Judgment Post-Stratification with Machine Learning Techniques: Adjusting for Missing Data in Surveys and Data Mining

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Tian Chen, M.S.

Graduate Program in Statistics

The Ohio State University

2013

Dissertation Committee:

Tao Shi, Ph.D., Co-Advisor
Elizabeth. A. Stasny, Ph.D., Co-Advisor
Omer Ozturk, Ph.D.
Abstract

Missing data is found in every type of data collection. How to deal with missing data has long been discussed in the survey sampling literature. It has not, however, been the topic of much research involving the huge data sets common in the data mining setting. In this dissertation, we combine ideas from the survey sampling and data mining literature to develop methods for handling missing data in both contexts.

Judgement Post-Stratification (JPS) is a data analysis method, motivated by ranked set sampling (RSS), that uses judgement ranking for post-stratification. This dissertation briefly introduces RSS and JPS. Then it connects the JPS method with machine learning (ML) techniques in two ways. One is to use the ML techniques to build a ranking function therefore solving the judgement ranking problem. The other is to compare the estimates from the JPS method with these well-known ML techniques and provide efficiency measurements for the JPS method.

We investigate the effect of set size, the number of units ranked at one time, through simulation studies. We also consider possible extensions for JPS, such as proportional proration. To our knowledge, we provide the first systematic study of the influences of three types of missing data on various ML techniques using simulated data. Finally, two real life examples are used to demonstrate the application of the JPS method in real world problems.
This is dedicated to my parents, Hua and Guolin, and my grandma, Yimin, for their love and endless support.
I’m extremely thankful to have Professors Elizabeth A. Stasny and Tao Shi as my co-advisers, for their patience, mentorship and encouragement. Without their guidance and help, this dissertation would not have been possible and my research journey would not have been this great.

I would also like to thank my committee member, Professor Omer Ozturk, who organizes the Ranked Set Sampling reading group that I attendeded and from which I got many ideas and inspiration.

It has been a wonderful five years in the Statistics Department at The Ohio State University. It was my great pleasure to study here and I would like to express my deepest appreciation to all of my professors and to my colleagues Erin Leatherman and Katie Thompson.
Vita

June 2004-June 2008 .................. B.S. Statistics,
The Renmin University of China,
China
September 2008-September 2010 .......... M.S. Statistics,
The Ohio State University, USA
September 2010-present ............... Graduate Teaching Associate,
The Ohio State University, USA

Fields of Study

Major Field: Statistics
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Vita</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>x</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Organization of this Thesis</td>
<td>4</td>
</tr>
<tr>
<td>2. Background and Literature Review</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Missing Data</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1 Missing Data Mechanisms</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2 Missing Data Analysis Methods</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Supervised and Semi-supervised Learning Methods</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1 Review of Standard Supervised Learning Methods</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Review of Standard Semi-Supervised Learning Methods</td>
<td>18</td>
</tr>
<tr>
<td>3. Ranked Set Sampling and Judgement Post-Stratification</td>
<td>21</td>
</tr>
<tr>
<td>3.1 Motivation</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Judgement Ranking</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Ranked Set Sampling</td>
<td>26</td>
</tr>
</tbody>
</table>
3.4 Judgement Post-Stratification ........................................... 28
3.5 Connections and Differences between JPS and RSS .................... 31

4. Simulation Study for the JPS Method under Three Missing Data Scenarios 33
   4.1 Univariate Simulation Study ........................................... 33
      4.1.1 Data Generation .................................................. 34
      4.1.2 Results of Simulation Study ..................................... 41
      4.1.3 The Effects of Sample Size and Set Size ....................... 43
      4.1.4 Proportional Proration .......................................... 47
      4.1.5 Comparison of JPS with Expectation Maximization .......... 50
   4.2 Bivariate Simulation Study ........................................... 54
      4.2.1 MCAR ............................................................. 56
      4.2.2 MAR ............................................................. 60
      4.2.3 MNAR ............................................................ 61
   4.3 A More Complicated Example ......................................... 62

5. JPS Applications in Two Examples ........................................ 70
   5.1 The EHR Survey ...................................................... 75
   5.2 Satellite Data ........................................................ 78

6. Contributions and Future Work ............................................ 89
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>JPS estimator compared to SRS estimator under different missing data mechanisms</td>
</tr>
<tr>
<td>4.2</td>
<td>JPS estimator compared to SRS estimator for different sample size and set size combinations</td>
</tr>
<tr>
<td>4.3</td>
<td>JPS and JPS(proportional proration) estimators compared to SRS estimator under MCAR assumption</td>
</tr>
<tr>
<td>4.4</td>
<td>JPS and JPS(proportional proration) estimators compared to SRS estimator under MAR assumption</td>
</tr>
<tr>
<td>4.5</td>
<td>JPS and JPS(proportional proration) estimators compared to SRS estimator under MNAR assumption</td>
</tr>
<tr>
<td>4.6</td>
<td>Comparison of estimators using simulated data under MCAR assumption: sample size $n=100$, set size $m=3$, iteration $k=5,000$</td>
</tr>
<tr>
<td>4.7</td>
<td>Comparison of estimators using simulated data under MCAR assumption: sample size $n=100$, set size $m=3$ and $m=6$, iteration $k=5,000$</td>
</tr>
<tr>
<td>4.8</td>
<td>Comparison of estimators using simulated data under MAR assumption: sample size $n=100$, set size $m=3$, iteration $k=5,000$</td>
</tr>
<tr>
<td>4.9</td>
<td>Comparison of estimators using simulated data under MNAR assumption: sample size $n=100$, set size $m=3$, iteration $k=2,000$</td>
</tr>
<tr>
<td>4.10</td>
<td>Comparison of estimators for book example data under MCAR assumption: sample size $n=100$, set size $m=3$, iteration $k=5,000$</td>
</tr>
</tbody>
</table>
5.1 Data summary for explanatory variables in HIT survey . . . . . . . 72

5.2 Comparison of estimators for satellite matched data under MCAR assumption: sample size $n = 200$, set size $m = 3$, iteration $k = 5,000$ . . 82

5.3 Comparison of estimators for satellite matched data under MNAR assumption: sample size $n = 200$, set size $m = 3$, iteration $k = 2,000$ (excludes all 0 and all 1 samples) . . . . . . . . . . . . . . . . . . 86
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Histograms of the population (in blue) and the learning dataset (in red) under the MCAR assumption</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Histograms of the population (in blue) and the learning dataset (in red) under the MAR assumption</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>Histograms of the population (in blue) and the learning dataset (in red) under the MNAR assumption</td>
<td>40</td>
</tr>
<tr>
<td>4.4</td>
<td>The change in MSE for set sizes ranging from 3 to 15; sample size is 100</td>
<td>46</td>
</tr>
<tr>
<td>4.5</td>
<td>MSE comparison for JPS, SRS and EM under MCAR assumption</td>
<td>53</td>
</tr>
<tr>
<td>4.6</td>
<td>MSE comparison for JPS, SRS and EM under MAR assumption</td>
<td>54</td>
</tr>
<tr>
<td>5.1</td>
<td>The A-train formation</td>
<td>74</td>
</tr>
<tr>
<td>5.2</td>
<td>Image of one MODIS data collection over the Pacific Ocean. The red line in the image indicates the CloudSat/CALIPSO path overlaid on MODIS data.</td>
<td>76</td>
</tr>
<tr>
<td>5.3</td>
<td>Boxplot of Estimation Error (MCAR) for SRS, JPS, RF, SVM and 5NN, sample size $n = 100$, set size $m = 3$, iteration $k = 5,000$</td>
<td>83</td>
</tr>
<tr>
<td>5.4</td>
<td>Boxplot of Estimation Error (Systematic Sampling) for SRS, JPS, RF, SVM and 5NN, sample size $n = 100$, set size $m = 3$, iteration $k = 5,000$</td>
<td>87</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

1.1 Overview

Our ability to obtain and generate data has grown extensively over recent years. The explosive expansion in the amount of available data also brings the same problem we have long dealt with in the survey sampling literature: missing data. The question is how to adjust estimates to adjust for the bias that typically results from differential nonresponse. In data mining settings, however, most practitioners ignore the cases with missing data because of the huge amount of available data. This ignores the caution we give to most introductory statistics students that if there is bias in the data collection process, taking a larger sample merely repeats the bias on a larger scale. How to better understand the missing data mechanism and its influence on the estimation is the research topic we explore in this dissertation. In the problem of estimating the population mean (proportion) of a variable at interest ($Y$) with a small fraction of a dataset being missing, the common practices are either ignoring the missing part and only analyzing the observed data, imputing values for the missing data, or reweighting the observed data to adjust for the missing data.
Judgment Post-Stratification (JPS), is a method that may be used to weight the observed data to account for the missing data problem. Similar to Ranked Set Sampling (RSS), it uses ranking of the unmeasured characteristic of interest according to auxiliary information on one or more related variables to provide more structure to the sample. The traditional ways of ranking include adopting an expert’s opinion or through use of concomitant variables. In this thesis, I present my research on JPS for estimating a population proportion under the situation that univariate or multivariate concomitant variables are observed. I also investigate the applications of the JPS method and its comparisons with other standard data mining methods under three standard missing data scenarios.

With modern data collection techniques, we often face large datasets with many variables, collected without the use of formal sampling designs. In situations where a large proportion of the data is missing on the variable of interest, handling missing data carefully is important for reducing bias in estimates. For example, the Ohio Electronic Health Records Survey conducted in 2010 to estimate the adoption rate of Electronic Health Records systems by health providers, used a simple random sampling design but a response rate lower than 20% made design-based analysis suspect. Exploratory data analysis suggested that the response group and the nonresponse group were significantly different on variables available in the sampling frame. Another example from a much different setting involves cloud fraction estimation from satellite data. The highly accurate cloud indicator data based on lidar and radar (on CALIPSO and CloudSat) only cuts through a small portion of the area of interest in a thin line, but additional radiation data in 36 different wavelengths are available for
a whole section of the earth from another remote sensing instrument (MODIS). The challenge in these problems is that the number of responses is relatively small and responses do not represent the whole population very well. Under these situations, JPS provides us with a tool to improve the estimation of the population proportion for a variable compared to estimates using straight sample proportions.

We believe JPS is naturally connected with a semi-supervised learning setup from the machine learning literature. Semi-supervised learning algorithms use both the observed (“labeled” in machine learning terminology) data and the unobserved (“unlabeled”) data to build a better classifier for classification problems. The semi-supervised learning algorithms are often applied in the situation where only a small proportion of the labels are observed because measuring units is expensive or time consuming. Meanwhile, there may be other variables related to the unobserved data that are cheap to obtain and might be helpful in estimation. This type of missing data is very common, as we noted in the previous examples. In such cases, we found that combining JPS procedures and the ideas from semi-supervised learning from the machine learning literature has a broad range of applications in these missing data problems. Moreover, various supervised learning methods can be applied to the observed data to obtain estimates. We can also implement JPS on top of these standard supervised learning methods to try to improve estimation.

The idea behind JPS is to let the observed units better represent the unobserved units by dividing them into ordered strata according to related available data. For example, if the sample contains more observed units from strata corresponding to
smaller order statistics, this suggests there are more unobserved units in the larger order statistics strata so that such units are under represented. In this case, JPS redistributes the weights on observed units to give those observed units that fall in the strata corresponding to higher order statistics larger weights. In this way, the unobserved units are better accounted for in the estimation.

1.2 Organization of this Thesis

The rest of this thesis is organized as follows.

Chapter 2 briefly introduces standard missing data mechanisms and machine learning techniques. Missing data is a very common problem during the data collection process. The missing observations in the data may cause bias in estimates as well as an increase in variance from the loss of information. When we try to learn from data with missingness, depending on what type of missingness is present, we may want to use different methods for estimation and watch for the influence of missing data. We review three different types of missing data mechanisms in this chapter.

Among many methods researchers developed to explore patterns and gain information from data, machine learning methods have been widely applied. Based on the goal and available observations in the datasets, there are three standard types of machine learning techniques. Usually our goal is to classify a new case. For this classification purpose, if we only use the labeled data, then the method is called supervised learning. If we use both labeled and unlabeled data, it is called semi-supervised learning. Both methods require some labeled data to work. In contrast, if the only
available data is unlabeled and the goal is clustering, then the method is defined as unsupervised learning. There is a gap, however, between the missing data literature and all these machine learning techniques. Previous researches rarely discuss the influence of missing data when applying different machine learning techniques. As is well known from the sampling survey literature, when the missingness is completely at random, it is ignorable. But there are others cases in which the missingness is not ignorable. So we want to explore in more detail how different estimation methods perform under various missing data scenarios.

Chapter 3 discusses RSS and JPS methods. RSS, as the name says, is a sampling method. When we take a random sample, we rely on the idea that “on average” our estimates from such samples will be unbiased and we hope to get a representative sample of the population. In reality, this does not always work. By chance, we may get a sample that contains extreme cases and, therefore, is not a good representation of the whole population. Meanwhile, we may have much auxiliary information to help us to decide the approximate size, or rank, of a unit in the whole population. So naturally, we want to use this available supplementary information to assist us in picking a better sample. That is the motivation for RSS. Similarly, JPS also uses auxiliary information, but it is not a sampling method. It has wider application since it does not require a special sampling design. When MacEachern, Stasny and Wolfe [13] first proposed JPS, they envisioned that the judgement ranking would be done by experts or visual inspection. In this thesis, we combine machine learning and modeling ideas in the ranking process. Thus for those difficult ranking situations with many explanatory variables, by using machine learning techniques or building
a model, we can still get the rankings needed by JPS and carry out the JPS procedure.

Chapter 4 describes the details of our simulation study for both the univariate case and bivariate case. As we mentioned, we want to build the connection between missing data and machine learning techniques, where we view the JPS method as a special form of the semi-supervised learning methods. To study how our methods perform under different missing data scenarios, we simulate two datasets based on mixture Gaussian distributions: one univariate case and one bivariate case, each with three types of missingness. By comparing the results from our various proposed methods under these three scenarios, we gain a better understanding of the relationship between the performance of different methods and the type of missingness. In addition to the simple mixture Gaussian case, we are also interested in the methods’ performance when the model assumption from this case is not met. We use a generated data example [10] from the machine learning literature to illustrate this situation.

We apply JPS to two real world applications in Chapter 5. One is the Electronic Health Records (EHR) survey data example. The goal for this example is to estimate the proportion of all health care providers who have an EHR system. Initial exploratory data analysis on this survey data shows that the response rate is very low and the response group and nonresponse group have substantial differences. We demonstrate the JPS method on this dataset and show the advantage of it in reducing the bias in the original sample estimate. The second example is the satellite data example, where our goal is to estimate the cloud cover proportion in the whole
MODIS image. When we aggregate the data from two different satellites, due to the equipment limitations, only a line in the whole MODIS image has the cloud proportion labeled. This creates a natural missing data scenario. Again, we mimic the missingness in this dataset and show that JPS is a desirable method to use here.

Chapter 6 summarizes the results of the thesis, and discusses the potential directions for future research based on this current work.
Chapter 2: Background and Literature Review

2.1 Missing Data

Missing data is a very common problem researchers have to face during the data collection and analysis process. Missing data, at a minimum, reduces the sample size and therefore, increases variance. In addition, differential missingness may lead to biased estimates. The optimal way to handle this is to avoid the missing data by trying to collect the data as completely as possible. In many cases, however, researchers have little or no control over nonresponse in survey data or the amount of labeled data in the data mining setting. Therefore, we focus on analyzing the reasons for missing data and on how to handle the missingness.

Missing data can have various possible explanations. It might be due to the limitation of the data frame. For example, in a survey to estimate average annual income where the phone book is used as a sampling frame, those who have changed their numbers but have not yet had their numbers updated in the phone book have no chance of being selected to participate in the survey. Similar examples could be that there are people with unlisted numbers or people who only have cell phones. This kind of missingness does not always provide us useful information about the data. In
other cases, however, missingness can contain important information about the data. For example, in the annual income survey example, some sampled units may refuse to cooperate. One possible explanation is that those who make much more than the others simply do not want to answer the question and reveal their income. Thus when we observe the presence of missing data, it is dangerous to assume all the missingness is unrelated to the variable of major interest.

2.1.1 Missing Data Mechanisms

There are three different types of missing mechanisms commonly used in the survey sampling literature [12]. Missing Completely At Random (MCAR) is the best-case scenario, where missingness is not related to the variable of interest $y$ or to auxiliary information $x$. If the data are MCAR, then missingness may be ignored for estimation purposes and standard statistical analyses may be used. The second scenario is Missing At Random (MAR), the most common assumption for many imputation methods such as multiple imputation [20]. Under the MAR assumption, the probability that a sampled unit is missing is only related to auxiliary information $x$ but not the variable of interest $y$. For example, in an income survey, we often get more responses from an urban area than from the suburbs. But if we look at the income for two areas separately, there may be no significant difference in the probability of getting nonresponses from the high income group and the low income group in these two areas. In this case, the missingness of the variable of interest $y$ varies across the population but is random in the subgroup based on observed $x$ values. The third type of missingness is Missing Not At Random (MNAR). Under a MNAR mechanism, the
probability of missing the variable of interest $y$ depends on the variable’s value. As in the average household income problem mentioned earlier, high income families may be less likely to report their incomes. Thus missingness of the income information depends on the income itself.

Under the MAR and MNAR scenarios, an SRS estimator would be biased on average. The MAR case can be considered to be ignorable if one has all the $x$ information that explains the differential nonresponse under the correct model assumption. Many imputation methods incorporate the MAR assumption in their algorithms. MNAR nonresponse is nonignorable and is the hardest to deal with analytically [12]. One needs to correctly model the underlying missing data mechanism that generates the data to handle MNAR appropriately. This is difficult to achieve because the variable of interest $y$ is correlated with the missingness.

### 2.1.2 Missing Data Analysis Methods

We can divide missing data analysis methods into four categories. The first type of methods analyze only the observed data. Such methods include complete case analysis and available case analysis. The second type is to model the data and use likelihood-based methods such as an Expectation Maximization (EM) algorithm. The third method is imputation where a plausible value is substituted for the missing one. There are many methods for imputation, and either single imputation or multiple imputation may be used. Finally there are procedures that reweight the observed
cases to account for the missing cases.

The easiest way to deal with missing data, although certainly not the best, is complete case analysis. It deletes all the cases with missing data, leaving only the complete cases to be included in the analysis. There are many problems with this method. First, if we try to fit a model for which some of the explanatory variables have observations missing, such as a regression model, as we add or remove variables, the data set will change accordingly. If we delete all the cases with any explanatory variable missing, the remaining data set may be too small and it is also a waste of the available information. Second, the analysis based on complete cases assumes MCAR. The analysis would be reliable if the observed sample is in fact a random sample from the population. But if the missing data mechanism is not MCAR, then the inference based on complete cases is likely to be biased.

Available-case analysis also discards available data to make estimation easier. Compared with complete-case analysis, it uses all the available data to estimate the parameters. For the univariate case, it is basically the same as complete-case analysis. But when there is more than one explanatory variable, the procedure is different. For example, suppose there are \( m \) cases with values for \( X_1 \) and \( n \) cases with values for \( X_2 \), where \( m \neq n \). Then the \( m \) cases will be used to estimate the mean and variance for \( X_1 \). In addition, the \( n \) cases will be used to estimate the mean and variance for \( X_2 \). All the cases with complete \( X_1 \) and \( X_2 \) are used for the estimation of correlation of \( X_1 \) and \( X_2 \). Therefore, we use a different number of cases to estimate different parameters in the data. Little and Rubin [12] pointed out that when variables are
highly correlated, available case analysis may not provide consistent estimates, and
the correlation estimates may be outside of $[-1, 1]$ because of the different number of
observations used to estimated the parameters.

Expectation maximization (EM) is an iterative method based on maximum like-
lihood for datasets that are partially unlabeled [7]. The goal is to estimate the most
likely parameters based on the observed data. It has two steps: the E-step and the
M-step. Let us assume the observed data is contained in $x$ and the missing part is
$z$; the parameter of interest is $\theta$. For the E-step, we use the observed data and the
current estimated parameters to estimate the missing data: $E_{Z|X, \theta(t)}[\log L(\theta; X, Z)]$.
For the M-step, we assume the missing data are known and maximize the likelihood
function: $\arg\max_{\theta} Q(\theta|\theta(t))$. The estimates from the E-step are used in the M-step.
The EM algorithm has been shown to increase the likelihood function at each itera-
tion and to converge to a local maximum.

There are many missing data imputation methods. We will mainly introduce
two types here. One is called single imputation. It fills the missing value with a
single value. The mean value is perhaps the simplest version of single imputation.
When the mean value is used for single imputation, the overall mean is equal to the
complete case mean, however, the variance is underestimated [12]. An extension of
single imputation is multiple imputation [20]. Multiple imputation is a model-based
method, which often assumes the multivariate normal distribution. The main idea
is to average the outcomes from different imputed completed datasets reflecting the
uncertainty in imputations. The first step is to generate completed datasets using a
non-deterministic imputation procedure. The second step is to average the estimates from the different datasets. We take multinomial imputation as an example. To fill in the missing values, we draw values randomly from the multivariate normal distribution estimated by the observed data to get multiply-imputed completed datasets. Then perform standard data analysis procedures on these completed datasets to get multiple estimates. Averaging these estimates provides the final estimate. The variances between and within completed data sets are combined to provide a variance estimate that reflects the uncertainty in the imputation process.

Our method, Judgement Post-Stratification (JPS) can be viewed as the fourth type of missing data analysis method. It uses auxiliary information to derive a weight for every observed unit. Then we calculate the estimate based on the weighted observed units. The goal is to get a less biased estimate through the reweighting procedure. An advantage of the JPS method is it does not require any assumption about the distribution of the data. Also, it can handle situations when the missingness is nonignorable. We show more details about the JPS method and its performance in Chapters 4 and 5.

2.2 Supervised and Semi-supervised Learning Methods

Machine Learning (ML) has wide application in the data mining area. Based on the goal of the analysis and available information, we can divide ML techniques into three types depending on whether labeled data, unlabeled data, or both are being
used. If we only learn from cases with known labels, the ML techniques are called supervised learning. If the goal is to put the similar unlabeled cases in the same group, that is, to do clustering, then we are using unsupervised learning. In reality, in most situations we have both labeled and unlabeled data. If the method uses both labeled and unlabeled data, then we call it semi-supervised learning. In this subsection, we provide a brief overview of these ML techniques.

Supervised learning methods use labeled data to train the classifier. Therefore to use a supervised learning technique we start with a training set \((x_i, y_i)\) for \(i = 1, 2, ..., n\) where the observations are \(i.i.d.\) samples drawn from a joint distribution \(f(x, y)\). The variable \(x\) could be an explanatory variable or a vector of explanatory variables and \(y\) is the class label. Here we only consider the case of a binary \(y\) although the method may be extended to more classes. The goal is to estimate a mapping relationship \(x \rightarrow y\). Supervised learning tries to minimize the classification error. Labeled data, however, could be expensive or time consuming to obtain. Meanwhile, in many situations, unlabeled data are abundant or relatively easy to collect. For example, in the speech recognition problem, recording a large amount of speech does not cost much but if we want to know the content of the speech and put a label on the speech, it requires that a person listen to the speech and type it into a transcript.

A machine learning method which only uses unlabeled data to find the hidden structure in \(x\) is called unsupervised learning. Unsupervised learning studies a dataset with \(i.i.d.\) points \((x_1, x_2, ..., x_n)\). The goal is to discover the structure in \((x_1, x_2, ..., x_n)\), that is, to estimate \(f(x)\). Thus we can do clustering on the data and classify a new
Semi-supervised learning is between supervised and unsupervised learning in that it handles datasets with both labeled and unlabeled data. The labeled part of the data is denoted by \(((x_1, y_1), (x_2, y_2), ..., (x_n, y_n))\) where the labels \(y_i\) are known. In addition, the dataset also has an unlabeled part \((x_{n+1}, x_{n+2}, ..., x_{n+m})\) where the labels are unknown. Usually the amount of labeled data is small and the amount of unlabeled data is relatively large. Semi-supervised learning can be viewed in two ways based on the goal of the analysis. It could be a supervised learning method with extra information on \(x\), in which case it uses unlabeled data to modify the result from the labeled data. Alternatively it could be considered as unsupervised learning with constraints.

In this thesis, the problem we care about is classification and population proportion estimation, so we will focus on introducing supervised learning and semi-supervised learning methods here.

### 2.2.1 Review of Standard Supervised Learning Methods

**Linear Discriminate Analysis**

Linear Discriminate Analysis (LDA) was first proposed by Fisher [8] as a classification method. Its goal is to reduce the dimensionality of the data while preserving as much of the difference between classes as possible. In other words, we want to find a projection of the data onto the lower dimension that has the largest separation.
between classes. We perform the method for the two classes case in our simulation study. Fisher’s LDA, which does not assume normality or equal covariance, maximizes the ratio of the between-classes variance to the within-classes variance. More generally, LDA under the assumption of a normal distribution predicts points as being from a certain class if the log of the likelihood ratio is greater than some threshold. Both methods lead to a linear classification boundary. LDA can be generalized to multiple discriminant cases.

K-Nearest Neighbor (KNN)

The idea behind the nearest neighbor method is very straightforward. We classify points based on the class of their nearest neighbors. Take the k nearest neighbor case as an example. For any unlabeled point \( x \), look for its \( k \) nearest neighbor labeled points. By majority vote or distance-weighted vote, we can predict the label for this point. So the prediction has two stages. First according to the distance measure, find the \( k \) nearest neighbors. Second, use the labels of these \( k \) nearest neighbor points to predict a label for \( x \). In summary, KNN uses the relationship of similarity and distance between the points to do predictions.

The prediction using majority votes is

\[
\hat{Y}(x) = I\left(\frac{1}{k} \sum_{x_i \in N_k(x)} Y_i > 0.5\right)
\]  

(2.1)

where \( N_k(x) \) represents the neighborhood of \( x \) that includes the \( k \) nearest points in the training set using Euclidean distance.
Support Vector Machines

Support vector machines (SVM) was first developed by Vapnik [25]. The basic version of SVM takes training data and predicts the class label for each point. It constructs a set of hyperplanes in a high dimension or infinite space, which can be used for classification or regression. A good separation is achieved by the hyperplane that has the largest distance to the nearest training points of the other class. When separating the samples by the hyperplanes, the samples that lie on the margin closest to the hyperplanes are called the support vectors. Finding the solution for SVM is an optimization problem:

\[
\min \{ \gamma, w, b \} \quad \frac{1}{2} ||w||^2 \quad (2.2)
\]

such that

\[
y_i(w \cdot x_i + b) \geq 1, \quad i = 1, \ldots, m
\]

where \( m \) is the training dataset size and \( (w, b) \) is the smallest function margin with respect to the training dataset among all the points in the training set. \( r \) is the smallest geometric margin of \( (w, b) \) on the training samples. The dual form of this problem reveals that the classification task is only a function of the support vectors.

Cortes and Vapnik [26] developed a modified version of SVM called soft margin SVM. They propose to trade a large margin from the hyperplanes for a small error in the classification. In this case the optimization problem becomes:

\[
\min \{ \gamma, w, b \} \quad \frac{1}{2} ||w||^2 + c \sum_{i=1}^{n} \xi_i \quad (2.3)
\]

such that

\[
y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m.
\]
similarly, here $m$ is the training dataset size and $(w, b)$ is the smallest function margin with respect to the training dataset among all the points in the training set. $r$ is the smallest geometric margin of $(w, b)$ on the training samples and $\xi_i$ is the distance that $i^{th}$ observation is permitted away from the functional margin.

Finally, the kernel trick is to construct a mapping from the original space into an inner product space. Thus we do not need to compute the specific mapping, instead, we compute a modified inner product which we call it a kernel. The kernel trick can be used for non-linear classification. The basic idea is to generalize the dot product in the dual form using a kernel. This allows the algorithm to work in a transformed feature space.

2.2.2 Review of Standard Semi-Supervised Learning Methods

Semi-supervised learning is a statistical learning method that falls between supervised learning and unsupervised learning. Self-training, developed by Scudder [21], was the earliest semi-supervised learning algorithm. Self-training trains a model using the labeled data only and then adaptively applies that model to the unlabeled data. Next it treats the predicted labels as part of labeled data and retrained the model. Although in some cases the self-training method can improve the predictions, see Yarowsky [29], the performance of self-training is unclear and has no rigorous theoretical proof.
Co-training, see Blum and Mitchell [2] is another semi-supervised learning method that tries to make use of a large amount of unlabeled data. The key for co-training is that it requires two views of the data and it has to meet two important assumptions. The view of the data means the data can be partitioned into disjoint subsets, in this case, labeled and unlabeled parts. The first assumption is that each view is sufficient to classify the data, the second is that the two views of the data are conditionally independent given the class label. Krogel and Scheffer [11] showed that co-training is only beneficial when both assumptions are satisfied.

Transductive inference, in contrast to inductive inference, uses the labeled data to predict the labels of the unlabeled data. It only predicts at a finite number of unlabeled data points without a general classification rule, while the goal of inductive inference is to estimate the relationship between $x$ and $y$ over the entire domain. An example of transductive inference is transductive support vector machines (TSVM).

Since the 1990s, more semi-supervised learning methods have become popular and are widely used to tackle a variety of real problems such as natural language recognition, text classification on websites, and image retrieval. Graph-based methods such as mincut [1], manifold regularization and discrete Markov random fields [30] are designed to solve problems like these. In this thesis, the semi-supervised learning method we consider is EM algorithm described below.
Expectation Maximization

The EM algorithm is a semi-supervised learning method that may be used to deal with missing data. It handles the data with partial labels as missing. First we start with the observed data and use this part to build an initial classifier. In the Expectation (E) step, we use the classifier to estimate the probability of the missing data belonging to each class. Next in the Maximization (M) step, we rebuild the classifier based on the results from the E-step. Iterate these two steps until the algorithm converges to obtain a local maximum. Thus the data with missing labels are used to modify the classifier in each iteration. The EM algorithm works well when the model assumptions are met. If the model assumption is doubtful, however, the missing data may not add more value in the estimation.
Chapter 3: Ranked Set Sampling and Judgement
Post-Stratification

In this Chapter, we will briefly introduce two methods: Ranked Set Sampling (RSS) and Judgement Post-Stratification (JPS). We start with the motivation for the methods then go to the implementation. Through the descriptions of RSS and JPS, we will further discuss the connections and differences between these two methods.

3.1 Motivation

In many practical problems, we want to estimate a population mean or proportion. For example, in a political survey, the proportion of American voters who support the Democratic candidate and the proportion of voters who support the Republican are of interest. The result would help us to predict who is more likely to be the next president. Companies such as Amazon.com may want to keep track of customers who view a particular product to estimate what proportion of them actually purchase it. This estimate would help the company to make better business plans. When we translate these problems to mathematical language, we notice that all the variables of interest are binary variables which takes 0-1 values. Thus the question is how to estimate the mean or proportion of this binary variable.
The most straightforward sampling method used in practice is simple random sampling. One desired feature of simple random sampling is it ensures that the mean estimator is unbiased on average. In reality, of course, only one sample is taken and the “on average” idea does not guarantee that researchers will obtain a good sample every time. In some cases, by chance, the simple random sample may not be a good representation of the whole population of interest. For example, in the previous political survey example, by chance, a researcher might get a sample in which there is a much larger representation of Republicans than in the population. Some more structured sampling methods have been developed to address this problem. Such methods include sampling with probability proportional to size, stratified sampling and ranked set sampling (RSS). Similar to stratified sampling, RSS uses additional information to help selecting the units to be in the sample, which achieves the goal of getting a more representative sample.

Obtaining a better sample by bringing more structure to the sampling process is the motivation behind the development of RSS. RSS was first proposed by McIntyre [14] proposed the idea to estimate average yield of arable crops. To collect the actual measurements of yield of arable crops, one needs to mow and weight the hay in each field, an expensive and time consuming process. An expert in agriculture, however, could estimate and rank the yield visually much faster and at little cost. By using this auxiliary information to rank groups of selected fields, RSS enabled McIntyre to select a more representative sample and improved the precision of his estimate without increasing the measurement expenses substantially while only measuring fields
with certain ranks.

Here is how McIntyre carried out RSS. First he selected a simple random sample of fields and let the expert judgement rank the yields of all the fields by visual inspection. Only the field with largest rank was selected for measurement in full. Its accurate measured yield was recorded. Then he took another simple random sample of the same size and this time only measured the field with the second largest visual rank. He did this for each sample until fields with rank 1 to $n$ were measured. He demonstrated that, for a typical unimodal distribution, the mean of a RSS sample is slightly less than $(n+1)/2$ times more efficient than the simple random sample mean. This work was fundamental for RSS’s later development.

RSS did not attract much attention until Takahasi and Wakimoto [23] first derived the result that the sample mean of a ranked set sample is an unbiased estimator of the population mean and its variance is always smaller than the variance of the simple random sample mean with the same sample size under the assumption of perfect ranking. Later Dell and Clutter [24] considered the imperfect ranking situation. They proved that when ranking is not perfect, that is, the $i^{th}$ ordered unit is not necessarily the $i^{th}$ order statistic, the RSS sample mean is still an unbiased estimator of the population mean. Its variance is smaller than the variance of the simple random sample estimator as long as the ranking is better than random ordering. Stokes [22] was the first to use concomitant variables in RSS. Bohn and Wolfe [3] published initial work in modeling imperfect judgment ranking and it was further discussed in Presnell and Bohn [19]. Presnell and Bohn [19] also studied U-statistics based on

The applications and advantages of RSS can be realized in two situations. First is when the measurement of the variable of interest is expensive or time consuming while the judgment ranking of this variable is usually relatively easier to obtain visually by experts in a certain field. Second is when there exists one or more concomitant variables that are correlated with the variable of interest and we can interpret this relationship via modeling. The model would be used to provide rankings for the variable of interest. The goal of RSS is to ensure sampling of each order statistic, therefore bringing more structure to the sample.

3.2 Judgement Ranking

The key step in both RSS and JPS is to get the judgement ranking. Judgement rankings are not necessarily equivalent to order statistics. In other words, the judgemental feature decides that rankings do not have to be perfect. When RSS was first developed, the majority of the judgement ranking was from visual comparisons, done by experts in a certain field. For example, in the estimation of average yield of arable crops problem, experts visually inspect the sampled fields and rank their estimates of yields of arable crops from smallest to largest. The judgement ranking can also be
obtained from one single explanatory variable. Chen, Stasny and Wolfe [5] showed an example of using buttocks circumference or reciprocal of arm circumference separately to rank the estimated body mass index (BMI).

The difficulty of producing judgement rankings grows as the number of sampled units increases. The typical set size choice is 2 to 5 to permit efficient judgement ranking.

Another way of getting the judgement ranking is by building a model using the observed data. For RSS, Chen, Stasny and Wolfe [5] use a logistic regression model with the top few useful explanatory variables to produce a single estimated probability for ranking. In this thesis, we expand this idea to the JPS setting. In Chapter 5, I will use logistic regression, random forests, nearest neighbor, support vector machines and linear discriminant analysis to model the data. I then get the judgement ranking using the predictions from the model. One advantage of getting the judgment ranking from modelling the data is it makes ranking easier. The judgment ranking process, through selecting variables judgementally then building a model, is simplified to ranking one single predicted probability for each unit. Thus, using auxiliary information in a model relaxes the restrictions on the set size and enables us to increase the set size when necessary. In addition, modeling naturally handles multivariate explanatory variables cases. For the univariate case, using a model is equivalent to ranking a single explanatory variable.
3.3 Ranked Set Sampling

Now I will describe the implementation of balanced ranked set sampling. We first select a simple random sample of $m^2$ units from the population. None of the units are fully measured at this point. The sampled units are evenly divided into $m$ sets at random. The number of units within each set, $m$, is referred to as the set size. Within each set, the $m$ units are ranked by an expert visually or by a preselected model. For the first set, only the unit ranked smallest is selected to be in the sample and fully measured. For the second set, the unit ranked second smallest is included in the sample. Repeat this $m$ times. Only the $i^{th}$ ranked unit is fully measured in the $i^{th}$ set. The RSS sample thus contains $m$ units with rankings from 1 to $m$ in each set. The remaining $m(m - 1)$ units are discarded. In total $m$ units are measured. This is called one cycle. This procedure is repeated $n$ times, and a balanced ranked set sample of size $nm$ is obtained. These $nm$ units are then measured. The resulting data may be represented as follows:

\[
Y_{[1]}^1 \ldots Y_{[1]}^i \ldots Y_{[1]}^n
\]

\[
Y_{[2]}^1 \ldots Y_{[2]}^i \ldots Y_{[2]}^n
\]

\[
\ldots
\]

\[
\ldots
\]

\[
Y_{[m]}^1 \ldots Y_{[m]}^i \ldots Y_{[m]}^n.
\]

The judgment ranking may be different from the true ranking. So we use different
brackets to distinguish these two. Round brackets are used for perfect ranking. The first order statistic in the $j^{th}$ set can be represented as $Y_{(1)j}$. Square brackets are used for imperfect ranking or judgment ranking. So $Y_{[1]j}$ refers to the smallest judgment order statistic in the $j^{th}$ set. It may or may not have the smallest actual measurement among the $k$ units in the $j^{th}$ set.

For the $j^{th}$ cycle, the measurements are denoted as $Y_{[1]j}, Y_{[2]j}, \ldots, Y_{[m]j}$. They are independent but not identically distributed. On the other hand, $Y_{[1]1}, Y_{[1]2}, \ldots, Y_{[1]n}$ are i.i.d. distributed across the cycles. Under the perfect ranking assumption, their cumulative distribution function (CDF) is the same as that of the first order statistic $F_{1:m}(y)$. The balanced ranked set sample $Y_{[1]j}, Y_{[2]j}, \ldots, Y_{[m]j}$ ($j = 1, 2, \ldots, n$) is a combination of $m$ i.i.d. samples from each order statistic. It is balanced because the number of units sampled from each order statistic is the same.

The RSS estimator for population proportion is

$$\hat{p}_{RSS} = \frac{1}{m} \sum_{r=1}^{m} \hat{p}_{[r]} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{r=1}^{m} Y_{[r]i}.$$ 

It is an unbiased estimator of the population proportion and it has been shown that its variance is no greater than the variance of the SRS estimator.

To implement unbalanced RSS, we impose is no constraint of equal numbers of judgment ordered units in the RSS sample. Instead, for a fixed set size $m$, one specifies the number of each judgment ordered units $(n_1, n_2, \ldots, n_m)$ where $n_i$ records the
number of units with judgment rank $i$ to be selected in the RSS sample for full measurement. The total sample size for the unbalanced RSS sample is $n = \sum_{i=1}^{m} n_i$.

This idea from RSS to use judgement ranking of units to give a context for the units in the final sample, thereby better representing the whole population and increasing precision, can be extended to other settings. We describe one such extension, Judgement Post-Stratification, in the next subsection.

3.4 Judgement Post-Stratification

Judgment Post-Stratification (JPS), proposed by MacEachern, Stasny and Wolfe [13], is a data analysis method in which the ranking of auxiliary information or judgement ranking is used to improve estimates from a SRS. JPS uses the same ranking concept that is used in RSS, but does not require any special data collection procedure. Therefore researchers retain the flexibility to analyze the sample as a SRS if desired. In their paper, MacEachern et al. [13] derived the JPS sample estimator and proved theoretically, and demonstrated empirically, that the conditional variance of this estimator converges almost surely to the variance of a balanced RSS sample.

MacEachern et al. [13] also discussed the means of allowing imprecise rankings. In such case, instead of assigning a rank to the unit with probability one, one assigns a probability distribution on the ranks. Three methods are proposed in the paper for handling imprecise rankings [13]: proportional proration, random allocation and Rao-Blackwellization. Proportional proration assigns a proportion of the variable of interest to each stratum based on the ranker’s strength of belief that this variable
has each rank. Under random allocation, instead of assigning part of the variable of interest to each stratum as proportional proration does, one randomly allocates the observation to a single stratum based on a multinomial distribution derived from the same proportions as in proration. The last method is a modification of random allocation. It is defined as the conditional expectation of the method two estimator. Thus it is a Rao-Blackwellized version of the method two estimator. A potential problem is the occurrence of empty stratum. In such cases, one can delete the empty stratum and apply the method to the reduced strata.

To illustrate the use of JPS, MacEachern et al. [13] give two empirical examples. The first involves allometry data relating brain weight to body weight in different species of mammals. The result provides evidence of substantial gain in precision using the JPS estimation rather than the SRS estimation. The second example studies graduate student performance for 39 students relating their GPAs at the end of the second year in graduate school to their admission scores. Again the JPS estimator is shown to be much more efficient than the SRS estimator based on a comparable number of fully measured units. The paper also explored the possibility of having multiple rankers.

Other researchers studied JPS extensively after it was developed. Ozturk [16] and Wang [27] improved the method by imposing a stochastically ordered assumption. Thus the improved estimator can be used to calibrate the effect of imperfect ranking. Frey and Ozturk [9] further proposed constrained estimation without additional
assumption in small sample settings. They showed that in the JPS case, better con-
strained small sample estimates of the overall CDF and the population mean can be
obtained. Finally Ozturk [18] showed that combining ranking information in JPS
samples could provide a substantial improvement over the same estimator based on
judgment ranking information of a single best ranker.

In this thesis we study the implementation of JPS in the concomitant variable
setting. After we collect a sample, in which the response variable Y is binary, we
record the units having the response variable values as \((X_1, Y_1), (X_2, Y_2), \ldots,(X_n, Y_n)\).
X can be a single explanatory variable or a vector of explanatory variables. For each
pair of fully observed units, \((m - 1)\) randomly selected units without the response
variables are used for matching and ranking. The originally observed unit along with
the \((m - 1)\) other units makes up a set of size \(m\). Within each set, units are ranked
according to the estimated probability that \(Y\) is 1 using the explanatory variables
via logistic regression (Chen, Stasny and Wolfe (2005)) or via other methods. Then
the ranked units in the \(i^{th}\) set are given by \(X[i_1], X[i_2], \ldots, X[i_m]\) for \(i = 1, 2, \ldots, n\).
Only the ranking, \(R_i\), for the fully observed unit \(X_i\) is recorded for \(i = 1, 2, \ldots, n\).
The \((m - 1)\) matched units are used solely for assisting in the ranking and their own
rankings are not of interest. Repeating the procedure \(n\) times, we get all the fully
observed units with rankings: \((X_1, Y_1, R_1), (X_2, Y_2, R_2), \ldots,(X_n, Y_n, R_n)\). The \(R_i\) may
take on any value from 1 to \(m\).

After the ranking is accomplished, we assign the fully observed units into \(m\) strata
according to their \(R_i\) values. Let \(R_{ih} = I(R_i = h)\) for \(i = 1, 2, \ldots, n; h = 1, 2, \ldots, m\)
indicate that the $i^{th}$ unit belongs to the $h^{th}$ stratum. Then the number of units falling in the $h^{th}$ stratum can be represented as $\sum_{i=1}^{n} R_{ih}$ $i = 1, 2, ..., n; h = 1, 2, ..., m$, and the estimator of the proportion of successes for each stratum is $\hat{p}_h = \frac{\sum_{i=1}^{n} Y_i R_{ih}}{\sum_{i=1}^{n} R_{ih}}$. By averaging all the strata estimators, we get the JPS estimator for the population proportion:

$$\hat{p}_{JPS} = \frac{1}{m} \sum_{h=1}^{m} \hat{p}_h = \frac{1}{m} \sum_{h=1}^{m} \frac{\sum_{i=1}^{n} Y_i R_{ih}}{\sum_{i=1}^{n} R_{ih}}.$$  

When the observed data are from a SRS from the population of interest, the JPS estimator for a population proportion is unbiased and asymptotically as efficient as balanced RSS. There are two extreme cases: random ranking and perfect ranking. When the ranking is random, in which case each unit is equally likely to be given any rank from 1 to $m$, then the JPS estimator is equivalent to the SRS estimator. Thus, without informative judgment rankings, JPS makes no improvement over SRS. When the ranking is perfect, the expectations of the estimated strata means are equal to order statistic means. In this case, the JPS estimator is equivalent to the RSS estimator.

### 3.5 Connections and Differences between JPS and RSS

RSS and JPS share the same fundamental idea. They are both inspired by judgment ranking using it to bring more structure to the sample. For both methods, a set of $m$ units are ranked by either an expert or other ranking scheme. Only one of the $m$ units in each set is chosen for full measurement (RSS) or is already fully measured (JPS). The remaining $m - 1$ units are discarded. This procedure is repeated to get the desired sample of size $n$. JPS builds a bridge between RSS and SRS. When
the ranking information is not considered in JPS, the sample is simply the original SRS. When one conditions on the strata sizes, the sample becomes the same as an unbalanced ranked set sample.

RSS and JPS, however, are still quite different in many ways. RSS is more of a data collection method rather than a sampling or analysis method. In contrast, JPS relaxes the constraint and enables researchers to implement the same judgment ranking idea without collecting the data differently from a SRS. A RSS can be viewed as a stratified sample where the strata are determined by ranks [13] and a JPS sample is a post-stratified sample based on ranks. In short, RSS ranks the units before sampling and JPS does the ranking for a given sample.

In RSS, one can choose to use a balanced RSS design or an unbalanced RSS design. In either case, the numbers of observations from each stratum are fixed in advance. On the other hand, in JPS, one has no control over the post-strata sizes. There is another difference in terms of ranking the units. For RSS, the ranker is forced to rank the units in a set without ties. Doing this ranking could be difficult and not permitting ties could increase the ranking error, leading to lower efficiency. JPS can solve this problem by allowing imprecise ranking in which case units may be assigned probabilities of belonging to each rank. In this thesis, we use JPS as a tool to adjust the missing data in survey sampling as well as data mining problems.
Chapter 4: Simulation Study for the JPS Method under Three Missing Data Scenarios

The nature of missing data makes it hard to distinguish between different missing data mechanisms based only on the observed data. To understand the effects of missing data on our JPS estimator, we simulate datasets for all three types of missingness described in Chapter 2. We consider a univariate case and a bivariate case first. Then we discuss a more complicated mixture Gaussian case. For the univariate case, we can directly rank the observations by the explanatory variable. For the bivariate case and more complicated mixture Gaussian case, we use various supervised learning methods to aggregate the variables first then rank the units in a set. Meanwhile, we are able to make comparisons of JPS estimators with estimators from other semi-supervised learning methods or supervised learning methods.

4.1 Univariate Simulation Study

We generate a univariate mixture Gaussian model with known means and standard deviations. Each mixture component corresponds a certain class ($Y = 1$ or $Y = 0$). The mixture proportion $p$ is known and set to be 0.5. The missingness indicator is generated differently under MCAR, MAR and MNAR mechanisms as described below. The standard deviation of the JPS estimator is estimated using simulation
with \( k \) replicates. \( \hat{p}_{JPS}^{(r)} \) represents the JPS estimator of the population proportion for the \( k^{th} \) sampling iteration. We compare the performance of the JPS estimator with that of the SRS estimator using the reduction of the square root of the sample mean squared error as a summary measure. The proportional reduction of the square root of the mean squared error is

\[
\text{reduction} = 1 - \sqrt{\frac{MSE(\hat{p}_{JPS})}{MSE(\hat{p}_{SRS})}}, \tag{4.1}
\]

where

\[
MSE(\hat{p}_{JPS}) = \frac{\sum_{k=1}^{R} (\hat{p}_{JPS}^{(r)} - p)^2}{(k - 1)}, \tag{4.2}
\]

and

\[
MSE(\hat{p}_{SRS}) = \frac{\sum_{k=1}^{R} (\hat{p}_{SRS}^{(r)} - p)^2}{(k - 1)}. \tag{4.3}
\]

When data are missing completely at random, both JPS and SRS estimators are unbiased. In that case, the measure can be simplified to the standard deviation reduction:

\[
\text{reduction}_{MCAR} = 1 - \frac{SD(\hat{p}_{JPS})}{SD(\hat{p}_{SRS})}.
\]

### 4.1.1 Data Generation

We generate data in the univariate auxiliary variable case with three different types of missingness in two steps. In the first step, we generate a full dataset with \((X,Y)\). Because we first illustrate the univariate case, here \(X\) and \(Y\) are both vectors. In step two, based on the missingness type, we generate a missing data indicator \(M\) and split the dataset into two parts: a learning dataset and a testing dataset.
learning data is the data with known labels, from which we learn the patterns of the
data. Testing data is the data without labels, using which we test the performance
of our methods.

The framework for generating the full dataset is as follows. There are two com-
ponents, $\mathcal{N}(u_1, \sigma^2_1)$ and $\mathcal{N}(u_2, \sigma^2_2)$, in our mixture model with mixture weights $1 - p$
and $p$. The first component corresponds to the $Y = 0$ group; the second one corre-
sponds to the $Y = 1$ group. Since $Y$ is a binary random variable, its density function is

$$f(y) = p^y * (1 - p)^{1-y}, \quad y = 0 \text{ or } 1. \quad (4.4)$$

Given the value of $y$, the density of the auxiliary variable, $x$, is

$$f(x|y) = \psi_y(x) \quad (4.5)$$

where

$$\psi_0(x) \sim \mathcal{N}(u_1, \sigma^2_1) \text{ and } \psi_1(x) \sim \mathcal{N}(u_2, \sigma^2_2).$$

The parameters $u_1$ and $u_2$ are the means of the two Guassian distributions in the
mixture model. In this simulation, we set $u_1 = 0$, $u_2 = 2$. Correspondingly, $\sigma^2_1$
and $\sigma^2_2$ are the variances for the two Guassian distributions in the mixture model. We
assume $\sigma^2_1 = \sigma^2_2 = 1$. We generate 10,000 points from this mixture distribution with
$p = 0.5$ as our population proportion.

**MCAR Sample**

After we obtain the simulated full dataset $(X,Y)$, we turn our attention to how
a particular sample is drawn from it. Let $M$ denote the missing data indicator. To
generate MCAR missingness, we randomly select a sample of 2,000 numbers between 1 to 10,000. Let $M$ for the corresponding cases be 1, and for the remaining cases, $M$ is 0. The cases with $M = 1$ comprise our training dataset. For points in the training dataset, we have the data in the form of $(X, Y)$. The remaining data with $M = 0$ form our testing dataset. For the testing dataset, we discard the $Y$ values and treat them as missing data, using only the $X$ values for our estimation. After we split the data into the training dataset and the testing dataset, the distribution of the explanatory variables remains the same as we show below. Figure 4.1 gives visual evidence that the learning dataset and the population have the same distribution.

$$f(x|M = 0) = f(x|M = 1) = f(x) = p\psi_0(x) + (1 - p)\psi_1(x).$$ \hspace{1cm}(4.6)

Now we have the MCAR dataset ready for the simulation study. Since we also want to investigate the effect of set size on the MSE of our estimators, we use settings of $m = 3, 5, 10, 15$ and 20. First we take a simple random sample of size $n$ from the training dataset, and take another simple random sample of size $n(m - 1)$ from the testing dataset. We match each labeled data point in the training set with $(m - 1)$ unlabeled data points from the testing set and together that forms a set of size $m$. To carry out JPS, within each set we order the $X$ variable from 1 to $m$ and record the rank $R$ for the labeled data point from the training dataset. The remaining points from the testing dataset are used solely for ranking purposes and are discarded after ranking. We repeat this procedure $n$ times for every labeled points in the training
set. Then we get a complete labeled dataset with rankings in a form of \((X, Y, R)\) and may post-stratify our sample into \(m\) strata based on their \(R\) values.

For the results reported below, we chose \(k = 50,000\) simple random samples of size 100 from the training set. Each sample from the training set is matched with another simple random sample of size 100\((m - 1)\) from the testing set to generate rankings. The ranked samples from the training set are used to obtain the JPS estimator. The 50,000 repetitions are used to approximate the sampling distribution of the JPS estimator. We also calculate the SRS estimates using only the samples from the training set and obtain the corresponding sampling distribution of the SRS.
estimator as well.

MAR Sample

For MAR, to be consistent with the MCAR scenario, we use the same mixture Gaussian population generated previously. The only difference is how we generate the missing data indicator variable, $M$. Instead of random selection, under MAR missingness is related to the explanatory variable $X$. We define the probability of a value being missing as

$$f(M|x) = \left(\frac{1}{1 + e^{-x-0.85}}\right)^M \ast \left(\frac{e^{-x-0.85}}{1 + e^{-x-0.85}}\right)^{1-M} \text{ for } M = 0, 1. \quad (4.7)$$

We choose the parameter as $-0.85$ in the missing indicator function to keep the observed versus missing data proportion the same as the MCAR case, about 20% observed. Thus we generate the missing data indicator $M$ with probabilities calculated though the $x$ values. Note that $f(M|x)$ is an increasing function of $x$. We split the dataset into the training dataset containing those units with $M = 0$ and testing dataset of those with $M = 1$. We show the comparison of the whole population and the learning dataset in Figure 4.2. Note that, in contrast to the histograms in Figure 4.1 generated assuming MCAR, the learning data set under our MAR assumption generally falls in the left tail of the population distribution.

MNAR Sample

In the MNAR case, the missingness depends on the $Y$ values. We assign two different missingness probabilities, $p_0$ and $p_1$, for the $Y = 0$ and $Y = 1$ groups
Figure 4.2: Histograms of the population (in blue) and the learning dataset (in red) under the MAR assumption respectively. Under MNAR we take the probability of a value being missing to be

$$f(M|y) = p_y^M \times (1 - p_y)^{1-M} \quad y = 0, 1, \text{ and } M = 0, 1.$$ 

In our simulation, we use $p_0 = 0.67$ and $p_1 = 0.92$. The choice of $p_0$ and $p_1$ ensures that we observe about 20% of the data. Figure 4.3 shows a histogram of the learning data set overlaid on the whole population data set.
As we can see from Figure 4.2 and Figure 4.3, these MAR and MNAR missingness models generate different learning datasets although the difference is almost unidentifiable by the marginal distribution of $x$ and it is hard to foresee the possible influence each has on the estimation.
In the following subsection we compare estimates under JPS and SRS for these three data sets to study the effects of the missing data mechanism on the estimators.

### 4.1.2 Results of Simulation Study

In this thesis, our goal is to estimate the population proportion $p$. In this part of simulation, we will illustrate the JPS method using $k = 50,000$ repetitions and a sample of size $n = 100$. For the univariate case, we only generated a single explanatory variable. Therefore ranking is conducted by simply comparing the $x$ values. If the explanatory variable and the response variable are positively correlated, then a large $x$ value means we are more likely to observe a “success” in the sample and vice versa.

Under MCAR, both the JPS and SRS estimators are unbiased. Our MCAR simulation results in Table 4.1 show that the JPS point estimators are better for every set size choice. Overall, JPS always outperforms SRS. For the MCAR case, the $\sqrt{MSE}$ reduction is mainly from the decrease in the standard deviation. For both MAR and MNAR, the $\sqrt{MSE}$ reduction is from the decrease in the bias. Among three missingness mechanisms, JPS has the best performance for MAR. The JPS performance is also related to the set size. The $\sqrt{MSE}$ achieves the minimum when the set size is around 10, and the $\sqrt{MSE}$ reduction is maximized to be 22.5% compared to SRS estimator. Thus for a fixed sample size of 100 and our generated distribution, the optimal set size is around 10. When the set size increases from 3 to 10, we see that the $\sqrt{MSE}$ reduction increases from 12.2% to 22.5%. So we get better stratification
of the sample when we start to increase the set size. However, as expected, we don’t gain anymore when the set size is too large. As we can see from the table, once the set size is greater than 10, when we continue increasing the set size, the $\sqrt{MSE}$ reduction begins to decrease. This is because for a fixed sample size, a large set size means we are more likely to have empty strata. Meanwhile, the ranking accuracy decreases when the set size is too large. Both will increase the variability in the JPS estimation.

Table 4.1 also shows the results of the MAR simulation. Recall under MAR, missingness is related to the explanatory variables. We generated the missing data indicator in such a way that more data points with large $x$ values are missing. Thus, the SRS estimate of $\hat{p} = 0.205$ is heavily biased for the true population proportion $p = 0.5$. As a nonparametric method, JPS does not require assumptions about the missingness or the underlying population distribution. When the set size is 10, the JPS estimate is 0.478. JPS has adjusted for most of the bias in the sample so that the estimate is much closer to the true proportion than is the SRS estimate. We see a similar pattern as for the MCAR case in the $\sqrt{MSE}$ reduction. As the set size increases from 3 to 20, the reduction first increases from 53.3% to 73.8% then decreases slightly to 72.8%. The maximum reduction of 73.8% is obtained when the set size is around 15. We can also see that there is a trade off between bias and variance. JPS estimators have smaller bias but larger standard deviation for all set size choices compared to SRS estimators. To reduce the bias, JPS puts smaller weights on the units in the strata of larger order statistics. The assignment of the units, however, depends on each sample and it varies considerably from sample to sample. Therefore it results in a larger standard deviation for the JPS estimate. Overall, in the setting
of MAR, the smaller bias results in a reduction in $\sqrt{MSE}$ that is striking.

Finally, the last block of Table 4.1 shows the results of the MNAR simulation. First we note that MNAR performs better in terms of MSE than SRS in all settings. It always has a smaller bias but a slightly larger variance. In the best case it reduces $\sqrt{MSE}$ by 50.2% when the set size is 15. Comparing the reduction of the bias in the point estimate of $p$ obtained under MNAR with the reduction under MAR for all set sizes, we see that although JPS under MNAR still corrects for the bias, the reduction in the bias is not as successful as under the MAR situation. The best JPS estimate of $p$ is 0.37 under MNAR while it is 0.478 under MAR compared to the true proportion of 0.5. We also compared the percentage $\sqrt{MSE}$ reduction under MNAR with the reduction under MAR. The trend is the same for MNAR and MAR. As the set size increases, the reduction first gets larger from 28.2% to 50.2% then starts to drop. The percentage $\sqrt{MSE}$ reduction for MNAR is smaller than that for the same set size under MAR.

4.1.3 The Effects of Sample Size and Set Size

We also investigated the effects of sample size and strata size on improvements in the JPS estimator compared to the SRS estimator. We find that although we do not find a universal optimal combination of $n$ and $m$, there are still useful rules we can follow.
<table>
<thead>
<tr>
<th>Missing data mechanism</th>
<th>SRS estimator</th>
<th>JPS estimator</th>
<th>JPS versus SRS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Stdev</td>
<td>Mean</td>
</tr>
<tr>
<td><strong>MCAR</strong></td>
<td>0.508</td>
<td>0.049</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.504</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>0.504</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td>0.503</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td>0.503</td>
</tr>
<tr>
<td><strong>MAR</strong></td>
<td>0.205</td>
<td>0.039</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.447</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>0.478</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td>0.473</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td>0.461</td>
</tr>
<tr>
<td><strong>MNAR</strong></td>
<td>0.222</td>
<td>0.041</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.336</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>0.366</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td>0.370</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td>0.365</td>
</tr>
</tbody>
</table>

Note: The true proportion is $p=0.5$.

Table 4.1: JPS estimator compared to SRS estimator under different missing data mechanisms
The simulation is set up as following. The learning data and testing data are the same as we used in the MCAR simulation. Three different sample size and set size combinations are chosen. Sample size takes value of 200 and 400. Set size takes value of 4 and 8. Repetition \( k = 50,000 \) and results are shown in Table 4.2. We find when we double the set size from 4 to 8, the \( \sqrt{MSE} \) reduction increases from 16.6% to 24%. Thus, the efficiency of the estimate of the population proportion, \( p \), improves about 45%. Meanwhile, when we double the sample size from 200 to 400, the \( \sqrt{MSE} \) reduction increases from 16.6% to 17.6%. This improvement is about 6%. Therefore in this case, when the explanatory variable contains information about the label, increasing the set size works better to achieve greater \( \sqrt{MSE} \) reduction compared to increasing the sample size. This is a useful conclusion since in many situations, labels are expensive and hard to obtain but auxiliary information may be relatively cheap.

Now if we only focus on changing the set size, we will find there exists an optimal set size choice for a fixed sample size. For the same MCAR data and fixed sample size of \( n = 100 \), we change the set size from 3 to 15; the results are shown in Figure 5.2. The MSE of JPS first decreases from 0.0018 to 0.0015 when we increase the set size.
Figure 4.4: The change in MSE for set sizes ranging from 3 to 15; sample size is 100.

from 3 to 8. When we keep increasing the set size up to 15, however, there is no additional reduction in MSE. Instead, we see the MSE starts to increase when the set size is too large. Thus in this setting, the optimal set size is around 8.

In conclusion, by using JPS and choosing a reasonable set size, we are able to gain efficiency without increasing the sample size. We need to keep in mind, however, that this conclusion is drawn from our simulated data where we have control over the relationship between x and y and missingness is completely at random.
4.1.4 Proportional Proration

Another extension of the JPS method is to refine the ranking process. So far we have broken the ties randomly when there is a tie in the ranking, and we assign the unit into one of the stratum. MacEachern, Stasny and Wolfe [13] also mentioned how to handle ties using proportional proration in their paper. They discuss how the ranker is not certain about the ranking, the ranker can assign part of the observation into different strata based on how confident the ranker is about each ranking. In our problem setting, the judgment ranking is done using certain ranking procedures such as a logistic regression ranking procedure or random forests ranking procedure, so to incorporate the uncertainty idea into the ranking, we need to do JPS multiple times. Instead of matching each observation with \((m - 1)\) unlabeled units once, we can matched it with a few other sets of \((m - 1)\) unlabeled units and calculate the proportion of times the original unit falls into the various strata. Then we assign part of the observation into different strata and get the JPS estimator using these ranking probabilities as described in MacEachern et al. [13]. We applied this procedure to our data to explore more about the features and efficiency of this method.

We conduct a small simulation for this purpose. First we use the MCAR dataset. To create ties, we round all the \(x\) values to a single decimal place. In a sample of size 100, by averaging over all the iterations, we find that we obtain ties about 5% of the time during the ranking process. We use the proportional proration JPS method and our usual JPS method to estimate the population proportion and repeat the process for 50,000 iterations. The results are shown in Table 4.3.
Table 4.3: JPS and JPS(proportional proration) estimators compared to SRS estimator under MCAR assumption

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>MSE</th>
<th>$\sqrt{MSE}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPS</td>
<td>0.50</td>
<td>0.00183</td>
<td>0.0428</td>
</tr>
<tr>
<td>JPS(proportional proration)</td>
<td>0.50</td>
<td>0.00183</td>
<td>0.0426</td>
</tr>
<tr>
<td>SRS</td>
<td>0.51</td>
<td>0.00243</td>
<td>0.0439</td>
</tr>
</tbody>
</table>

As we see from the Table 4.3, by using proportional proration to deal with the ties, even though there are only about 5% ties in the sample, there is an improvement of the proportional proration JPS method compared to using a random mechanism to break ties in the JPS method. We suspect, therefore, that it would be quite useful to use proportional proration to handle ties when we have all category variables and conduct the ranking using logistic regression. In that case we observe many ties in the ranking procedure when we have a small set size, and we believe that using proportional proration will help increase the accuracy of estimation.

We also carried out a simulation to study proportional proration using the MAR dataset. Again to create ties, we round all the $x$ values into a single decimal place. We observe about 5% ties during the ranking process. We use the proportional proration JPS method and our usual JPS method to estimate the population proportion and repeat the process for 50,000 iterations. The results are shown in Table 4.4.

We can see from the Table 4.4, by using proportional proration to deal with the ties, the $\sqrt{MSE}$ decreases only slightly. There are only about 5% of the ties in the
Table 4.4: JPS and JPS(proportional proration) estimators compared to SRS estimator under MAR assumption

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>MSE</th>
<th>$\sqrt{MSE}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPS</td>
<td>0.337</td>
<td>0.0316</td>
<td>0.1778</td>
</tr>
<tr>
<td>JPS(proportional proration)</td>
<td>0.337</td>
<td>0.0316</td>
<td>0.1776</td>
</tr>
<tr>
<td>SRS</td>
<td>0.204</td>
<td>0.0920</td>
<td>0.3034</td>
</tr>
</tbody>
</table>

Table 4.5: JPS and JPS(proportional proration) estimators compared to SRS estimator under MNAR assumption

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>MSE</th>
<th>$\sqrt{MSE}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPS</td>
<td>0.230</td>
<td>0.077</td>
<td>0.278</td>
</tr>
<tr>
<td>JPS(proportional proration)</td>
<td>0.230</td>
<td>0.077</td>
<td>0.278</td>
</tr>
<tr>
<td>SRS</td>
<td>0.222</td>
<td>0.082</td>
<td>0.286</td>
</tr>
</tbody>
</table>

In summary, JPS using proportional proration improves the estimator when there are ties in the ranking procedure. In situations where we have all category variables...
and conduct the ranking using logistic regression, we may observe many ties in the ranking especially when we have a small set size. In that case JPS using proportional proration can help increase the estimation accuracy as we have seen in the simulation study.

### 4.1.5 Comparison of JPS with Expectation Maximization

JPS is different from many other classification-based estimation methods. For a regular supervised learning method, the classification boundary is important. A classification rule around this boundary determines the error rate and the estimation accuracy. JPS, however, does not provide a classification rule. The traditional JPS method does not predict each missing value, but focuses on predicting the overall proportion \( p \). Thus JPS is closer to methods based on using the EM algorithm in certain ways. We briefly compare the EM algorithm with the JPS method through the following simulation.

Let \( x = (x_1, x_2, ..., x_n) \) be a sample of \( n \) independent observations from a mixture of two multivariate normal distributions, and let \( y = (y_1, y_2, ..., y_n) \) be the latent variables that indicate the component from which the observation originates. Suppose

\[
(X_i|Y_i = 1) \sim N(\mu_1, \Sigma_1) \quad \text{and} \quad (X_i|Y_i = 2) \sim N(\mu_2, \Sigma_2)
\]

where \( P(Y_i = 1) = p_1 \) and \( P(Y_i = 2) = p_2 = 1 - p_1 \). The goal is to estimate the means and covariances of each Gaussian distribution as well as the mixing proportion. Thus, the parameter vector to be estimated is \( \theta = (p_1, \mu_1, \mu_2, \Sigma_1, \Sigma_2) \).
To carry out the EM algorithm, first we need the likelihood function:

\[
L(\theta; x, y) = P(x, y|\theta) = \prod_{i=1}^{n} \sum_{j=1}^{2} I(y_i = j)p_i f(x_i; \mu_i, \Sigma_i)
\]

(4.8)

where I is an indicator function and f is the probability density function of a Gaussian distribution. As we mentioned earlier, the EM algorithm has two steps: the E-step and the M-step. For the E-step, given the current estimate of the parameter, \(\theta^{(t)}\), the conditional distribution of the \(z_i\) is determined by Bayes theorem, as follows:

\[
T_{j,i}^{(t)} = P(z_i = j|x_i, \theta^{(t)}) = \frac{p_j^{(t)} f(x_i; \mu_j^{(t)}, \Sigma_j^{(t)})}{p_1^{(t)} f(x_i; \mu_1^{(t)}, \Sigma_1^{(t)}) + p_2^{(t)} f(x_i; \mu_2^{(t)}, \Sigma_2^{(t)})}.
\]

(4.9)

Then for the M-step, we use the quadratic form of \(Q(\theta|\theta^{(t)})\) to determine the maximum value of \(\theta\), as follows:

\[
Q(\theta|\theta^{(t)}) = E[\log L(\theta; x, y)] = \sum_{i=1}^{n} \sum_{j=1}^{2} T_{j,i}^{(t)} \left[\log(p_j) - \frac{1}{2} \log|\Sigma_j| - \frac{1}{2} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) - \frac{d}{2} \log(2\pi)\right].
\]

(4.10)

(4.11)

We also have the constrain that \(p_1 + p_2 = 1\). Therefore, we can solve for \(p_1^{(t+1)}\).

\[
p_1^{(t+1)} = \arg \max Q(\theta|\theta^{(t)}) = \arg \max \left\{\sum_{i=1}^{n} T_{1,i}^{(t)} \log p_1 + \sum_{i=1}^{n} T_{2,i}^{(t)} \log p_2\right\}
\]

(4.12)

\[
p_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{j,i}^{(t)}}{\sum_{i=1}^{n} (T_{1,i}^{(t)} + T_{2,i}^{(t)})}
\]

(4.13)

One iterates between the E-step and the M-step until convergence. Our goal is to estimate the success probability, therefore, \(p_2 = P(Y = 2)\) is the parameter of interest.
In the mixture model discussed above, there are five unknown parameters. To simplify the problem, for our simulation we assume the means and variance covariance matrix are known and the only unknown parameter is \( p_2 \). Using the MCAR assumption, we apply EM, JPS and SRS to estimate the population proportion. To study the performance under various mean differences and true proportions, we use means \((\mu_1, \mu_2) = (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5)\) and proportions \( p = 0.5, 0.65, 0.85 \). \( \mu \) represents the mean difference \( \mu = \mu_2 - \mu_1 \). The results are shown in Figure 4.5.

From Figure 4.5, we see that for a parametric method, under the correct model assumption, the EM-algorithm does really well in predicting the population proportion for MCAR data. The JPS method, as we expected, performs better than SRS but worse than EM. The performances of all methods improve, of course, as the true population proportion gets further away from 0.5. In addition, the estimated MSEs get smaller as the mean differences get bigger. These occur because as the true population proportion gets larger and the mean difference (the distance between the means of mixture Gaussians) increases, the two groups are more separated which makes prediction easier.

We next repeat the simulation comparing EM, JPS, and SRS in the MAR case. \( \mu \) represents the mean difference \( \mu = \mu_2 - \mu_1 \). The results are shown in Figure 4.6. For the three methods we compare here, clearly EM using the labeled data and correct model assumption is the best among all three methods. The JPS method is between SRS and EM. Since JPS makes use of information in labeled and unlabeled data, it therefore outperforms SRS. The other thing we observe here is that when the mean
\( \mu = 0, 1, 2, 3, 4, 5 \) for different values of \( p \), the population proportion

Figure 4.5: MSE comparison for JPS, SRS and EM under MCAR assumption

difference gets bigger, the MSE also increases for all the methods. That is because when we simulate the data under the MAR assumption, the missingness is related to the \( x \) values. When the mean difference gets bigger, the labeled data is more unbalanced, therefore, the SRS estimation bias is larger.

From both the MCAR and MAR simulations, we can see that JPS, as a nonparametric method, improves over SRS, and is very reliable. Its performance is always
Figure 4.6: MSE comparison for JPS, SRS and EM under MAR assumption

between the EM method (with the correct assumption) and SRS for all the situations.

4.2 Bivariate Simulation Study

We have already analyzed the JPS method, as a semi-supervised learning method, to explore its performance under three missing data mechanisms in the univariate case. We have compared it to the SRS method and seen great improvement. One interesting
aspect would be how JPS compares to other machine learning methods for the same dataset. We are also interested in the JPS performance in the bivariate case. Various machine learning methods including JPS will be studied in this section. In Chawla and Karakoulas [4], the authors set up a framework to evaluate empirically different machine learning techniques. We will extend their framework to the JPS case to help us understand the effect of sample selection bias and the suitable applications for JPS.

Supervised learning develops a mapping between the auxiliary information \((X)\) and response variable \((Y)\) solely from the labelled data. We will review linear discriminant analysis (LDA), K-nearest neighbor (KNN) and support vector machines (SVM) as examples for supervised learning methods. Labelled data can be expensive in terms of time or money, while unlabelled data may be easy to obtain. Naturally, therefore, we want to use the unlabelled data to help improve our estimates. This is the motivation for semi-supervised learning. Our method, using Judgment Post-Stratification (JPS), can be viewed as a special type of semi-supervised learning. The traditional semi-supervised learning methods are classification methods. They use both labelled and unlabelled data to build a classifier and predict the label for each case. Our JPS method is designed to solve the population proportion estimation problem and, therefore, it uses both labelled and unlabelled data to calculate estimates of the population proportion instead of predicting the label for each case.

To better understand how JPS performs under different missing data scenarios and to compare the results with other existing supervised learning methods such as SVM, we generate three datasets: MCAR, MAR and MNAR separately. In real world
examples, it is hard to distinguish MAR from MNAR. By using a simulated dataset, we can systematically study how the missingness characteristics affect the results of different estimation methods. In addition, we hope we can make inference about the missingness in the data through observing the behaviour of estimation methods.

For the univariate case, ranking is done by directly comparing the $x$ value, therefore, no modelling process is involved. To compare JPS with other ML techniques, we will generate a bivariate case. The missingness types we are interested in are still MCAR, MAR and MNAR.

### 4.2.1 MCAR

We first investigate the simplest missing data case, MCAR. For MCAR, a mixture Gaussian distribution is generated. We consider a bivariate Gaussian as explanatory variable here. There are two components $N(u_1, \Sigma_1)$ and $N(u_2, \Sigma_2)$ in the mixture model with mixture weights $1 - p$ and $p$. The first component corresponds to the $Y = 0$ group and the second part corresponds to the $Y = 1$ group. Let $M$ denote the missing data indicator. The distributions of $y$ and $x|y$ are shown as follows:

$$f(y) = p^y \times (1 - p)^{1-y}, \quad y = 0 \text{ or } 1 \quad (4.14)$$

$$f(x|y) = \psi_y(x) \quad (4.15)$$

where

$$\psi_0(x) \sim N(u_1, \Sigma_1) \text{ and } \psi_1(x) \sim N(u_2, \Sigma_2).$$

We generate 10,000 points from the mixture Gaussian with $p = 0.5$ as our population parameter of interest. We choose $u_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $u_2 = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}$, $m = 3$ and
\[ \Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \]. We split the data into training and testing portions with a ratio of 1:4. For the cases in the training dataset, we keep their \( Y \) labels so they are in a form of \((X, Y, M = 0)\). For the units in the testing dataset, we discard the \( Y \) values and treat them as data without labels. So they are in a form of \((X, M = 1)\).

For each repetition, we take a simple random sample of size \( n \) from the training dataset, and take another simple random sample of size \( n(m - 1) \) from the testing dataset. When we carry out SVM, KNN and logistic regression, we train a classifier or build a model based only on the sample from the training dataset. For JPS, each point in the sample from training dataset is matched with \( m - 1 \) points in the sample from the testing dataset. We call this a set of size \( m \). Therefore, in each set only one data point has a label.

One important step in JPS is to rank the data points in a set using their auxiliary information. When the explanatory variable is univariate, as for the simulations in Chapter 3, we can directly rank the \( x \) values. For the case with two (or more) auxiliary variables, we first need to aggregate the information in the explanatory variables into one value and then use it for ranking. Therefore, we develop our JPS method based on the predicted probabilities given by SVM, KNN and logistic regression. Within a set, we rank the data points from 1 to \( m \) according to their predicted probabilities. Only the rank of the unit having a label is recorded. We repeat the ranking process \( n \) times, once for each set, and obtain rankings for all \( n \) data points from the training dataset. Then we post-stratify the sample into \( m \) strata according to the ranking and
obtain the corresponding JPS estimate.

For comparison purposes, we also apply SVM, KNN and LDA directly to the \( mn \) data points in both samples to calculate the proportion estimators. Both the JPS method and the modified JPS method use a total of \( mn \) data points so we can compare their performance fairly. We iteratively sample 50,000 times from the learning data to approximate the sampling distribution of the SRS, JPS, SVM, KNN and LDA estimators. We use the SRS estimator as our baseline and compare every other method with it using various measures. These measures include bias, root mean squared error, in-sample error rate and root mean squared error reduction (compared to SRS). In-sample error rate is calculated by applying the trained model to the sampled units. Table 4.6 shows the results of this simulation when the missing data mechanism is MCAR.

From Table 4.6, We notice that, compared to the traditional methods of RF, LDA, SVM and 5NN, using the same amount of data but also applying JPS, the \( \sqrt{MSE} \) can be greatly reduced. We use RF for the original random forests method and JPS(RF) for the JPS method using rankings from RF. The RF method by itself compared to SRS has a \( \sqrt{MSE} \) reduction of 9.3% and JPS(RF) increases the reduction to 12.3%. Similary JPS(LDA) represents the JPS method using rankings from LDA. JPS(LDA) has a \( \sqrt{MSE} \) reduction of 14.6% compared to 10.7% for LDA alone. In the cases of SVM and KNN where we choose \( K = 5 \), SVM and 5NN perform worse than SRS using \( nm \) data points. In addition, their in-sample prediction error rates are relatively high, about 14.1% and 12.6% respectively. Thus we do not expect their predicted
probability for unlabeled data to be very accurate.

One of the important features of JPS, is that it does not require highly accurate predicted values. As long as the ranking information reflects the order of the likelihood of being a success, JPS will be able to improve the estimator. That is exactly what we see in the results of our simulation study. For both SVM and 5NN, the JPS(SVM) and JPS(5NN) are used for the JPS method using rankings from SVM and 5NN. The JPS(SVM) and JPS(5NN) improve from negative $\sqrt{MSE}$ reductions to 14.7% and 10.9% $\sqrt{MSE}$ reduction, respectively. If we look more closely at the SVM and 5NN rows of Table 4.6, we can see that most of the improvement comes from the reduction in the standard deviation. This results agree with what we have see in the univariate case and the bivariate case simulation studies for MCAR.

The right half of Table 4.6 provides the results from estimators using all the data (10,000 data points). Thus the RF, LDA, SVM and 5NN classifiers are trained using 10,000 points. All the estimators have a better performance when more data is used; the reduction is mainly a result of the decrease in the standard deviation.

From the MCAR case we have seen that JPS can be built on any predictive method, which means that the JPS rankings can be done by any predictive model. The predictive model JPS chooses does not have to outperform SRS. Under the condition of using the same amount of data, the JPS method can improve upon the original method by reducing its standard deviation. Therefore using the JPS method
in combination with ML methods results in a greater $\sqrt{MSE}$ reduction.

The other important feature we wish to note is that the performance of JPS is also influenced by the set size and sample size. Because we want to make a fair comparison to the other traditional methods, we keep the sample size of 100 the same but change the set size. The results of this study are shown in Table 4.7.

In Table 4.7 we see that, as we expected, when we increase the set size from 3 to 6, the performance of JPS improves. The improvements are slightly different depend on which traditional supervised learning method JPS is built upon. The JPS(SVM) has the greatest increase in the $\sqrt{MSE}$ reduction: from 14.7% to 20.9%. This demonstrates that the JPS method, in addition to making improvements over the original methods, can improve even more when the proper set size is chosen.

4.2.2 MAR

The second type of missing data we consider is MAR. For MAR, to be consistent with the MCAR simulation study, we use the same mixture Gaussian population generated for the MCAR study. The missing data mechanism, however, is different. We construct a new missing indicator variable $M$ which follows the conditional density

$$ f(M|x) = \frac{1}{1 + e^{-10x_1 - x_2}}. $$

Note that $f(M|x)$ is an increasing function in the $w = x_1 + x_2$ direction. We observe about 30% of the data ($M=0$) and the remaining 70% ($M=1$) are treated as missing. We then split the data according to the $M$ values. If $M = 0$ then the data point
belongs to the training dataset, otherwise it belongs to the testing dataset. We also apply LDA, RF, SVM, 5NN with JPS on the MAR dataset. The results are presented in Table 4.8.

From Table 4.8, we see that JPS based on ML methods improves the $\sqrt{MSE}$ reduction over the original methods in most cases. The JPS(SVM) increases the $\sqrt{MSE}$ reduction from -8.361\% to 33.23\%, which is the greatest improvement. The JPS(5NN) has the second largest improvement from 11.848\% to 23.011\%. In conclusion, by applying JPS using ML predicted probabilities, we can improve the performance for most ML methods.

### 4.2.3 MNAR

The third type of missing data we consider is MNAR. For MNAR, we generate the missing indicator variable $M$ as a function of the $Y$ values. We assign two different missing data probabilities, $p_0$ and $p_1$, for the $Y = 0$ and $Y = 1$ groups. Thus, the conditional density of $M$ given the $y$ value is

$$f(M|y) = p_y^M \cdot (1 - p_y)^{1-M} \quad y = 0 \text{ or } 1.$$  

In our simulation, we use $p_0 = 0.4$ and $p_1 = 0.6$. This choice of $p_0$ and $p_1$ ensures that we observe about 20\% of the data. The learning dataset is very unbalanced, with very few units from the $Y = 0$ group. The proportion of $Y = 1$ cases in the learning dataset is about 0.9 while the true population proportion is 0.5. This creates many extreme samples in the simulation. If the sample is too extreme, neither the original ML method or the JPS method can do much to reduce the bias in the result and we
want to exclude these samples. In fact, neither method works when the sample is too extreme. In our table calculation, we exclude about 3,000 samples with proportion smaller than 0.1 or greater than 0.9. The result of the simulation for MNAR are shown in Table 4.9.

In Table 4.9, we see that adding a JPS step to the estimator improves the result of SVM from 11.8% to 19.7%. For RF and 5NN, however, JPS does not result in much improvement over the original methods. LDA is still the best among all the methods. It seems that under the MNAR assumption, using the JPS method does not improve the results. Using more data points appears to be a better way to increase the $\sqrt{MSE}$ reduction and produce better estimators.

### 4.3 A More Complicated Example

So far in our simulation studies, we generated the univariate case as well as the bivariate case with uncorrelated means and diagonal variance covariance matrix. For methods like LDA that rely heavily on the assumption of normality, we expect to see good performances in our simulation. There are many cases, however, where classification methods like LDA which produces a linear boundary is not the most appropriate choice. To study our methods in a case where a linear boundary is not reasonable, we use a data set discussed in the book written by Hastie et al. [10].

The dataset is generated as follows. First 10 mean values are generated from a bivariate Gaussian distribution, $N \left( \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right)$ and labelled as class 0. Then 10
more mean values are drawn from $N \left( \begin{pmatrix} 0 \\ 1 \\ \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \end{pmatrix} \right)$ and labelled class 1. For each class Hastie et al. [10] generated 100 observations to construct a learning dataset of size 200. For each observation, they first randomly pick a mean $\mu_k$, either from $Y = 0$ class or $Y = 1$ class. Because each class has ten randomly drawn mean values, the probability that one observation takes any of them is 0.1. The next step is to draw the observation from $N(\mu_k, I/5)$ where $I$ is the identity matrix. Naturally, the proportion is 0.5 in the learning dataset. Because of the way the authors generate the data, the true population proportion is also 0.5. So this is a MCAR situation.

We still use a sample of size 100 and a set of size 3. We calculate the test error rate to compare RF, LDA, SVM and 5NN methods with the JPS method. The test error rate is calculated by applying the classification rule to the testing dataset. Results of the comparisons are shown in Table 4.10.

Table 4.10 shows that the test error rate is always higher than the in-sample error rate for all the methods. This is fairly reasonable because we use the learning dataset to generate the classifier and apply it to the same dataset to get the in-sample error rate. Therefore the in-sample error rate is lower than the test error rate. Also we notice the $\sqrt{MSE}$ reduction is closely related to the test error rate. SVM has the highest test error rate among all the methods we consider, and, correspondingly, its $\sqrt{MSE}$ reduction is the lowest. Comparing the JPS type estimator with the standard ML estimators, we see that for all methods, by applying the JPS method, we see improvements in the $\sqrt{MSE}$ reduction. Among all the methods, SVM and 5NN
have the largest improvements, by 57% and 19%, respectively.

From the simulations, we can conclude that applying JPS will reduce the standard deviation for MCAR data and reduce the bias for MAR and MNAR. In most situations, adding the JPS step to the ML methods brings improvements in the $\sqrt{MSE}$ reduction. Among the four ML methods we tried, SVM and 5NN have the greatest improvements. When we use a more complicated dataset to test our methods, we see that the linear boundary does not work so well anymore. By adding the JPS step, the results gets much better. Thus, adding the JPS step is a good way to reduce the $\sqrt{MSE}$. 
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Stdev</th>
<th>In-sample error rate(%)</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
<th>Mean</th>
<th>Stdev</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
<td>0.477</td>
<td>0.048</td>
<td>0.052</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPS(rf)</td>
<td>0.485</td>
<td>0.044</td>
<td>0.046</td>
<td>12.292</td>
<td></td>
<td>0.488</td>
<td>0.042</td>
<td>0.043</td>
<td>17.071</td>
</tr>
<tr>
<td>RF</td>
<td>0.486</td>
<td>0.046</td>
<td>0.000</td>
<td>9.302</td>
<td></td>
<td>0.488</td>
<td>0.042</td>
<td>0.043</td>
<td>17.071</td>
</tr>
<tr>
<td>JPS(lda)</td>
<td>0.482</td>
<td>0.042</td>
<td>0.045</td>
<td>14.568</td>
<td></td>
<td>0.486</td>
<td>0.032</td>
<td>0.034</td>
<td>34.629</td>
</tr>
<tr>
<td>LDA</td>
<td>0.484</td>
<td>0.044</td>
<td>14.377</td>
<td>10.735</td>
<td></td>
<td>0.486</td>
<td>0.032</td>
<td>0.034</td>
<td>34.629</td>
</tr>
<tr>
<td>JPS(svm)</td>
<td>0.483</td>
<td>0.042</td>
<td>0.044</td>
<td>14.685</td>
<td></td>
<td>0.486</td>
<td>0.041</td>
<td>0.043</td>
<td>17.386</td>
</tr>
<tr>
<td>SVM</td>
<td>0.483</td>
<td>0.052</td>
<td>14.095</td>
<td>-4.388</td>
<td></td>
<td>0.486</td>
<td>0.041</td>
<td>0.043</td>
<td>17.386</td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.482</td>
<td>0.044</td>
<td>0.046</td>
<td>10.867</td>
<td></td>
<td>0.487</td>
<td>0.043</td>
<td>0.044</td>
<td>15.022</td>
</tr>
<tr>
<td>5NN</td>
<td>0.485</td>
<td>0.051</td>
<td>12.601</td>
<td>-0.980</td>
<td></td>
<td>0.487</td>
<td>0.043</td>
<td>0.044</td>
<td>15.022</td>
</tr>
</tbody>
</table>

Table 4.6: Comparison of estimators using simulated data under MCAR assumption: sample size $n = 100$, set size $m = 3$, iteration $k = 5,000$
Table 4.7: Comparison of estimators using simulated data under MCAR assumption: sample size $n = 100$, set size $m = 3$ and $m = 6$, iteration $k = 5,000$
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Stdev.</th>
<th>In-sample error rate(%)</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
<th>Mean</th>
<th>Stdev.</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
<td>0.119</td>
<td>0.019</td>
<td>0.379</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPS(RF)</td>
<td>0.352</td>
<td>0.136</td>
<td>0.200</td>
<td>47.321</td>
<td></td>
<td>0.289</td>
<td>0.151</td>
<td>0.258</td>
<td>32.020</td>
</tr>
<tr>
<td>RF</td>
<td>0.290</td>
<td>0.140</td>
<td>0.000</td>
<td>33.822</td>
<td></td>
<td>0.289</td>
<td>0.151</td>
<td>0.258</td>
<td>32.020</td>
</tr>
<tr>
<td>JPS(LDA)</td>
<td>0.276</td>
<td>0.114</td>
<td>0.250</td>
<td>34.082</td>
<td></td>
<td>0.439</td>
<td>0.102</td>
<td>0.118</td>
<td>68.978</td>
</tr>
<tr>
<td>LDA</td>
<td>0.408</td>
<td>0.097</td>
<td>9.702</td>
<td>65.192</td>
<td></td>
<td>0.439</td>
<td>0.102</td>
<td>0.118</td>
<td>68.978</td>
</tr>
<tr>
<td>JPS(SVM)</td>
<td>0.295</td>
<td>0.151</td>
<td>0.253</td>
<td>33.230</td>
<td></td>
<td>0.134</td>
<td>0.185</td>
<td>0.409</td>
<td>-7.754</td>
</tr>
<tr>
<td>SVM</td>
<td>0.125</td>
<td>0.173</td>
<td>10.510</td>
<td>-8.361</td>
<td></td>
<td>0.134</td>
<td>0.185</td>
<td>0.409</td>
<td>-7.754</td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.216</td>
<td>0.073</td>
<td>0.292</td>
<td>23.011</td>
<td></td>
<td>0.192</td>
<td>0.118</td>
<td>0.329</td>
<td>13.385</td>
</tr>
<tr>
<td>5NN</td>
<td>0.183</td>
<td>0.112</td>
<td>8.940</td>
<td>11.848</td>
<td></td>
<td>0.192</td>
<td>0.118</td>
<td>0.329</td>
<td>13.385</td>
</tr>
</tbody>
</table>

Table 4.8: Comparison of estimators using simulated data under MAR assumption: sample size $n = 100$, set size $m = 3$, iteration $k = 5,000$
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Stdev.</th>
<th>In-sample error rate(%)</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
<th>Mean</th>
<th>Stdev.</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
<td>0.884</td>
<td>0.016</td>
<td>0.386</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPS(RF)</td>
<td>0.772</td>
<td>0.040</td>
<td>0.277</td>
<td>28.394</td>
<td></td>
<td>0.751</td>
<td>0.052</td>
<td>0.258</td>
<td>33.306</td>
</tr>
<tr>
<td>RF</td>
<td>0.766</td>
<td>0.047</td>
<td>0.000</td>
<td>0.271</td>
<td>29.768</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPS(LDA)</td>
<td>0.802</td>
<td>0.038</td>
<td>0.306</td>
<td></td>
<td>20.805</td>
<td>0.761</td>
<td>0.044</td>
<td>0.237</td>
<td>38.710</td>
</tr>
<tr>
<td>LDA</td>
<td>0.761</td>
<td>0.044</td>
<td>7.150</td>
<td>0.267</td>
<td>31.018</td>
<td></td>
<td>0.731</td>
<td>0.044</td>
<td>31.018</td>
</tr>
<tr>
<td>JPS(SVM)</td>
<td>0.806</td>
<td>0.039</td>
<td>0.310</td>
<td></td>
<td>19.739</td>
<td>0.830</td>
<td>0.078</td>
<td>0.341</td>
<td>11.831</td>
</tr>
<tr>
<td>SVM</td>
<td>0.830</td>
<td>0.078</td>
<td>7.423</td>
<td>0.341</td>
<td>11.831</td>
<td></td>
<td>0.809</td>
<td>0.085</td>
<td>32.213</td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.819</td>
<td>0.035</td>
<td>0.323</td>
<td></td>
<td>16.516</td>
<td>0.798</td>
<td>0.057</td>
<td>6.835</td>
<td>21.127</td>
</tr>
<tr>
<td>5NN</td>
<td>0.798</td>
<td>0.057</td>
<td>6.835</td>
<td>0.305</td>
<td></td>
<td></td>
<td>0.775</td>
<td>0.060</td>
<td>26.724</td>
</tr>
</tbody>
</table>

Table 4.9: Comparison of estimators using simulated data under MNAR assumption: sample size $n = 100$, set size $m = 3$, iteration $k = 2,000$
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Stdev.</th>
<th>In-sample error rate(%)</th>
<th>Test error rate(%)</th>
<th>$\sqrt{MSE}$</th>
<th>$\sqrt{MSE}$ reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
<td>0.500</td>
<td>0.036</td>
<td></td>
<td></td>
<td>0.064</td>
<td></td>
</tr>
<tr>
<td>JPS(RF)</td>
<td>0.460</td>
<td>0.042</td>
<td></td>
<td>0.044</td>
<td>31.541</td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0.434</td>
<td>0.140</td>
<td>0.000</td>
<td>0.046</td>
<td>27.047</td>
<td></td>
</tr>
<tr>
<td>JPS(LDA)</td>
<td>0.475</td>
<td>0.037</td>
<td></td>
<td>0.046</td>
<td>27.786</td>
<td></td>
</tr>
<tr>
<td>LDA</td>
<td>0.465</td>
<td>0.044</td>
<td>26.295</td>
<td>0.047</td>
<td>25.946</td>
<td></td>
</tr>
<tr>
<td>JPS(SVM)</td>
<td>0.466</td>
<td>0.039</td>
<td></td>
<td>0.043</td>
<td>31.758</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>0.476</td>
<td>0.075</td>
<td>25.581</td>
<td>0.080</td>
<td>-26.529</td>
<td></td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.463</td>
<td>0.038</td>
<td></td>
<td>0.041</td>
<td>35.864</td>
<td></td>
</tr>
<tr>
<td>5NN</td>
<td>0.455</td>
<td>0.053</td>
<td>14.918</td>
<td>0.053</td>
<td>16.534</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.10: Comparison of estimators for book example data under MCAR assumption: sample size $n = 100$, set size $m = 3$, iteration $k = 5,000$
Chapter 5: JPS Applications in Two Examples

In this chapter, I will illustrate the JPS method in two real-world examples. One is the Electronic Health Records (EHR) data example. EHR systems are being adopted by many medical practices as a way of providing better health care, curbing medical expenses, and preventing mistakes in medical treatment. EHR systems contain information about patients’ medical histories, medications, allergies, diagnoses, and test results. Storing health records electronically facilitates bringing together information from many sources so that practitioners who access a patient’s records through an EHR have more accurate and complete information on which to base a treatment or diagnosis for the patient.

The 2010 Ohio Medicaid Health Information Technology Survey (HIT) was a survey of health care providers across the state conducted to estimate the proportion of Medicaid providers who had already adopted an EHR. The survey was conducted by The Ohio Colleges of Medicine Government Resource Center, The Ohio State University Statistical Consulting Service, and the Ohio Department of Job and Family Services. The sampling frame for the HIT Survey was the list of 7,973 Medicaid health care providers in the 2009 Medicaid Claims Data Provider file. In this example we will focus on the 2,746 primary care physicians included in the file. The
sampling frame included the provider’s location (exact address and county), number of office visits to the practice, number of patients in the practice, provider’s name, and, when available, provider’s email address, phone number, and fax number. The variable of interest in our example is whether or not a primary care physician has an EHR system. We code cases as a “1” if the practice has purchased an EHR and “0” if they do not have an EHR.

A simple random sample of 1,356 primary care physicians was selected and, on August 23, 2010, those physicians were mailed a survey with a postage-paid return envelope. Best survey practices were followed and sampled physicians who did not respond were sent reminder postcards, then a second copy of the survey, and a second reminder postcard. Four weeks into the data collection period researchers made follow-up phone calls to practitioners to encourage completion of the survey. Finally a fax was sent to all non-responding practices with a known fax number. Practices contacted by phone or fax had the option of completing the questionnaire over the phone or of returning the survey by mail, email, or fax. Following this intensive non-response follow-up effort, there were only 195 survey responses from the primary care physicians and 1,161 non-respondents.

Although a low response rate such as that obtained for the HIT survey does not necessarily mean nonresponse bias will affect the results, it is certainly a potential problem. Table 5.1 shows the summary statistics of the potential explanatory variables in the population, sample and respondent group. We notice that the respondent group is different from the population in terms of the percentage providing an email
address and the percentage providing a phone number/fax number in the data frame. The nonresponses are likely to be related to the auxiliary information that is available to us. For instance, during the follow up survey, those whose phone numbers or fax numbers were listed in the data frame could be more easily reached. This suggests a missing at random (MAR) mechanism for nonresponse [12]. It is also possible that the nonresponses are related to the EHR status of interest, corresponding to a missing not at random (MNAR) [12] nonresponse mechanism. The data summaries in Table 5.1 suggest that the variable of interest, say EHR status, may also differ in the respondent group and population. In the case of either MAR or MNAR missingness, we need to adjust the population proportion estimators to decrease or eliminate the nonresponse bias. We propose using JPS to help adjust for the missing data and compare our results to those using standard SRS methods.

The second example we use to demonstrate the effectiveness of JPS is the satellite data example. Different from the missingness caused by nonresponse in the EHR example, the missingness in the satellite data example is a result of the limitations of the measurement equipment. The problems with the EHR dataset are the insufficient amount of data (only 195 responses) and the nonrepresentativeness in the response. For the satellite data, the amount of available data is not the crucial problem. I will

<table>
<thead>
<tr>
<th></th>
<th>% has email</th>
<th>% has fax</th>
<th>% has phone</th>
<th>% in central Ohio</th>
<th># of practices</th>
<th>% owns EHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early respondents</td>
<td>0.608</td>
<td>0.775</td>
<td>0.961</td>
<td>0.137</td>
<td>102</td>
<td>0.441</td>
</tr>
<tr>
<td>Late respondents</td>
<td>0.609</td>
<td>0.874</td>
<td>0.954</td>
<td>0.207</td>
<td>87</td>
<td>0.517</td>
</tr>
<tr>
<td>All respondents</td>
<td>0.608</td>
<td>0.820</td>
<td>0.958</td>
<td>0.169</td>
<td>189</td>
<td>0.476</td>
</tr>
<tr>
<td>Sample</td>
<td>0.395</td>
<td>0.702</td>
<td>0.860</td>
<td>0.135</td>
<td>1345</td>
<td>unknown</td>
</tr>
<tr>
<td>Population</td>
<td>0.414</td>
<td>0.719</td>
<td>0.866</td>
<td>0.171</td>
<td>2746</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Table 5.1: Data summary for explanatory variables in HIT survey
introduce the background information about the satellite data collection and then describe the potential problems in the data.

NASA has launched a group of satellites that collect massive amounts of global climate data to support scientific studies in climate modelling and validation. Among those satellites, a convoy of “A-Train” satellites, as shown in Figure 5.1, has emerged as one of the most powerful tools scientists have for climate study. This formation of satellites, which currently includes the Aqua, CloudSat, Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) and Aura satellites, barrels across the equator each day at around 1:30 p.m. local time each afternoon, giving the satellite constellation its name; the “A” stands for “afternoon.” A history of the A-train satellite can be found at http://www.nasa.gov/mission-pages/a-train/a-train.html.

For studies of cloud properties, MODIS (the main instrument on Aqua), CloudSat and CALIPSO are the major players in the A-train collation. The observations collected by these three sensors greatly complement each other in cloud studies. The Aqua MODIS is a spectroradiometer that passively records radiation in 36 different spectral bands with a spatial resolution from 250 m to 1 km. Using the radiation data, MODIS records cloud properties over a large area. CloudSat and CALIPSO, two active sensors, follow Aqua by just 93 kilometers, about 12.5 seconds. As a result, CALIPSO’s lidar beam and CloudSat’s radar coincide on Earth’s surface about ninety percent of the time. Thanks to this special alignment in the A-train, these three instruments collect atmosphere information on a location at almost the same
time. CloudSat has a radar that is more than 1000 times more sensitive than any other existing weather radar. The CALIPSO lidar senses the cloud layers that the radar can not observe at higher spatial resolution. Together they provide information about the vertical structure of clouds and aerosols. The instruments are energy consuming, however, and they can only measure a small area at one time (http://cloudsat.atmos.colostate.edu/instrument).

Cloud detection using MODIS and CloudSat/CALIPSO data could play a significant role in global climate studies. Since even small changes in cloud proportion or
distribution have a great impact on climate, it is of great scientific interest to accurately estimate the cloud proportion or distribution. Figure 5.2 shows one MODIS image, with the CloudSat/CALIPSO path (a red line) overlaid, collected over the Pacific Ocean in 2007. In this dataset, there are 1,928 MODIS pixels that are spatially matched with the CloudSat/CALIPSO observations and 2,746,692 unmatched MODIS pixels (outside of the CloudSat/CALIPSO scanning path). As the image illustrates, the highly accurate cloud properties retrieved on the CloudSat/CALIPSO path might be combined with the MODIS radiation observations outside of the path to improve the cloud retrieval and cloud property studies on the whole MODIS scene.

In this thesis, I will focus on the problem of estimating the cloud proportion in the whole MODIS image, using the data collected by MODIS and CloudSat/CALIPSO. I will discuss more in Chapter 6 about the possibility of using the continuous response variable and applying the JPS method for the continuous cases.

5.1 The EHR Survey

In the EHR survey data, as previously noted, we have information on the sampling frame that includes the provider’s location (exact address and county), number of office visits to the practice, number of patients in the practice, provider’s name, provider’s email address, phone number, and fax number if available. We recode the location by dividing Ohio into two areas and with the central Ohio coded as a ‘1’ and the remaining area coded as ‘0’. We also create a new variable indicating whether or not the response was late, based on when nonresponse follow-up was conducted. If the responses were received before September 21st, we code cases as ‘0’. Responses
received after September 21st are coded ‘1’. Among the 195 respondents, 189 can be matched by the National Provider Identifier (NPI) or medicare provider number with the records in data frame. From Table 5.1, we notice that 87 of them are late respondents and the remaining are early respondents. We see from the table that early and late respondents have different characteristics from the population in terms
of the percentages having email, fax, phone listed in the data frame; these percentages are all higher for either respondent group than those of the full population. The respondents group does not appear to represent all the primary care providers in the population.

To apply JPS we first produce a ranking variable using a logistic regression model. We start with seven explanatory variables in the model: email listed or not, phone listed or not, fax listed or not, central area or not, number of office visits, number of patients, early response or not and practice type. The binary response variable is whether or not the practice has an EHR system. Logistic regression results show that phone listed or not and fax number listed or not are the two significant explanatory variables. To avoid too many ties in our ranking sets, we also select number of visits as the third explanatory variable in the model:

\[
\text{logit}(p(\text{hasEHR})) = 1.189 - 0.0001266 \times \text{visits} - 0.7269 \times \text{hasemail} - 0.7916 \times \text{hasfax}.
\]

We use this logistic model to estimate the probabilities of having an EHR system and rank these estimated probabilities in each set for JPS. From Table 5.1, we see that the missingness is very likely related to the explanatory variables such as phone and fax number listed in the frame or not. To encourage more practices to complete the survey, follow-up phone calls were made and a fax was sent to practices when possible. Thus we believe MAR would be a reasonable assumption. The SRS estimator for having an EHR is 0.476. If choose set size \( m = 3 \), and use all the responses as the sample where the sample size \( n \) is 189, we get that the JPS estimator is 0.538. The difference between two estimators suggests it is reasonable to use the JPS method.
In the satellite data problem, our goal is to estimate the cloud proportion in the whole MODIS image, that is, the percentage of MODIS pixels that are cloud contaminated. Due to the limitation of the instruments, the dataset doesn’t contain cloud contamination labels for most of the pixels. In this section, we will illustrate using the JPS method to better estimate the cloud proportion given this missing label problem.

For each pixel from the CloudSat/CALIPSO data, we first find the nearest MODIS pixel. It is possible that one MODIS pixel can be matched with multiple CloudSat/CALIPSO pixels. Our goal is to find the one to one matched pair of CloudSat/CALIPSO pixel and MODIS pixel. Thus, if one MODIS pixel has been matched with multiple CloudSat/CALIPSO pixels, we pick the nearest CloudSat/CALIPSO pixel and call this pair of CloudSat/CALIPSO and MODIS pixels a match. There are 1,928 MODIS pixels spatially matched with the CloudSat/CALIPSO observations. CloudSat/CALIPSO data have cloud proportion measurements so these are the locations we consider as labeled. The remaining 2,746,692 unmatched MODIS pixels have no nearby CloudSat/CALIPSO measurements, therefore, these are the unlabeled data. As a result, the labeled data proportion is lower than 1%.

The CloudSat/CALIPSO cloud proportion variable is a continuous variable between [0,1]. For the purpose of our example, we convert the proportion to a binary
variable. If the proportion of cloud cover is lower than 0.10, then the binary variable is 0 meaning very few or no clouds at that pixel. If the proportion is larger than 0.10, then the binary variable is 1 meaning this is cloud contamination at that pixel. The explanatory variables are all of the 36 spectral band values obtained by the MODIS instrument.

To better understand the missing data mechanism for this problem, we generate missingness data for testing purposes according to two models: MCAR and MNAR. For MCAR, we generate the missing data indicator by randomly splitting the data into a training set and testing set. In fact, the CloudSat/CALIPSO dataset only measures a path through the area that MODIS measures. The sampling design is close to taking a systematic sampling. If the cloud cover measured by the CloudSat/CALIPSO and the unknown cloud cover in the area that MODIS measures are similar, then the estimates of the population proportion from the learning dataset and testing dataset will be close. If the estimates are dissimilar, that is an indication that the missingness in the full dataset is nonignorable. Both MCAR and MNAR use only matched MODIS pixels before we apply JPS to the whole graph generated by the MODIS data. The advantage for doing so is that we know the true cloud proportion on the CloudSat/CALIPSO scanned line so estimators can be compared to this value to assess the accuracy.

We start with the MCAR dataset. We use a RF-based ranking procedure to create the aggregated explanatory variable. Since RF can handle multivariate explanatory variables easily, we use all 36 spectral band values obtained by the MODIS instrument.
as our explanatory variables for the RF model.

We randomly select 200 labelled observations as our sample. We calculate three types of estimators for all the ML methods we demonstrate in this subsection. The first type of estimator builds a classifier and applies it to \( nm \) data points (the same number as we use in the JPS method for fair comparison). The second type of estimator applies the same classifier to the whole dataset. The third uses the JPS method in addition to the standard ML procedures. We use the estimated probabilities of the cloud cover label indicating clouds in \( mn \) data points from the classifier to perform the ranking. Then we apply the JPS procedure to get the JPS estimator. Both methods one and three use \( mn \) data points while method two uses 1,928 the data points.

We first consider the RF estimates. We use 200 data points to train the RF model. We apply the RF classifier on the \( n \) sampled points as well as on \( n(m - 1) \) matched points to calculate the RF estimator, which we denote by \( \hat{p}_{RF} \). This is the JPS version of the first type of estimator. Since the whole dataset contains 1,928 matched MODIS/CALIPSO points, we can also apply the RF classifier to all the points to obtain the second type of estimator \( \hat{p}_{RF,all} \). To implement JPS, we use the RF classifier to predict the labels on the \( nm \) observed and matched points. Therefore for the \( m \) points in a set, we obtain an estimated success probability for each point where success is defined as having cloud cover. After getting ranks for all 200 sampled points, we post-stratify them based on rankings and calculate the JPS estimate, \( \hat{p}_{JPS,rf} \), using the formula
\[
\hat{p}_{JPS,x} = \frac{1}{m} \sum_{h=1}^{m} \hat{p}_h = \frac{1}{m} \sum_{h=1}^{m} \sum_{i=1}^{n} \frac{Y_i R_{ih}}{R_{ih}}.
\] (5.1)

To make comparisons, we also perform SVM on this dataset. For every selected SRS from the 1,929 data points, we train a SVM classifier based on 200 observations. Then this classifier is applied to a total of \(mn\) data points used in the JPS method. We use \(\hat{p}_{SVM}\) to represent the estimated population proportion. If we get the ranking based on the SVM method and perform JPS on 200 points, \(\hat{p}_{JPS,svm}\) can be calculated. We also apply the SVM classifier trained on 200 points to all 1,928 data points and get the predicted cloud cover as \(\hat{p}_{SVM,all}\). Similarly, 5 nearest neighbor and logistic regression are used on the same amount of data \(mn\) as JPS uses to obtain estimates \(\hat{p}_{5NN}\) and \(\hat{p}_{LOGISTIC}\). Then \(\hat{p}_{SVM,all}, \hat{p}_{5NN,all}\) and \(\hat{p}_{LOGISTIC,all}\) are calculated by predicting the whole dataset.

To compare the estimates across all methods described above, we use \(\sqrt{MSE}\), prediction bias, in-sample error rate and cross validation error rate as measures to assess the quality of the estimates. Among these measures, \(\sqrt{MSE}\) and prediction bias are our measurements for prediction accuracy. In-sample error rate and cross validation are used to select a better classifier. We repeat sampling 5,000 times and present the results in Table 5.2 and Figure 5.3.

In Table 5.2, we present the summary of our results in two sections. The left portion of the table shows the results for estimators using \(mn\) data points. Therefore we can compare the results from JPS combined with the various ML methods used for
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Stdev.</th>
<th>In-sample error rate(%)</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
<th>Mean</th>
<th>Stdev.</th>
<th>√MSE</th>
<th>√MSE reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
<td>0.730</td>
<td>0.038</td>
<td>0.035</td>
<td>0.041</td>
<td></td>
<td>0.716</td>
<td>0.035</td>
<td>0.023</td>
<td></td>
</tr>
<tr>
<td>JPS(RF)</td>
<td>0.717</td>
<td>0.035</td>
<td>0.031</td>
<td>0.041</td>
<td></td>
<td>0.718</td>
<td>0.031</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0.718</td>
<td>0.031</td>
<td>0.000</td>
<td>0.041</td>
<td></td>
<td>0.714</td>
<td>0.035</td>
<td>0.035</td>
<td>14.105</td>
</tr>
<tr>
<td>JPS(SVM)</td>
<td>0.722</td>
<td>0.039</td>
<td>0.034</td>
<td>0.042</td>
<td></td>
<td>0.722</td>
<td>0.035</td>
<td>0.035</td>
<td>14.358</td>
</tr>
<tr>
<td>SVM</td>
<td>0.728</td>
<td>0.039</td>
<td>0.034</td>
<td>0.042</td>
<td></td>
<td>0.728</td>
<td>0.039</td>
<td>0.039</td>
<td>12.941</td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.779</td>
<td>0.051</td>
<td>0.040</td>
<td>0.040</td>
<td></td>
<td>0.779</td>
<td>0.051</td>
<td>0.051</td>
<td>1.328</td>
</tr>
<tr>
<td>5NN</td>
<td>0.721</td>
<td>0.040</td>
<td>0.040</td>
<td>0.040</td>
<td></td>
<td>0.721</td>
<td>0.040</td>
<td>0.040</td>
<td>-31.319</td>
</tr>
<tr>
<td>JPS(LOGISTIC)</td>
<td>0.681</td>
<td>0.044</td>
<td>0.045</td>
<td>0.053</td>
<td></td>
<td>0.681</td>
<td>0.044</td>
<td>0.044</td>
<td>-31.319</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.717</td>
<td>0.040</td>
<td>0.040</td>
<td>0.040</td>
<td></td>
<td>0.717</td>
<td>0.040</td>
<td>0.040</td>
<td>-5.037</td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.779</td>
<td>0.051</td>
<td>0.051</td>
<td>0.051</td>
<td></td>
<td>0.779</td>
<td>0.051</td>
<td>0.051</td>
<td>1.328</td>
</tr>
<tr>
<td>5NN</td>
<td>0.721</td>
<td>0.040</td>
<td>0.040</td>
<td>0.040</td>
<td></td>
<td>0.721</td>
<td>0.040</td>
<td>0.040</td>
<td>-31.319</td>
</tr>
<tr>
<td>JPS(LOGISTIC)</td>
<td>0.681</td>
<td>0.044</td>
<td>0.044</td>
<td>0.053</td>
<td></td>
<td>0.681</td>
<td>0.044</td>
<td>0.044</td>
<td>-31.319</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.717</td>
<td>0.040</td>
<td>0.040</td>
<td>0.053</td>
<td></td>
<td>0.717</td>
<td>0.040</td>
<td>0.040</td>
<td>-5.037</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of estimators for satellite matched data under MCAR assumption: sample size $n = 200$, set size $m = 3$, iteration $k = 5,000$
ranking when using the same amount of data. The right column shows the results for estimators using all the data points. The differences between the same ML method using different amounts of data, can help us decide when is better to use the whole dataset.

As shown in the left column of Table 5.2, some supervised learning methods, such as 5NN and SVM, when using the same amount of data as JPS, can be even worse than the SRS estimator. Using the same method as a ranking function for the JPS method, however, gains more reduction in the $\sqrt{MSE}$. Except for RF, applying JPS
always reduces the $\sqrt{MSE}$ compared to the original ML method without JPS and is consistently better than SRS. Among all the methods, when we use $mn$ data points for each iteration, RF performs the best with a $\sqrt{MSE}$ reduction of 24% and JPS using RF as the ranking function is the second best with a $\sqrt{MSE}$ reduction of 14%. JPS(svm) is ranked third with a $\sqrt{MSE}$ reduction of 13%. If we take a look at the in-sample error rate, from RF, SVM to 5NN, as the in-sample error rate increases, the $\sqrt{MSE}$ decreases. Since our sampling method is MCAR here, the sample should represent the whole population and the result is intuitive. If the classifier does better for the sampled units, it will also classify other points in the population better. Therefore in-sample error rate is a good indicator for classification method selection if the missing data is MCAR.

The right column of Table 5.2 shows the results when all data points are used. Again RF performs best with an impressive $\sqrt{MSE}$ reduction of 44%. Comparing the same method using different amounts of data under MCAR, we can see the estimators using all the data points perform better than those using $mn = 600$ observations for all the methods. This further supports our conclusion from the in-sample error rate.

The boxplots in Figure 5.3 show the differences between each of our estimators and the true proportion. The RF estimators are centered closest to 0. The JPS estimator generally has a smaller standard deviation, especially if it has fewer large errors compared to other methods. Therefore we can conclude that the JPS method does reasonably well compared to the existing supervised learning methods under MCAR and always improves over the original method. Moreover, it is constantly better than
To mimic our real problem in the case of satellite data, instead of random sampling, we try systematic sampling as described previously. In the whole picture, because CloudSat/CALIPSO only measures cloud cover on an almost continuous line, a similar sampling design is to divide the line into segments and choose one of them as the sample. To implement systematic sampling, we choose one random starting point and continuously sample 200 points from that point onward. The SRS estimate is calculated based on these 200 points. Similar to the MCAR case in our simulation study, for the supervised learning methods we use 200 data points to train a classifier. One potential problem with the systematic sampling scheme is it is possible that the sample is too extreme, with all points labeled 1 or labeled 0, in which case we can not train a classifier. Naturally, without the ranking variable, JPS can not be applied to adjust the bias in such a sample. As long as the sample contains points from both classes, however, JPS can use the ranking information to adjust the imbalance in the sample and make it more closely represent the population.

As we did in the MCAR case, we calculate LDA, 5NN, SVM, RF and logistic regression estimators using $mn = 600$ data points and using the whole dataset. Then the JPS estimator is obtained for each method. We repeated sampling 5,000 times for the simulation study. The systematic sampling comparison is presented in Table 5.3 and Figure 5.4.
<table>
<thead>
<tr>
<th>Method</th>
<th>Estimators using ( mn ) units</th>
<th>Estimators using all the units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Stdev.</td>
</tr>
<tr>
<td>SRS</td>
<td>0.633</td>
<td>0.273</td>
</tr>
<tr>
<td>JPS(RF)</td>
<td>0.664</td>
<td>0.221</td>
</tr>
<tr>
<td>RF</td>
<td>0.675</td>
<td>0.241</td>
</tr>
<tr>
<td>JPS(SVM)</td>
<td>0.647</td>
<td>0.212</td>
</tr>
<tr>
<td>SVM</td>
<td>0.673</td>
<td>0.303</td>
</tr>
<tr>
<td>JPS(5NN)</td>
<td>0.629</td>
<td>0.280</td>
</tr>
<tr>
<td>5NN</td>
<td>0.632</td>
<td>0.314</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of estimators for satellite matched data under MNAR assumption: sample size \( n = 200 \), set size \( m = 3 \), iteration \( k = 2,000 \) (excludes all 0 and all 1 samples)
In this systematic sampling, 2,659 out of 5,000 iterations resulted in samples with all observations labeled either 0 or 1. For these samples none of the methods can make any improvement. The only estimator we can obtain is the SRS estimator. We are more interested in the cases where the sample is extremely biased but various methods can still make an improvement. Therefore we delete the all 0 and all 1 samples and randomly select 2,000 samples among the remainder. From Table 5.3, we can see that JPS(SVM) performs the best among all the methods we tried. It provides a $21.7\% \sqrt{MSE}$ reduction compared to SRS. The second best estimator is JPS(RF) with a $\sqrt{MSE}$ reduction of 20.5%. We also notice that when the sample (200 observed points) is really biased, therefore, very different from the population
(all 1,928 points) due to our sampling method, the classifiers we trained from the sample using RF, 5NN and SVM are not good classifiers for the whole dataset. In fact, $\hat{p}_{RF, all}$, $\hat{p}_{5NN, all}$ and $\hat{p}_{SVM, all}$ all do worse than $\hat{p}_{RF}$, $\hat{p}_{5NN}$ and $\hat{p}_{SVM}$. The table also gives the in-sample error rate for RF, SVM and 5NN. We notice the trend that we saw in the MCAR case no longer holds here. The relationship between the in-sample error rate and $\sqrt{MSE}$ is no longer strictly positive. This demonstrates our belief that when the sample is not a good representation of the population, the classifier performances based on the sample can not be generalized to the whole population.

In summary, the JPS method has a fairly stable performance under both MCAR and systematic sampling situations. It does better than most of the supervised learning methods using the same amount of data. When JPS is applied based on the supervised learning method, it usually provides a better estimator with a greater $\sqrt{MSE}$ reduction.
Chapter 6: Contributions and Future Work

This dissertation focuses on exploring three ideas: the ranking process for the JPS method, the comparison of JPS with other ML techniques and the JPS performance under three standard missing data mechanisms.

The first contribution is the ranking process. When the JPS ranking process cannot be done by experts or visual inspection, we find a way to use the auxiliary variable itself (for the univariate case) or model the auxiliary variables to construct a ranking variable, enabling us to use one or more auxiliary variables in a common procedure. This expands the application of the JPS methods to multivariate auxiliary variables cases as illustrated in Chapter 4.

The second contribution is the comparison of JPS with other ML techniques. For this purpose, we show the comparison results in the bivariate simulation in Section 4.2. We construct three types of estimates for comparison. The simplest are the ML methods (RF, SVM, KNN and LDA) using all the labeled and unlabeled data. We also calculate the JPS estimate based on rankings produced by various ML methods using \( mn \) data points. In addition, we consider applying ML methods using the same data as JPS. From this simulation, we can see JPS can be built on
any predictive method. Some of the ML methods we considered perform even worse than SRS. By applying JPS, however, under the condition of using the same set of sampled data, the JPS method can improve over the original method and reduce the $\sqrt{MSE}$.

The third goal is to build the connection between three standard missing data mechanisms and data mining techniques. Missing data has been studied extensively in the survey sampling literature. This is the first time that we are aware of when researchers completed a systematic study of the effects of missing data in the data mining setting. In this thesis, we review the missing data in Chapter 2 and conduct simulations in Chapter 4 to demonstrate the effects of missing data in a data mining setting. In each simulation, we generate different missing indicator functions $f(m)$ to select samples for MCAR, MAR and MNAR mechanisms. Then we apply the methods of interest for comparison. We notice the JPS method reduces the standard deviation to achieve a smaller $\sqrt{MSE}$ when the missing data is MCAR. Under the MAR or MNAR assumption, JPS reduces the bias in the estimate to obtain a smaller $\sqrt{MSE}$ therefore outperforming SRS. Although MAR and MNAR are hard to distinguish through the marginal distribution $f(x)$, we can see clearly that the performances of various methods are very different for these two.

In Chapter 5 we have illustrated how to apply the JPS method in two real-life examples. The first is the Electronic Health Records (EHR) data example. The low response rate and the differences in the responders and nonresponders make this a perfect case to apply the JPS method. The second example is the satellite data cloud
cover prediction problem. The limitation of the measurement instruments creates the missingness in the data. Thus we apply the JPS method to adjust for the potential bias in the estimate.

Beyond this dissertation, there are some unresolved problems that might be interesting to investigate. In previous sections, exploratory analysis and some simulation studies gave us an idea about how set size affects estimates when JPS is applied. Current conclusions are based on simple simulations and indirect analysis. To make statistical inference about the effect of set size on the performance of the JPS method, more well-designed simulation studies and theoretical work are needed. We want to develop a framework for evaluating the gains from JPS under different distribution separation, proportion, and sample size assumptions. We are also interested to know if JPS is like other semi-supervised learning methods, where unlabeled data helps in some cases but not always. It would be valuable to quantify the effect of unlabeled data versus labeled data.

Another possible interesting direction we can explore is analyzing continuous response variables. So far all the datasets and applications we explored using JPS assume binary response variables. There are, of course, many other cases in which the response variable may be continuous. Instead of proportion estimation, we could easily transform the JPS method for mean estimation. We would expect the similar performance from the JPS method in that case. When the response variable is continuous, other than the mean estimation, we could further study the cumulative
density function by choosing different cut off values for the continuous response variable. Given the same amount of available information, we would learn more about the response by extending the application of the JPS method.
Bibliography


