overall estimate of θ. The quantity $T_m = \bar{U}_m + (1 + m^{-1})B_m$ is the total variance of $(\theta - \bar{\theta}_m)$.

For confidence intervals, Rubin (1987, p. 130) recommends using a Student's t approximation,

$$T_m^{-1/2} (\theta - \bar{\theta}_m) \sim t_v, \tag{A.14}$$

where the degree of freedom is

$$v = (m-1)\left[1 + \frac{\bar{U}_m}{(1 + m^{-1})B_m}\right]^2. \tag{A.15}$$

Generally speaking, the validity of the inference will hold if the following assumptions are satisfied (Rubin 1987, p. 128): (i) proper imputation methods, (ii) valid complete-data inferences, and (iii) samples large enough.[32] More precise

[32] Generally speaking, assumption (ii) holds under the posited true response mechanism, specified sampling mechanism, and the true population data distribution (Rubin 1987, p. 116).

formulas for those assumptions are listed in formula (3.3.2), (3.3.3), (4.2.3)-(4.2.10) in the book by Rubin (1987).

Appendix 1.1.3 MICE Imputation Algorithm

This section refers to Van Buuren and Oudshoorn (2000) for detailed descriptions of MICE. Let data matrix X^C be $\left[X_1^C, X_2^C, X_3^C, \dots, X_p^C\right]_{n\times p}$ consisting of p predictors that are all complete. Let data matrix X^{IC} be $\left[X_1^{IC}, X_2^{IC}, X_3^{IC}, \dots, X_q^{IC}\right]_{n\times q}$ consisting of q predictors that are all incomplete. The problem is to impute the missing values of X^{IC} by drawing from $P(X^{IC})$, the unconditional multivariate distribution of X^{IC}. Let t denote an iteration counter. Assuming that data are missing at random (MAR), one may repeat the following sequence of Gibbs sampler iterations, i.e., drawing the incomplete varaible conditional on the complete variables and the most recently drawn values of all other incomplete variables.

For X_1^{IC}: draw imputations $X_1^{IC(t+1)}$ from

$$P\left(X_1^{IC}\middle|X_2^{IC(t)}, X_3^{IC(t)}, \dots, X_q^{IC(t)}, X_1^C, X_2^C, \dots, X_p^C\right),$$

For X_2^{IC}: draw imputations $X_2^{IC(t+1)}$ from

$$P\left(X_2^{IC}\middle|X_1^{IC(t)}, X_3^{IC(t)}, \dots, X_q^{IC(t)}, X_1^C, X_2^C, \dots, X_p^C\right),$$

…

For X_q^{IC}: draw imputations $X_q^{IC(t+1)}$ from

$$P\left(X_q^{IC}\middle|X_1^{IC(t)}, X_2^{IC(t)}, \dots, X_{q-1}^{IC(t)}, X_1^C, X_2^C, \dots, X_p^C\right),$$

Rubin and Schafer (1990) show that if $P(X^{IC})$ is multivariate normal, then iterating linear regression models like $X_1^{IC} = X_2^{IC(t)}\beta_{i2} + X_3^{IC(t)}\beta_{i3} + \cdots + X_q^{IC(t)}\beta_{iq} + \varepsilon_i$ with $\varepsilon_i \sim N\left(0, \sigma_i^2\right)$ will produce a random draw from the desired distribution.

Appendix 1.2 Lemma 2.1 with Continuous Variable z

Lemma 2.1. Let $\mathbb{Z}$ be the value space of the continuous variable z. The inequality

$g(\boldsymbol{\psi}^*) \geq g(\mathbf{0})$ holds if $\boldsymbol{\psi}^*$ is optimum for maximizing function $\hat{g}(\boldsymbol{\psi})$, where

$$g(\boldsymbol{\psi}) = \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}) \text{ , and}$$

$$\hat{g}(\boldsymbol{\psi}) = \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \int_{\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^*) \ln p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}) d\tilde{z} \text{ .}$$

In the expression of $g(\boldsymbol{\psi})$, $\Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi})$ indicates the marginal probability of $s = s_i$ conditional on $\boldsymbol{x}_i$, which is obtained by integrating the joint probability of z_i and s_i conditional on $\boldsymbol{x}_i$ with respect to z_i, namely $\int_{\mathbb{Z}} \Pr(s_i, \tilde{z}|\boldsymbol{x}_i; \boldsymbol{\psi}, \boldsymbol{\theta}^*) d\tilde{z}$.

Proof sketch:

First, it can be proved that $g(\boldsymbol{\psi}) \geq \hat{g}(\boldsymbol{\psi})$:

$$\begin{aligned}
g(\boldsymbol{\psi}) &= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}) \\
&= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \Pr(s_i|\boldsymbol{x}_i; \boldsymbol{\psi}, \boldsymbol{\theta}^*) \\
&= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \int_{\mathbb{Z}} \Pr(s_i, \tilde{z}|\boldsymbol{x}_i; \boldsymbol{\psi}, \boldsymbol{\theta}^*) d\tilde{z} \\
&= \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \ln \int_{\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^*) p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}) d\tilde{z} \\
&\geq \sum_{i=1}^{m} \ln p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi}) + \sum_{i=m+1}^{m+n} \int_{\mathbb{Z}} q(\tilde{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^*) \ln p(s_i|\tilde{z}, \boldsymbol{x}_i; \boldsymbol{\psi}) d\tilde{z} \\
&= \hat{g}(\boldsymbol{\psi}).
\end{aligned}$$

The first equality holds since the relationship between parameter $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$ are not constrained (e.g., by imposing a prior assumption). The last inequality is supported by Jensen's inequality.

When $\boldsymbol{\psi} = \mathbf{0}$, $p(s_i|z_i, \boldsymbol{x}_i; \boldsymbol{\psi} = \mathbf{0})$ is constant with respect to z_i. Under this condition, $g(\boldsymbol{\psi}) \geq \hat{g}(\boldsymbol{\psi})$ holds with equality, namely $g(\mathbf{0}) = \hat{g}(\mathbf{0})$. Since $\boldsymbol{\psi}^*$ is optimum for maximizing $\hat{g}(\boldsymbol{\psi})$, there is $\hat{g}(\boldsymbol{\psi}^*) \geq \hat{g}(\mathbf{0})$. Since $g(\boldsymbol{\psi}) \geq \hat{g}(\boldsymbol{\psi})$ holds in general cases, there is $g(\boldsymbol{\psi}^*) \geq \hat{g}(\boldsymbol{\psi}^*) \geq \hat{g}(\mathbf{0}) = g(\mathbf{0})$.

Q.E.D.

Appendix 1.3 Technical Details of Monte Carlo Likelihood Estimation

In this appendix section, we present the detailed estimation process for parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$.

The objective function in Equation (2.12) can be written as:

$$l_{full}(z_{obs}, s|x, y; \boldsymbol{\theta}, \boldsymbol{\psi}) = \sum_{i=1}^{m} \ln[\Pr(s_i, z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \sum_{i=m+1}^{m+n} \ln[\Pr(s_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})]$$

$$= \sum_{i=1}^{m} \ln[\Pr(s_i|x_i, y_i, z_i; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \sum_{i=m+1}^{m+n} \ln[\Pr(s_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})]$$

The Expectation of the objective function over the posterior distribution of z given observed values of x, y and s, as well as the parameter estimation at iteration t, $(\boldsymbol{\theta}^t, \boldsymbol{\psi}^t)$, is:

$$\mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\psi}|\boldsymbol{\theta}^t, \boldsymbol{\psi}^t) = \sum_{i=1}^{m} \ln[\Pr(s_i|x_i, y_i, z_i; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \sum_{i=m+1}^{m+n} \{\int \ln[\Pr(s_i, z|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] \times \Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})\, dz\}$$

$$= \sum_{i=1}^{m} \ln[\Pr(s_i|x_i, y_i, z_i; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \sum_{i=m+1}^{m+n} \{\int \ln[\Pr(s_i|x_i, y_i, z; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] \times \Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})\, dz\}$$

$$= \sum_{i=1}^{m} \ln[\Pr(s_i|x_i, y_i, z_i; \boldsymbol{\theta}, \boldsymbol{\psi}) \times f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] + \sum_{i=m+1}^{m+n} \{\int \ln[\Pr(s_i|x_i, y_i, z; \boldsymbol{\theta}, \boldsymbol{\psi})] \times \Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})\, dz\} + \sum_{i=m+1}^{m+n} \{\int \ln[f(z_i|x_i, y_i; \boldsymbol{\theta}, \boldsymbol{\psi})] \times \Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})\, dz\}$$

The posterior distribution of z given observed values of x, y and s, namely $\Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})$, can be written as:

$$\Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t}) = \frac{\Pr(z, s_i|x_i, y_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})}{\Pr(s_i|x_i, y_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})}$$

$$= \frac{\Pr\left(z, s_i \middle| x_i, y_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t}\right)}{\int \Pr\left(z, s_i \middle| x_i, y_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t}\right) dz}$$

$$\propto \Pr(z, s_i | x_i, y_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})$$

$$= \Pr(s_i | x_i, y_i, z; \boldsymbol{\psi^t}) \times f(z | x_i, y_i; \boldsymbol{\theta^t})$$

Therefore, the expectation is approximated by drawing z from $\Pr(z|x_i, y_i, s_i; \boldsymbol{\theta^t}, \boldsymbol{\psi^t})$ through the Metropolis–Hastings algorithm, a MCMC method, with the target stationary distribution being $\Pr(s_i|x_i, y_i, z; \boldsymbol{\psi^t}) \times f(z|x_i, y_i; \boldsymbol{\theta^t})$.

The maximization step maximizes the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ in the numerically approximated expectation results. Note that, since function $\mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\psi}|\boldsymbol{\theta}^t, \boldsymbol{\psi}^t)$ is written as the sum of the terms of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ separately, the optima of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ can be solved independently. We checked the convergence of the EM algorithm and the iterative steps are terminated when the squared distance between the *t*-th and the *t*+1-th iteration for parameter $\boldsymbol{\theta}$ is less than 10^{-6}.

Appendix 1.4 Supplementary Results on Bias Correction

Appendix 1.4.1 Tabulated Results of Bias in Coefficient Estimation under Different Missing Value Percentages

Figures 2-9, 2-10, and 2-11 in Chapter 2 presented the average absolute bias of the three regression coefficients. In this appendix section, we tabulate the results of each of the three regression coefficients for different missing value percentages (Tables A-1 through A-3 for missing value percentage at 10%, 20% and 40%, respectively). 1%, 5%, and 10% statistical significance are shaded and indicated with ***, **, and *, respectively. The results in Tables A1 through A3 are qualitatively similar to Table 2-10 in Chapter 2. As the missing value percentage increases, the bias of parameter estimation increases. This trend is particularly visible for the listwise deletion and the ML-MAR methods. The proposed method generates approximately unbiased estimates under the majority values of ψ_y and ψ_z even when the missing value

percentage is high. Similar to Table 2-10, the proposed method comes with fewer cells having significantly biased parameter estimation and overall lower magnitude in the bias.

		β_0				β_1				β_2			
Method	ψ_z \ ψ_y	0	2	4	6	0	2	4	6	0	2	4	6
LD	0	0.001	0.130***	0.154***	0.158***	-0.002	-0.117***	-0.155***	-0.158***	-0.003	0.115***	0.155***	0.160***
	2	0.001	0.128***	0.160***	0.163***	0.000	-0.111***	-0.168***	-0.177***	-0.006***	-0.001	0.085***	0.117***
	4	-0.002	0.105***	0.154***	0.161***	0.000	-0.070***	-0.151***	-0.177***	-0.004*	-0.072***	0.000	0.058***
	6	0.000	0.081***	0.138***	0.156***	0.002	-0.041***	-0.119***	-0.159***	-0.002	-0.084***	-0.059***	0.002
ML-MAR	0	-0.005	0.001	0.002	0.002	0.008	-0.001	-0.003	0.003	-0.017***	-0.001	0.002	-0.002
	2	0.068***	0.060***	0.043***	0.028***	-0.023***	-0.031***	-0.021***	-0.013***	-0.038***	-0.039***	-0.018***	-0.015***
	4	0.089***	0.091***	0.076***	0.054***	-0.036***	-0.055***	-0.049***	-0.034***	-0.056***	-0.074***	-0.049***	-0.032***
	6	0.099***	0.101***	0.095***	0.076***	-0.039***	-0.060***	-0.064***	-0.051***	-0.061***	-0.087***	-0.074***	-0.050***
ML-MC	0	-0.007	0.001	0.002	0.002	-0.002	-0.003	-0.004*	0.002	-0.002	0.003	0.004*	-0.001
	2	0.020***	0.003	0.003*	0.000	-0.008***	0.001	-0.001	-0.001	-0.014***	-0.001	0.003	-0.001
	4	-0.002	0.000	0.003	0.000	0.000	0.001	-0.002	-0.001	-0.003	-0.002	0.000	-0.001
	6	0.000	-0.003*	0.001	0.000	0.002	0.001	-0.002	-0.001	-0.002	-0.001	-0.001	0.003

Table A-1 Estimation of Beta Coefficients (Missing Value Percentage = 10%)

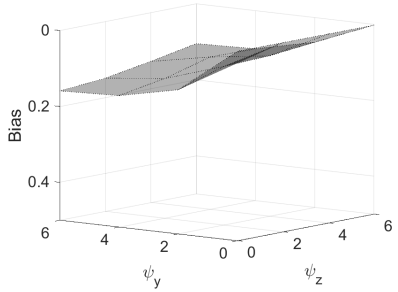
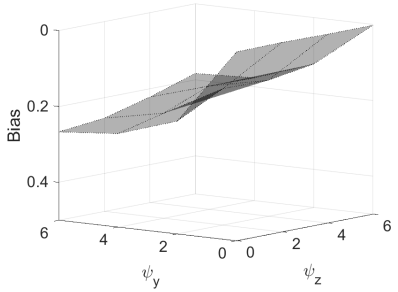
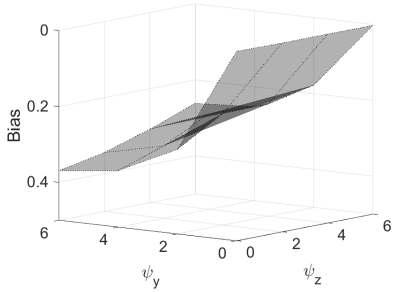
		β_0				β_1				β_2			
Method	ψ_z \ ψ_y	0	2	4	6	0	2	4	6	0	2	4	6
LD	0	-0.016**	0.248***	0.292***	0.300***	0.015*	-0.180***	-0.236***	-0.249***	-0.017**	0.181***	0.236***	0.248***
	2	0.004*	0.247***	0.294***	0.311***	0.001	-0.176***	-0.257***	-0.272***	-0.004	-0.002	0.126***	0.182***
	4	0.000	0.214***	0.292***	0.310***	0.003	-0.114***	-0.232***	-0.269***	-0.002	-0.120***	0.000	0.090***
	6	0.002	0.177***	0.271***	0.301***	0.002	-0.073***	-0.185***	-0.253***	-0.002	-0.138***	-0.094***	0.000
ML-MAR	0	-0.006	0.000	-0.001	-0.001	0.021***	-0.002	0.002	0.002	-0.033***	0.002	0.000	-0.003
	2	0.139***	0.124***	0.081***	0.060***	-0.036***	-0.059***	-0.038***	-0.023***	-0.056***	-0.060***	-0.032***	-0.019***
	4	0.183***	0.188***	0.153***	0.117***	-0.055***	-0.091***	-0.082***	-0.059***	-0.088***	-0.123***	-0.075***	-0.047***
	6	0.202***	0.214***	0.195***	0.161***	-0.064***	-0.099***	-0.109***	-0.097***	-0.099***	-0.144***	-0.116***	-0.077***
ML-MC	0	-0.003	-0.004	-0.002	-0.002	0.013**	-0.004**	0.001	0.002	-0.019***	0.009***	0.003	-0.001
	2	0.047***	0.008***	-0.003*	0.000	-0.009***	-0.004**	-0.001	0.001	-0.023***	-0.004*	0.000	0.002
	4	0.004**	0.002	0.002	0.000	0.003*	0.001	0.002	0.001	-0.006***	-0.006**	0.000	0.000
	6	0.002	0.002	0.002	0.000	0.003	-0.001	-0.002	-0.005**	-0.002	0.001	0.001	0.002

Table A-2 Estimation of Beta Coefficients (Missing Value Percentage = 20%)

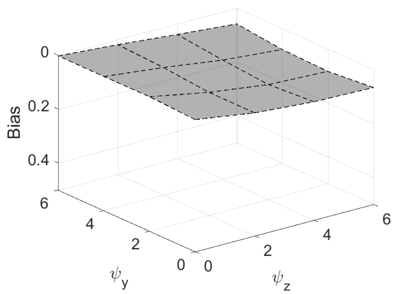
		β_0				β_1				β_2			
Method	ψ_y / ψ_z	0	2	4	6	0	2	4	6	0	2	4	6
LD	0	-0.003	0.492***	0.592***	0.616***	-0.010	-0.266***	-0.363***	-0.385***	0.011	0.265***	0.363***	0.383***
	2	-0.002	0.493***	0.593***	0.618***	0.001	-0.265***	-0.384***	-0.408***	0.000	0.000	0.190***	0.276***
	4	-0.001	0.458***	0.586***	0.620***	0.002	-0.191***	-0.354***	-0.405***	-0.001	-0.185***	-0.002	0.135***
	6	-0.001	0.400***	0.571***	0.615***	0.000	-0.122***	-0.300***	-0.380***	0.001	-0.242***	-0.147***	-0.001
ML-MAR	0	-0.002	0.000	-0.001	0.001	0.007	0.000	0.000	0.001	-0.005	-0.003	0.002	-0.002
	2	0.277***	0.269***	0.198***	0.146***	-0.055***	-0.099***	-0.076***	-0.049***	-0.084***	-0.083***	-0.041***	-0.026***
	4	0.392***	0.408***	0.347***	0.280***	-0.092***	-0.154***	-0.154***	-0.118***	-0.139***	-0.192***	-0.107***	-0.063***
	6	0.436***	0.475***	0.437***	0.370***	-0.113***	-0.163***	-0.195***	-0.170***	-0.159***	-0.251***	-0.180***	-0.113***
ML-MC	0	-0.004	0.003	-0.002	-0.001	-0.001	-0.006***	-0.002	-0.001	0.007	0.008***	0.007***	0.002
	2	0.191***	0.048***	0.000	0.000	-0.031***	-0.015***	-0.005**	-0.002	-0.070***	-0.016***	0.004	0.006**
	4	0.041***	0.029***	0.004	0.001	-0.004*	-0.009***	-0.001	0.000	-0.025***	-0.022***	0.000	0.003
	6	0.007***	0.011***	0.005**	0.000	-0.003	-0.002	-0.005***	0.001	-0.004*	-0.012***	0.003	0.002

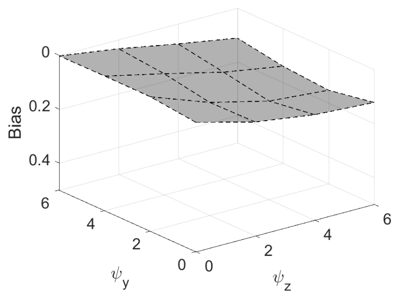
Table A-3 Estimation of Beta Coefficients (Missing Value Percentage = 40%)

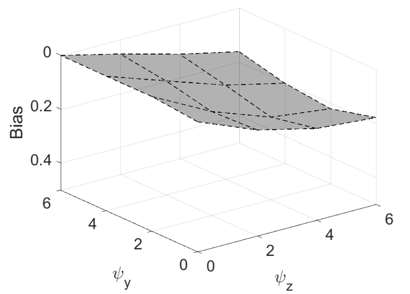
Appendix 1.4.2 Results of Common Missing Value Handling Methods Presented in 3D Plots

	Missing Value Percentage		
	10% missing	20% missing	30% missing
Listwise Deletion			

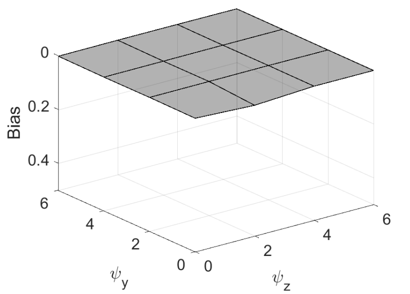
ML-MAR

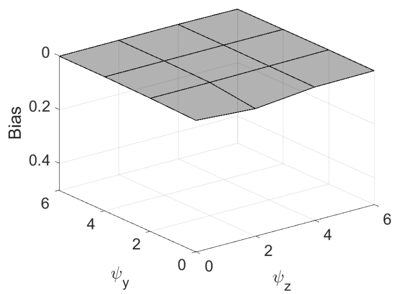


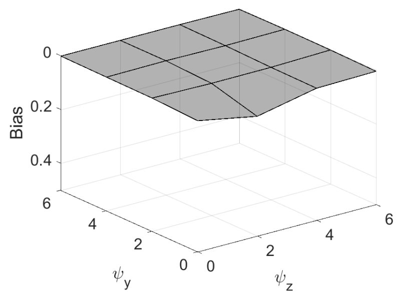




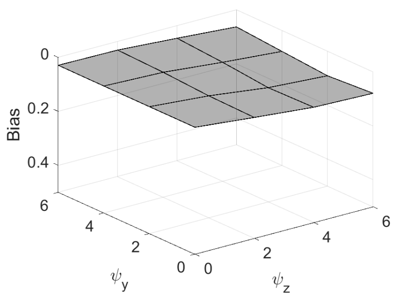
ML-Monte Carlo (proposed method)

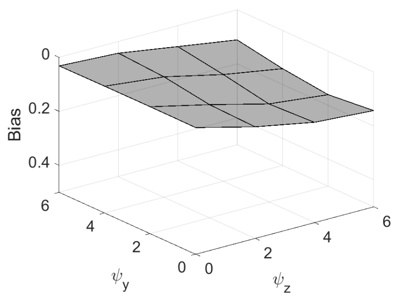


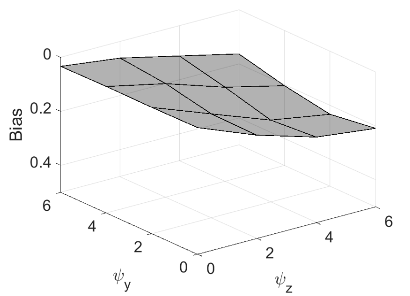




Multiple Imputation







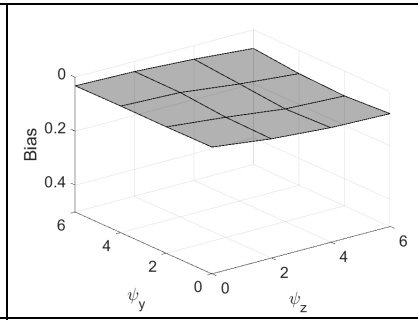
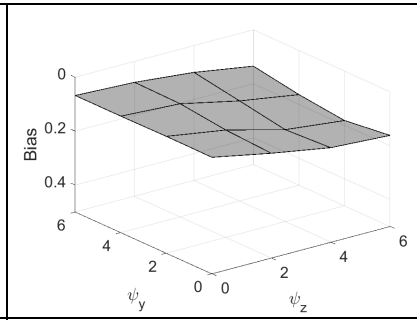
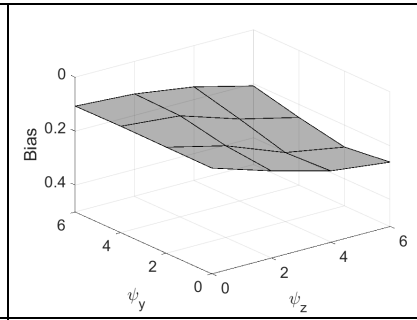
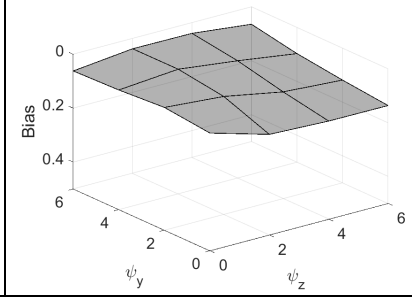
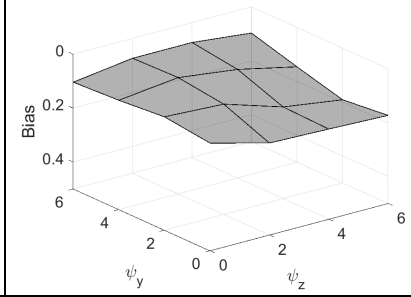
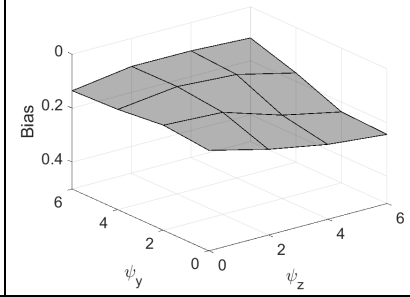
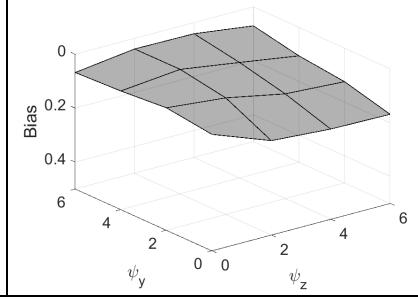
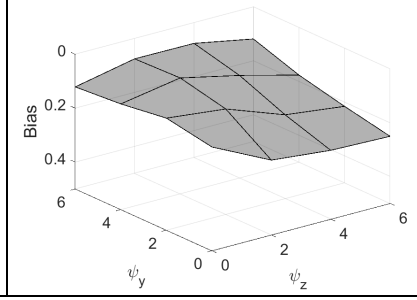
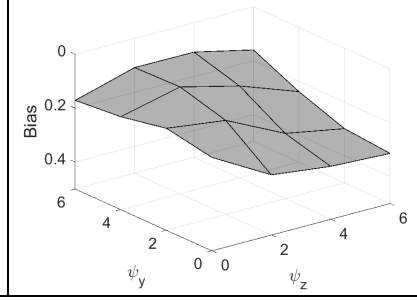
Conditional Mean Imputation (Single Imputation)			
Zero Imputation			
Mean Imputation			

Figure A-2 Bias of Regression Coefficients Using Different Missing Value Handling Methods

Overall, Table A-4 summarizes the bias of coefficient estimates using difference methods. The commonly used listwise deletion method does not cause bias when $\psi_y = 0$ (namely when $\psi_y = 0, \psi_z = 0$, or $\psi_y = 0, \psi_z \neq 0$). Multiple imputation and maximum likelihood ignoring the missingness mechanism generally do not cause bias when $\psi_z = 0$ (namely when $\psi_y = 0, \psi_z = 0$, or $\psi_y \neq 0, \psi_z = 0$). The proposed method generally obtains unbiased estimates under different values of ψ_y and ψ_z. Zero imputation, mean imputation and conditional mean imputation generally lead to bias under all different simulation settings.

	Simulation settings			
	$\psi_y = 0, \psi_z = 0$	$\psi_y \neq 0, \psi_z = 0$	$\psi_y \neq 0, \psi_z \neq 0$	$\psi_y = 0, \psi_z \neq 0$
Missingness Mechanism	MCAR	MAR	NMAR	NMAR
Listwise deletion	**Unbiased**	Biased	Biased	**Unbiased**
Multiple imputation	**Unbiased**	**Unbiased**	Biased	Biased
ML ignoring missingness mechanism	**Unbiased**	**Unbiased**	Biased	Biased
ML incorporating missingness mechanism	**Unbiased**	**Unbiased**	**Unbiased**	**Unbiased**
Zero Imputation	Biased	Biased	Biased	Biased
Mean imputation	Biased	Biased	Biased	Biased
Conditional Mean Imputation (Single Imputation)	Biased	Biased	Biased	Biased

Table A-4 Comparing Missing Value Handling Methods under Different Simulation Settings

Appendix 1.4.3 Results under Alternative Simulation Scenarios

We further experimented the proposed approach in two alternative simulation scenarios to show its robustness to the miss-specification of the missingness mechanism and its generalizability. In this appendix section, we tabulate results of bias for each of the three regression coefficients, β_0, β_1 and β_2, under the simulation settings in which (1) the underlying missingness mechanism is represented with a probit model whereas we specify it to be a logit model during the parameter estimation process (in Table A-5), (2) the regression model is in a generalized linear form where the relationship between the dependent variable and independent variables is represented in a logit model (in Table A-6). 1%, 5%, and 10% statistical significance are shaded and indicated with ***, **, and *, respectively. Results are based on one hundred replications for each simulation setting. Results show that the proposed method comes with few cells having significantly biased parameter estimation. Moreover, since the standard errors of the coefficients, β_0, β_1 and β_2 are around 0.03, the few biased cells are unlikely to result in problematic estimates based on the benchmark of one half of the standard error (Schafer and Graham 2002).

		β_0				β_1				β_2			
Missing%	ψ_z \ ψ_y	0	2	4	6	0	2	4	6	0	2	4	6
10%	0	0.0004	0.0013	-0.0052	-0.0052	-0.0085	-0.0035	-0.0021	0.0025	0.0124***	0.0024	0.0065	0.0013
	2	0.0002	0.0023	-0.0036	-0.0012	-0.0061	0.0016	0.0044	-0.0010	-0.0036	-0.0031	0.0054	-0.0019
	4	-0.0022	0.0050	-0.0019	-0.0052*	-0.0015	0.0032	-0.0045	-0.0007	-0.0008	0.0068	0.0062	-0.0002
	6	0.0030	0.0018	0.0066*	-0.0014	-0.0029	0.0016	-0.0079**	-0.0035	-0.0004	0.0035	0.0016	0.0014
20%	0	0.0010	0.0033	0.0000	-0.0078*	0.0005	-0.0090*	0.0020	0.0145***	0.0052	0.0044	-0.0086**	-0.0020
	2	-0.0020	-0.0012	0.0024	-0.0011	-0.0015	0.0083*	-0.0012	0.0039	0.0042	0.0026	0.0013	-0.0063
	4	-0.0019	-0.0051	0.0034	-0.0049	-0.0004	0.0074*	-0.0003	0.0082**	0.0049	-0.0004	-0.0032	-0.0058
	6	-0.0020	-0.0041	0.0021	-0.0018	0.0038	0.0030	0.0039	0.0015	0.0033	-0.0001	0.0026	0.0009
30%	0	-0.0071	0.0045	0.004	0.0008	-0.0026	-0.0068	-0.0081*	-0.0028	0.0103**	0.0079*	0.0015	0.0001
	2	0.0015	-0.0047	-0.002	0.0000	0.0007	-0.0089*	0.0073	-0.0024	-0.0058	0.0147***	0.0023	0.0003
	4	0.0014	0.0004	0.0006	0.0011	-0.0052	0.0022	-0.0005	-0.0079*	0.0141***	-0.0004	-0.0004	0.0094*
	6	-0.0033	-0.0023	0.0032	0.0000	-0.0005	0.0043	-0.0064*	-0.0045	0.0056	0.0026	0.0079	0.0056
40%	0	0.0081	0.0128*	0.0064	-0.0053	-0.0129	-0.0160***	-0.0086*	-0.0051	0.0222***	0.0009	0.0046	0.0054
	2	0.0045	-0.0055	0.0036	-0.0077	-0.0046	0.0025	-0.0001	-0.0071	-0.0064	0.0042	0.0036	0.0034
	4	0.0011	-0.0003	-0.0004	0.0006	0.0007	0.0031	0.0025	-0.0041	0.0030	-0.0060	-0.0005	0.0085
	6	-0.0037	-0.0036	0.001	0.0008	-0.0054	0.0002	-0.0028	0.0002	0.0024	-0.0008	-0.0040	-0.0033

Table A-5 Results of Estimation of Coefficients under Miss-specified Missingness Mechanism

		β_0				β_1				β_2			
Missing%	ψ_z \ ψ_y	0	2	4	6	0	2	4	6	0	2	4	6
10%	0	0.0049	0.0084	-0.0159**	0.0006	0.0089	0.0015	-0.0043	0.0000	0.0084	-0.0088	0.0027	-0.0085
	2	-0.0005	-0.0002	-0.0056	0.0003	0.0131	0.0062	0.0033	0.0077	-0.0107	-0.0066	-0.0132	-0.0087
	4	0.0053	-0.0018	-0.0138*	0.0148**	-0.0030	0.0073	0.0066	0.0083	-0.0043	-0.0119	-0.0142	-0.0234***
	6	-0.009	-0.0068	-0.0023	-0.0006	0.0039	0.0047	-0.0003	0.0168**	-0.0119	-0.0053	-0.0084	-0.0210**
20%	0	-0.0053	0.0006	0.0069	-0.0018	0.0027	0.0066	-0.0080	0.0001	-0.0058	0.0027	0.0101	0.0231
	2	0.0039	0.0067	0.0006	0.0030	-0.0010	-0.0073	0.0056	0.0063	-0.0054	-0.0007	0.0000	0.0019
	4	0.0021	0.0057	-0.0154**	0.0034	-0.0002	0.0042	-0.0025	0.0041	0.0047	0.0007	-0.004	0.0084
	6	-0.0091	0.0028	-0.0079	0.0133*	0.0006	-0.0087	0.0106	0.0092	-0.0115	0.0004	-0.0002	-0.0076
30%	0	-0.0170	0.0170*	0.0144*	0.0104	0.0072	0.0034	0.0018	-0.0027	0.0166	-0.0065	-0.0083	-0.0010
	2	0.0082	0.0074	0.0020	0.0067	-0.0149*	-0.0088	0.0057	0.0136	0.0026	-0.0046	-0.0046	-0.0114
	4	-0.0054	-0.0063	0.0003	0.0003	0.0068	0.0020	0.0207**	-0.0052	-0.0035	-0.0036	-0.0169	0.0109
	6	-0.0012	0.0121*	0.0031	0.0017	0.0066	0.0031	-0.0028	-0.0108	-0.0010	0.0027	-0.0069	-0.0113
40%	0	0.0139	0.0278**	0.0206	0.0592***	-0.0206*	-0.0181	-0.0323*	0.0437**	0.0325***	-0.0003	0.0560	-0.0684**
	2	0.0177**	0.0175**	0.0159**	0.0112	0.0152	0.0135	-0.0018	0.0107	-0.0191*	-0.0426***	-0.0085	-0.0115
	4	0.0047	0.0104	-0.0011	-0.0053	-0.0242***	-0.0083	-0.0007	-0.0018	0.0151	-0.0046	-0.0030	-0.0051
	6	0.0001	0.0016	-0.0003	-0.0076	0.0126	0.0030	0.0158*	0.0154*	-0.0120	-0.0076	-0.0110	-0.0132

Table A-6 Results of Estimation of Coefficients in Generalized Linear Regression

Appendix 1.4.4 Experimentation Results when Missing Values Occur in both Dependent and Independent Variables

In this appendix section, we explore the extension of our proposed Monte Carlo likelihood estimation to handle the situation where both the dependent variable and the independent variable contain missing values. We simulate the same data generation process as in Section 2.5.1 as below:

$$y = \beta_0 + \beta_1 x + \beta_2 z + \varepsilon,$$

where $\binom{x}{z} \sim N(\mu, \sigma^2)$, $\mu = \binom{0}{0}$, $\sigma^2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$, $\varepsilon \sim N(0,1)$. The values of coefficients are set to $\beta_0 = 0$, $\beta_1 = 1$ and $\beta_2 = -1$. One thousand data samples of the values of (x, z, y) are drawn from the above data generating process.

Missing values are imposed on variables on z and y according to the following mechanisms:

$$p(s_z = 1|x, y, z; \boldsymbol{\psi}_z) = \frac{1}{1+e^{-(\psi_{0z}+x+2y+\psi_{zz}z)}}$$

$$p(s_y = 1|x, y, z; \boldsymbol{\psi}_y) = \frac{1}{1+e^{-(\psi_{0y}+x+\psi_{yy}y+3z)}}$$

wherein ψ_{zz} takes value from {0,2,4} and ψ_{yy}takes value from {0,3,6}.The missing value percentage for both variable z and y is set to 20% by solving the intercept term ψ_{0z} or ψ_{0y}. Letting the value of ψ_{zz} (or ψ_{yy}) not to be zero makes the missingness mechanism for z (or y) NMAR.

To employ our Monte Carlo based computation approach in multi-variable missing scenario, we use Gibbs sampling in the E step to take care of the observations where both variables z and y are missing (this is alternative to using Metropolis–Hastings algorithm to handle observations that only z or y is missing, as indicated in Appendix 1.3).

We employ the Multivariate Imputation by Chained Equations (MICE) in R package (Van Buuren and Groothuis-Oudshoorn 2011) as a benchmark as it is a popular method for handling the situation of multiple missing variables (Shah et al. 2014). Detailed description of the MICE method is presented in Appendix 1.1.3. Moreover, we implemented MICE in two ways. First, we employ the data set consisting of x, incomplete y and incomplete z as the input of the MICE package. Second, we additionally include the two dummy variables, s_z and s_y, as input to allow possibly supplementary information employed by MICE.

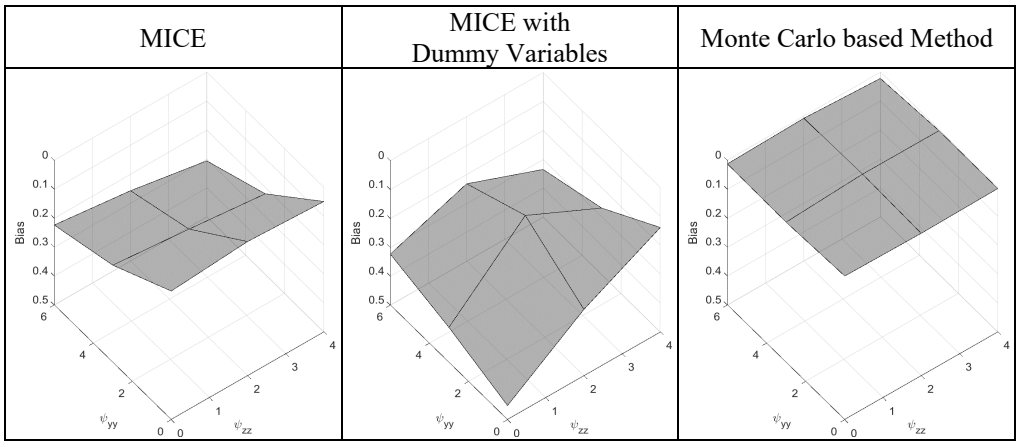


Figure A-3 Bias of Regression Coefficients When Missing Values Occur in Both Dependent and Independent Variables

Figure A-3 above shows average absolute bias of three beta coefficients using the two benchmark methods and the proposed method. Results show that MICE, which ignores the missingness mechanism generally leads to biased regression coefficients. Adding the dummy variables, s_z and s_y, does not help to reduce the bias. The proposed Monte Carlo based approach generates minimal bias under different values of ψ_{zz} and ψ_{yy}. Finally, it is worth noting that, although maximum likelihood estimation under the situation of missing variables being missing can be computationally solved, the theoretical guarantee on the statistical properties under NMAR, such as identifiability and consistency of the estimates are under explored. Therefore, the computation method implemented in this section can be viewed as a sensitivity analysis for analyzing data with multiple incomplete variables.

APPENDIX 2 SUPPLEMENTARY MATERIALS FOR CHAPTER 3

Appendix 2.1 Proof of Theorem 3.2

Theorem 3.2 According to **Lemma 3.1** and **Lemma 3.2**, for any $\delta > 0$, we have:

$$\Pr\left(|L_T - E^{r=0}| \geq \frac{p}{p+q}\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)} + \frac{q}{p+q}\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}\right) \leq \delta.$$

Proof sketch

To simplify notations, let $\zeta \equiv \frac{p}{p+q}\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)} + \frac{q}{p+q}\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}$. Let $l_i(\boldsymbol{\theta})$ denote $l\left(\boldsymbol{x}_i^{D-S}, y_i^{D-S}; \boldsymbol{\theta}\right)$ or $l\left(\boldsymbol{x}_i^{D-S}, y_i^{D-S}; \boldsymbol{\theta}\right)$ – whether the data record is a same-distribution or diff-distribution one can be told by the number of terms in the summation (p for diff-distribution data while q for same-distribution data).

First, by triangle inequality, we have

$$\begin{aligned}|L_T - E^{r=0}| &= \left|\frac{1}{p+q}\sum_{i=1}^{p}[w_i l_i(\boldsymbol{\theta})] - \frac{p}{p+q}E^{r=0} + \frac{1}{p+q}\sum_{i=1}^{q}[l_i(\boldsymbol{\theta})] - \frac{q}{p+q}E^{r=0}\right| \\ &\leq \frac{p}{p+q}|L_{D-S} - E^{r=0}| + \frac{q}{p+q}|L_{S-S} - E^{r=0}|.\end{aligned}$$

Thus, we have

$$\Pr(|L_T - E^{r=0}| \geq \zeta) \leq \Pr\left(\frac{p}{p+q}|L_{D-S} - E^{r=0}| + \frac{q}{p+q}|L_{S-S} - E^{r=0}| \geq \zeta\right).$$

Then, according to Lemma 3.1 and Lemma 3.2 (adjusting the confidence level of Lemma 3.1 and Lemma 3.2 from δ to $\frac{\delta}{2}$), we have

$$\Pr\left(\frac{q}{p+q}|L_{S-S} - E^{r=0}| \geq \frac{q}{p+q}\frac{b}{\sqrt{q}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}\right) \leq \frac{\delta}{2}, \text{ and}$$

$$\Pr\left(\frac{p}{p+q}|L_{D-S} - E^{r=0}| \geq \frac{p}{p+q}\frac{a}{\sqrt{p}}\sqrt{\frac{1}{2}\ln\left(\frac{4}{\delta}\right)}\right) \leq \frac{\delta}{2}.$$

Finally, according to the union bound: $\Pr(A_1 \cup A_2) \leq \Pr(A_1) + \Pr(A_2)$, we have

$$\Pr\left(\frac{p}{p+q}|L_{D-S} - E^{r=0}| + \frac{q}{p+q}|L_{S-S} - E^{r=0}| \geq \zeta\right) \leq \delta.$$

Overall, we have

$$\Pr(|L_T - E^{r=0}| \geq \zeta) \leq \Pr\left(\frac{p}{p+q}|L_{D-S} - E^{r=0}| + \frac{q}{p+q}|L_{S-S} - E^{r=0}| \geq \zeta\right) \leq \delta.$$

Q.E.D.

Appendix 2.2 Supplemented Results for Figure 3-5

This sub-section presents (1) results referred to by Footnote 30, in which slightly increasing number of same-distribution source data – at least 1.1 times of the number of predictors plus ten, are used as the same-distribution source data records (Figure A-4); (2) tabulated results of Figure 3-5 (Table A-7), (3) summary statistics for each method used in Figure 3-5 (Table A-8).

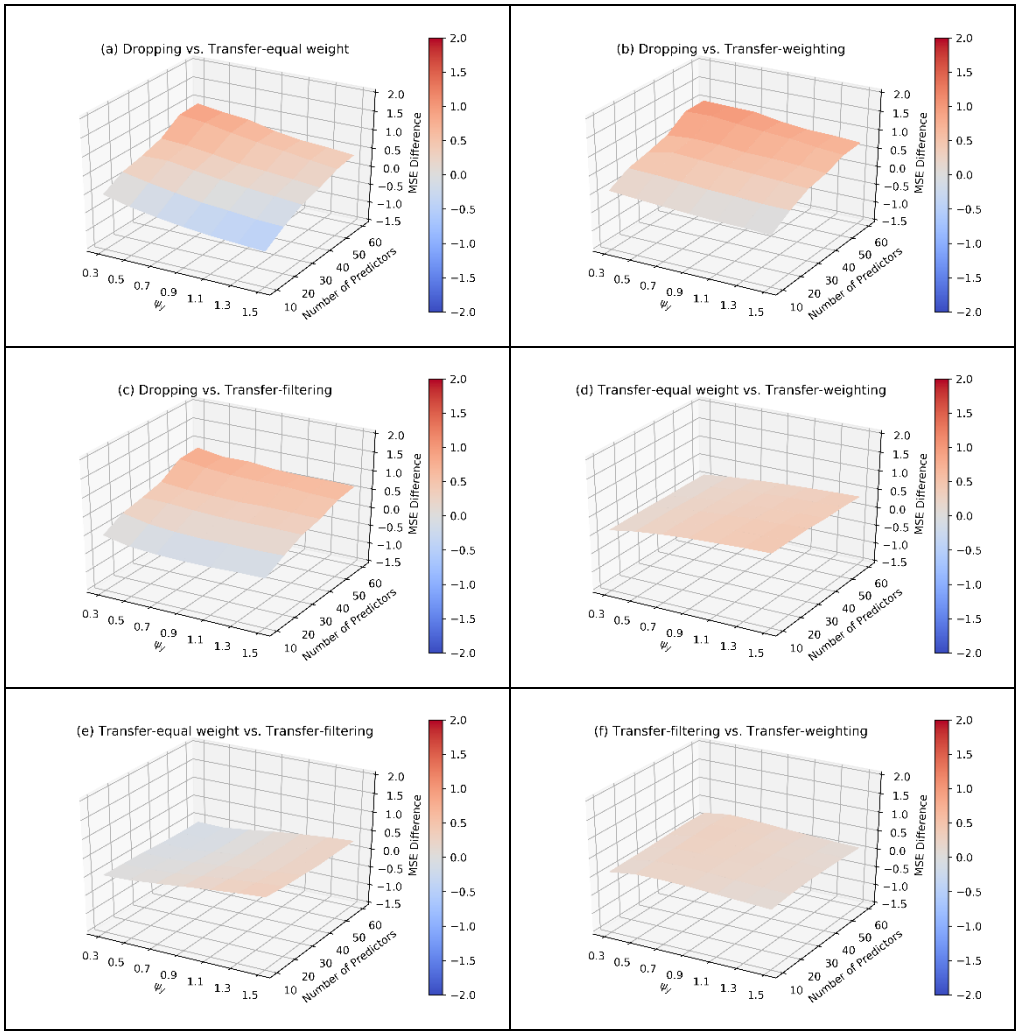


Figure A-4 Pairwise Comparison Using More Same-Distribution Source Data

		Number of Predictors					
	ψ_y	10	20	30	40	50	60
Figure 3-5(a)	0.3	-0.0419 (0.00)	0.2722 (0.00)	0.4749 (0.00)	0.7874 (0.00)	1.2229 (0.00)	1.4902 (0.00)
	0.5	-0.2172 (0.00)	0.1436 (0.00)	0.5014 (0.00)	0.7967 (0.00)	1.1713 (0.00)	1.557 (0.00)
	0.7	-0.339 (0.00)	0.0353 (0.0367)	0.4033 (0.00)	0.7132 (0.00)	1.1172 (0.00)	1.4346 (0.00)
	0.9	-0.4365 (0.00)	-0.0686 (0.0001)	0.2992 (0.00)	0.6276 (0.00)	1.0443 (0.00)	1.265 (0.00)
	1.1	-0.509 (0.00)	-0.1218 (0.00)	0.2288 (0.00)	0.5704 (0.00)	0.8815 (0.00)	1.1028 (0.00)
	1.3	-0.5537 (0.00)	-0.1772 (0.00)	0.1738 (0.00)	0.4915 (0.00)	0.7854 (0.00)	1.0219 (0.00)
	1.5	-0.6014 (0.00)	-0.2199 (0.00)	0.0924 (0.00)	0.4237 (0.00)	0.6916 (0.00)	0.8838 (0.00)
Figure 3-5(b)	0.3	0.0582 (0.00)	0.3739 (0.00)	0.5767 (0.00)	0.8765 (0.00)	1.3151 (0.00)	1.5701 (0.00)
	0.5	0.0045 (0.3606)	0.3617 (0.00)	0.7136 (0.00)	0.9851 (0.00)	1.354 (0.00)	1.7164 (0.00)
	0.7	-0.0405 (0.00)	0.333 (0.00)	0.6858 (0.00)	0.9759 (0.00)	1.3623 (0.00)	1.6515 (0.00)
	0.9	-0.0815 (0.00)	0.2848 (0.00)	0.6372 (0.00)	0.9303 (0.00)	1.3365 (0.00)	1.525 (0.00)
	1.1	-0.1185 (0.00)	0.2655 (0.00)	0.6079 (0.00)	0.9134 (0.00)	1.2167 (0.00)	1.3969 (0.00)
	1.3	-0.1296 (0.00)	0.2413 (0.00)	0.5781 (0.00)	0.8632 (0.00)	1.1314 (0.00)	1.3422 (0.00)
	1.5	-0.1606 (0.00)	0.2182 (0.00)	0.5234 (0.00)	0.8114 (0.00)	1.0698 (0.00)	1.2256 (0.00)
Figure 3-5(c)	0.3	-0.036 (0.00)	0.2646 (0.00)	0.4491 (0.00)	0.7148 (0.00)	1.1271 (0.00)	1.3879 (0.00)
	0.5	-0.1839 (0.00)	0.1346 (0.00)	0.4515 (0.00)	0.6893 (0.00)	1.042 (0.00)	1.4285 (0.00)
	0.7	-0.2369 (0.00)	0.0786 (0.00)	0.4172 (0.00)	0.6731 (0.00)	1.0616 (0.00)	1.3887 (0.00)
	0.9	-0.2566 (0.00)	0.0539 (0.0047)	0.3781 (0.00)	0.6763 (0.00)	1.083 (0.00)	1.3116 (0.00)
	1.1	-0.2521 (0.00)	0.0821 (0.00)	0.3835 (0.00)	0.6981 (0.00)	1.0088 (0.00)	1.2336 (0.00)
	1.3	-0.2267 (0.00)	0.0897 (0.00)	0.3961 (0.00)	0.6788 (0.00)	0.979 (0.00)	1.2194 (0.00)
	1.5	-0.2189 (0.00)	0.0997 (0.00)	0.3772 (0.00)	0.6666 (0.00)	0.9473 (0.00)	1.1322 (0.00)

Table A-7 Tabulated Results of Figure 3-5

(Continue on Next Page)

		Number of Predictors					
	ψ_y	**10**	**20**	**30**	**40**	**50**	**60**
Figure 3-5(d)	**0.3**	0.1001 (0.00)	0.1017 (0.00)	0.1017 (0.00)	0.0891 (0.00)	0.0922 (0.00)	0.0799 (0.00)
	0.5	0.2216 (0.00)	0.2181 (0.00)	0.2122 (0.00)	0.1884 (0.00)	0.1827 (0.00)	0.1594 (0.00)
	0.7	0.2985 (0.00)	0.2977 (0.00)	0.2824 (0.00)	0.2627 (0.00)	0.2451 (0.00)	0.2169 (0.00)
	0.9	0.355 (0.00)	0.3534 (0.00)	0.338 (0.00)	0.3027 (0.00)	0.2921 (0.00)	0.26 (0.00)
	1.1	0.3905 (0.00)	0.3873 (0.00)	0.3791 (0.00)	0.3431 (0.00)	0.3352 (0.00)	0.294 (0.00)
	1.3	0.4241 (0.00)	0.4184 (0.00)	0.4043 (0.00)	0.3717 (0.00)	0.346 (0.00)	0.3203 (0.00)
	1.5	0.4407 (0.00)	0.438 (0.00)	0.431 (0.00)	0.3877 (0.00)	0.3782 (0.00)	0.3418 (0.00)
Figure 3-5(e)	**0.3**	0.0059 (0.3735)	-0.0076 (0.1619)	-0.0259 (0.00)	-0.0726 (0.00)	-0.0959 (0.00)	-0.1023 (0.00)
	0.5	0.0333 (0.0001)	-0.009 (0.2964)	-0.0499 (0.00)	-0.1073 (0.00)	-0.1293 (0.00)	-0.1286 (0.00)
	0.7	0.1021 (0.00)	0.0433 (0.00)	0.0139 (0.1067)	-0.04 (0.00)	-0.0556 (0.00)	-0.0459 (0.00)
	0.9	0.1799 (0.00)	0.1225 (0.00)	0.0789 (0.00)	0.0487 (0.00)	0.0387 (0.00)	0.0466 (0.00)
	1.1	0.2569 (0.00)	0.2039 (0.00)	0.1548 (0.00)	0.1277 (0.00)	0.1273 (0.00)	0.1307 (0.00)
	1.3	0.327 (0.00)	0.2668 (0.00)	0.2223 (0.00)	0.1872 (0.00)	0.1936 (0.00)	0.1975 (0.00)
	1.5	0.3824 (0.00)	0.3196 (0.00)	0.2848 (0.00)	0.2429 (0.00)	0.2557 (0.00)	0.2483 (0.00)
Figure 3-5(f)	**0.3**	0.0942 (0.00)	0.1093 (0.00)	0.1276 (0.00)	0.1617 (0.00)	0.188 (0.00)	0.1822 (0.00)
	0.5	0.1883 (0.00)	0.2271 (0.00)	0.2621 (0.00)	0.2958 (0.00)	0.312 (0.00)	0.2879 (0.00)
	0.7	0.1964 (0.00)	0.2544 (0.00)	0.2686 (0.00)	0.3028 (0.00)	0.3007 (0.00)	0.2627 (0.00)
	0.9	0.1751 (0.00)	0.2309 (0.00)	0.259 (0.00)	0.254 (0.00)	0.2535 (0.00)	0.2134 (0.00)
	1.1	0.1336 (0.00)	0.1834 (0.00)	0.2243 (0.00)	0.2154 (0.00)	0.2079 (0.00)	0.1633 (0.00)
	1.3	0.0971 (0.00)	0.1516 (0.00)	0.182 (0.00)	0.1844 (0.00)	0.1524 (0.00)	0.1228 (0.00)
	1.5	0.0583 (0.00)	0.1185 (0.00)	0.1462 (0.00)	0.1448 (0.00)	0.1225 (0.00)	0.0934 (0.00)

Table A-7 Tabulated Results of Figure 3-5

(Continue from Previous Page)

Note: Rows in each panel indicate different values of ψ_y which are taken from {0.3, 0.5, 0.7, …, 1.5}. The columns indicate different number of predictors which are taken from {10, 20, …, 50, 60}. Numbers in parentheses are the p-values for conducting the T-test for comparing MSE of the respective two approaches.

		Number of Predictors					
	ψ_y	10	20	30	40	50	60
Dropping	0.3	0.5279 (0.0839)	0.846 (0.4910)	1.0546 (0.6397)	1.3557 (0.9621)	1.8091 (1.0317)	2.0594 (1.0509)
	0.5	0.5125 (0.0896)	0.8769 (0.4093)	1.242 (0.6528)	1.521 (0.7528)	1.9157 (0.9157)	2.2806 (1.3921)
	0.7	0.5124 (0.0975)	0.89 (0.3755)	1.2656 (0.6714)	1.5551 (0.7256)	1.9814 (0.9207)	2.2739 (0.9993)
	0.9	0.5035 (0.0952)	0.8733 (0.3776)	1.2507 (0.5495)	1.5546 (0.6804)	1.9949 (0.9447)	2.1874 (0.9652)
	1.1	0.4944 (0.0930)	0.8817 (0.3931)	1.2423 (0.5295)	1.5566 (0.7185)	1.8924 (0.7333)	2.0841 (0.9378)
	1.3	0.4956 (0.0992)	0.8715 (0.3952)	1.2346 (0.4901)	1.5213 (0.6291)	1.8419 (0.7710)	2.0465 (0.9316)
	1.5	0.4827 (0.0954)	0.8624 (0.3935)	1.1864 (0.4636)	1.4861 (0.6260)	1.7812 (0.7248)	1.9408 (0.7869)
Transfer - equal weight	0.3	0.5698 (0.0298)	0.5737 (0.0295)	0.5797 (0.0290)	0.5682 (0.0304)	0.5862 (0.0306)	0.5693 (0.0282)
	0.5	0.7297 (0.0350)	0.7332 (0.0365)	0.7406 (0.0334)	0.7243 (0.0359)	0.7445 (0.0363)	0.7236 (0.0325)
	0.7	0.8514 (0.0380)	0.8547 (0.0403)	0.8623 (0.0393)	0.8419 (0.0387)	0.8642 (0.0407)	0.8392 (0.0351)
	0.9	0.9399 (0.0392)	0.9419 (0.0410)	0.9515 (0.0408)	0.927 (0.0410)	0.9505 (0.0420)	0.9224 (0.0365)
	1.1	1.0033 (0.0402)	1.0034 (0.0418)	1.0136 (0.0400)	0.9862 (0.0403)	1.0109 (0.0430)	0.9812 (0.0374)
	1.3	1.0493 (0.0409)	1.0487 (0.0423)	1.0608 (0.0401)	1.0297 (0.0403)	1.0565 (0.0438)	1.0246 (0.0373)
	1.5	1.0841 (0.0404)	1.0823 (0.0421)	1.094 (0.0407)	1.0623 (0.0418)	1.0896 (0.0440)	1.0569 (0.0377)
Transfer - weighting	0.3	0.5639 (0.1339)	0.5813 (0.1175)	0.6056 (0.1160)	0.6409 (0.1328)	0.682 (0.1219)	0.6715 (0.1236)
	0.5	0.6964 (0.1804)	0.7422 (0.1892)	0.7905 (0.1804)	0.8316 (0.1715)	0.8738 (0.1522)	0.8521 (0.1314)
	0.7	0.7493 (0.1907)	0.8114 (0.2123)	0.8484 (0.1884)	0.882 (0.1574)	0.9198 (0.1499)	0.8851 (0.1281)
	0.9	0.7601 (0.1908)	0.8193 (0.1936)	0.8725 (0.1705)	0.8783 (0.1564)	0.9119 (0.1386)	0.8758 (0.1198)
	1.1	0.7464 (0.1844)	0.7996 (0.1798)	0.8588 (0.1564)	0.8585 (0.1428)	0.8836 (0.1279)	0.8505 (0.1151)
	1.3	0.7223 (0.1729)	0.7819 (0.1678)	0.8385 (0.1394)	0.8425 (0.1341)	0.8629 (0.1173)	0.8271 (0.0996)
	1.5	0.7017 (0.1599)	0.7627 (0.1523)	0.8092 (0.1307)	0.8194 (0.1297)	0.8339 (0.1107)	0.8086 (0.0954)
Transfer - filtering	0.3	0.4697 (0.0396)	0.472 (0.0438)	0.478 (0.0343)	0.4791 (0.0610)	0.494 (0.0562)	0.4893 (0.0672)
	0.5	0.5081 (0.0578)	0.5151 (0.0705)	0.5284 (0.0643)	0.5359 (0.0904)	0.5618 (0.0973)	0.5642 (0.1243)
	0.7	0.5529 (0.0820)	0.557 (0.0768)	0.5798 (0.0803)	0.5792 (0.0848)	0.6191 (0.1165)	0.6224 (0.1255)
	0.9	0.585 (0.0721)	0.5884 (0.0728)	0.6135 (0.0853)	0.6242 (0.0992)	0.6584 (0.1039)	0.6624 (0.1239)
	1.1	0.6128 (0.0823)	0.6162 (0.0794)	0.6345 (0.0799)	0.6432 (0.0909)	0.6757 (0.0983)	0.6872 (0.1174)
	1.3	0.6252 (0.0829)	0.6303 (0.0744)	0.6564 (0.0798)	0.6581 (0.0820)	0.7105 (0.1176)	0.7043 (0.1064)
	1.5	0.6434 (0.0887)	0.6442 (0.0815)	0.6631 (0.0739)	0.6746 (0.0878)	0.7114 (0.1010)	0.7152 (0.1042)

Table A-8 Summary Statistics for Each Method Used in Figure 3-5

Appendix 2.3 Tabulated Results of Figure 3-6

		Number of Predictors					
	ψ_y	**10**	**20**	**30**	**40**	**50**	**60**
Figure 3-6(a) Bias^2	**0.3**	-0.1698 (0.00)	-0.167 (0.00)	-0.1676 (0.00)	-0.1641 (0.00)	-0.1658 (0.00)	-0.1584 (0.00)
	0.5	-0.3587 (0.00)	-0.3559 (0.00)	-0.3503 (0.00)	-0.3416 (0.00)	-0.3506 (0.00)	-0.3358 (0.00)
	0.7	-0.4975 (0.00)	-0.4923 (0.00)	-0.4868 (0.00)	-0.4801 (0.00)	-0.4867 (0.00)	-0.4703 (0.00)
	0.9	-0.6007 (0.00)	-0.5942 (0.00)	-0.5938 (0.00)	-0.5731 (0.00)	-0.5859 (0.00)	-0.5697 (0.00)
	1.1	-0.6739 (0.00)	-0.6671 (0.00)	-0.6651 (0.00)	-0.6428 (0.00)	-0.6571 (0.00)	-0.64 (0.00)
	1.3	-0.7291 (0.00)	-0.7199 (0.00)	-0.7193 (0.00)	-0.6946 (0.00)	-0.7104 (0.00)	-0.6933 (0.00)
	1.5	-0.7725 (0.00)	-0.7581 (0.00)	-0.7632 (0.00)	-0.7372 (0.00)	-0.7536 (0.00)	-0.7334 (0.00)
Figure 3-6(b) Variance	**0.3**	0.128 (0.00)	0.4392 (0.00)	0.6425 (0.00)	0.9515 (0.00)	1.3888 (0.00)	1.6485 (0.00)
	0.5	0.1415 (0.00)	0.4996 (0.00)	0.8517 (0.00)	1.1383 (0.00)	1.5218 (0.00)	1.8928 (0.00)
	0.7	0.1586 (0.00)	0.5276 (0.00)	0.8901 (0.00)	1.1933 (0.00)	1.6039 (0.00)	1.9049 (0.00)
	0.9	0.1643 (0.00)	0.5256 (0.00)	0.893 (0.00)	1.2007 (0.00)	1.6302 (0.00)	1.8347 (0.00)
	1.1	0.1649 (0.00)	0.5453 (0.00)	0.8939 (0.00)	1.2132 (0.00)	1.5386 (0.00)	1.7429 (0.00)
	1.3	0.1754 (0.00)	0.5427 (0.00)	0.8931 (0.00)	1.1862 (0.00)	1.4958 (0.00)	1.7152 (0.00)
	1.5	0.1712 (0.00)	0.5383 (0.00)	0.8556 (0.00)	1.1609 (0.00)	1.4452 (0.00)	1.6173 (0.00)

Table A-9 Tabulated Results of Figure 3-6

Note: Rows in each panel indicate different values of ψ_y which are taken from {0.3, 0.5, 0.7, …, 1.5}. The columns indicate different number of predictors which are taken from {10, 20, …, 50, 60}. Numbers in parentheses are the *p*-values for conducting the *t*-test for comparing Bias^2/Variance of the respective two approaches.

Appendix 2.4 Full Results of Figure 3-7

Figure A-5 Prediction Performance and the Timing of Adjusting the Prediction Model (Full Results)