Investigating Unsupervised Machine Learning Methods for Datasets with Missing Values

Connor J. Charlton

A Report presented for the degree of
Bachelor of Mathematical Sciences

Department of Mathematics and Computer Science
Durham University
United Kingdom
April 2022

Declaration

This piece of work is a result of my own work except where it forms an assessment based on group project work. In the case of a group project, the work has been prepared in collaboration with other members of the group. Material from the work of others not involved in the project has been acknowledged and quotations and paraphrases suitably indicated.

Use of publicly available software packages is indicated by the library(.) code lines at the top of each R script and not stated elsewhere. Likewise LaTeX code used for the formatting of much of this document is taken from publicly available templates, in particular much of it comes from the Durham University thesis template.

Acknowledgements

I would like to thank Dr Tahani Coolen-Maturi for supervising me during this project, and in particular for the freedom she gave me in allowing me to explore the field of unsupervised machine learning and select the topic of my report with few limitations and for the guidance she gave me in directing me to valuable research resources and helping me to navigate the field.
I would also like to thank Dr Emmanuel Ogundimu and Dr Sebastian Schmon for the insight they provided me, during discussions following my mid-term project presentation, with regard to the generality of MICE imputation methods. Knowledge I gained from this conversation had a significant influence on the direction of my subsequent research.

Contents

List of Figures

CHAPTER 1

Introduction

Machine learning is a subfield of artificial intelligence that has become ubiquitous across STEM fields, having applications in medical diagnosis, market segmentation, financial asset/derivative/currency trading, computer vision, robotics, speech and image recognition, automatic language translation, game strategy optimisation and many other areas. Interestingly as well, machine learning is a relatively new field of research, compared to other areas of mathematics and statistics, having only been developed over the course of the last century or less. Indeed the term *Machine learning (ML)* was not first coined until 1959 and there has been some dispute over the years regarding precisely how to define the term. The term is now most popularly defined, due to Tom Mitchell, by:

Definition 1.1 (Machine Learning) A computer program[me] is said to learn from experience E with respect to some class of tasks T, by some performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. [1]

Within the field of machine learning, there are two main sub-fields - supervised machine learning and unsupervised machine learning. *Unsupervised machine learning* is the process of inferring information about the distribution, $F(\mathbf{x})$, of some p-dimensional random variable, $\mathbf{X}$, (the task T) using a dataset, $X_{n\times p}$, comprised of n unlabelled observations, $\mathbf{x_i}$, of $\mathbf{X}$ (the experience E). This is in contrast to *supervised machine learning*, in which the goal is to infer information about the conditional distribution, $F(y|\mathbf{x})$, (the task T) using a dataset, $\{\mathbf{y}, X_{n\times p}\}$, comprised of n labelled observations, $\{y_i, \mathbf{x}_i\}$ (the experience E). However, importantly, - in both cases - the goal is not necessarily to determine the distribution ($F(\mathbf{x})$ or $F(y|\mathbf{x})$), but often instead to determine certain features of the distribution. For example, association rules analysis (ARA) is a field of unsupervised machine learning which seeks to determine subsets or pairs of subsets of the space, $\mathcal{X}$, of all possible values of $\mathbf{X}$, which have a high probability of occurring or whose two constituent subsets jointly have a high probability of occurring, or for which one of the subsets has a high conditional probability of occurring given the other one occurs. [1] Similarly, cluster analysis (CA) is a field of unsupervised machine learning which seeks either

to segment a population of instances into k clusters (possibly under some constraints regarding the maximum or minimum permitted number of instances per cluster) such as to minimise the total of the distances/dissimilarities (under some specified distance function) between all same-cluster instances, or more ambitiously to determine the most likely number of component distributions of $F(\mathbf{x})$, when $F(\mathbf{x})$ is suspected to be a mixture distribution and to determine (with as high a degree of certainty as possible) which instances were drawn from which component distribution.

There is an abundance of literature on the application of methods to solve problems related to these two fields (ARA and CA) to complete datasets (one's which are not missing any values). However, in practice datasets are rarely complete, and so in most real-world scenarios, we must either: adapt the machine learning methods to support missing values; or, in conjunction with applying the required machine learning methods, employ methods for resolving the issues resulting from the absence of values. In the latter chapters of this report, many such approaches are investigated. However, first the theory behind methods in the fields of association rule analysis and cluster analysis must be described and explained in the context of their application to complete datasets.

CHAPTER 2

Fields of Unsupervised Machine Learning

2.1 Association Rules Analysis

Association rules analysis (ARA) is a field of unsupervised machine learning that seeks to identify regions (or pairs of regions) of the $\mathcal{X}$ space that have a high probability (or conditional probability) of occurring. More specifically ARA can be thought of as a class composed of two sub-classes of problems:

Class I We wish to identify as many as possible of the regions $\mathcal{X}_i$ of $\mathcal{X}$ which have more than a $t \in [0, 1)$ probability of occurring (where t is some user-specified or problem specific value).

Class II We segment the space $\mathcal{X}$ of possible values of the random vector $\mathbf{X}$ into sub-spaces $\mathcal{X}_1, ..., \mathcal{X}_p$ of possible values of the random components $\mathbf{X}_1, ..., \mathbf{X}_p$ of $\mathbf{X}$ respectively. We then wish to identify ordered pairs $\mathcal{X}_i = (\mathcal{X}_{i,\text{ant}}, \mathcal{X}_{i,\text{con}})$ of subsets of $\mathcal{X}$ for which $\mathcal{X}_{i,\text{ant}} \subseteq \bigcup_{i \in \mathcal{P}^*} \mathcal{X}_i$, $\mathcal{X}_{i,\text{con}} \subseteq \bigcup_{i \notin \mathcal{P}^*} \mathcal{X}_i$ for some $\mathcal{P}^* \subset \{1, ..., p\}$ (i.e. $\mathcal{X}_{i,\text{ant}}$ and $\mathcal{X}_{i,\text{con}}$ correspond to sub-regions of the spaces of possible values of different components of $\mathbf{X}$) and for which both...

a) ...$\mathbb{P}\left(\sum_{i \in \mathcal{P}^*} \mathbf{X}_i \mathbf{e}_i \in \mathcal{X}_{i,\text{ant}}, \sum_{i \notin \mathcal{P}^*} \mathbf{X}_i \mathbf{e}_i \in \mathcal{X}_{i,\text{con}}\right) > t$ and...

b) ...for which $\mathbb{P}\left(\sum_{i \notin \mathcal{P}^*} \mathbf{X}_i \mathbf{e}_i \in \mathcal{X}_{i,\text{con}} \,\middle|\, \sum_{i \in \mathcal{P}^*} \mathbf{X}_i \mathbf{e}_i \in \mathcal{X}_{i,\text{ant}}\right) > c$.

We call $\mathcal{X}_{i,\text{ant}}$ the *antecedent* and $\mathcal{X}_{i,\text{con}}$ the *consequent*. For a class I problem, we call the probability, that an observation x of $\mathbf{X}$ lies within the required set $\mathcal{X}_i$, the *support* of $\mathcal{X}_i$ and denote it $\mathcal{T}(\mathcal{X}_i)$. For a class II problem we define the probability expression in **a)** as the *support* of $\mathcal{X}_i$ - also denoted $\mathcal{T}(\mathcal{X}_i)$ - (notice that it is conceptually the same as the definition for a class I problem, since both refer in some sense to the probability that the observation lies within the required set) and define the probability expression in **b)** as the *confidence* of $\mathcal{X}_i = (\mathcal{X}_{i,\text{ant}}, \mathcal{X}_{i,\text{con}})$, which we denote $\mathcal{C}(\mathcal{X}_{i,\text{ant}} \Rightarrow \mathcal{X}_{i,\text{con}})$.[1] For a

[1] There are actually several other frequently used measures of interestingness for filtering pairs of subsets - most notably the lift, conviction and leverage - but for simplicity, we will filter using only support and confidence.

class I problem, we call the requirement that all sought sets have support strictly greater than t the *conjunctive rule*, and similarly for a class II problem, we collectively call the set of requirements **a)** and **b)** the *conjunctive rule*; so in both cases, the conjunctive rule is essentially the requirement that distinguishes sought-after sets from non-sought-after sets.

If $\mathbf{X}$ is a discrete (all entries are categorical or ordinal) random variable, then, at a theoretical level, both classes of problems are trivial to solve; we simply list all possible subsets or pairs of subsets (under the specified constraints) of $\mathcal{X}$ and compute the relevant sample frequencies and use them as our estimates of the required probabilities. However, this is computationally infeasible for large datasets; for example consider a class I problem (the simpler class of problem) for which $(\forall i \in \{1, ..., p\})$ the i^{th} component of $\mathcal{X}$ can take p_i distinct values. The number of possible subsets of $\mathcal{X}$ (and therefore the number of subsets for which the sample frequency must be calculated) is $\prod_i (2^{p_i} - 1)$.

So, for example, if the variables $\mathbf{X}_1, ..., \mathbf{X}_p$ represented a variety of medically significant attributes such as prevalence of q medical conditions, number of hours of exercise per day, demographics (age, ethnicity, sex), ONS occupational group (of 1463 possible groups) etc., then for some variables, $p_i \gg 2$ and importantly the number of possible subsets of $\mathcal{X}$ is at least $3^{q+2}.\left(2^{24} - 1\right).\left(2^{119} - 1\right).\left(2^{1463} - 1\right)$, with this value exploding rapidly as new variables, with a large cardinality of possible values, are added. In particular note that, in computational terms, the binary variables increase this value very little, while the variables that can take multiple values (such as occupational group) greatly impact it, to the extent that it becomes computationally infeasible to calculate. So in order to yield results the goals of our analysis must be less ambitious so that the algorithms used can be more efficient.

2.1.1 Market basket analysis

In market basket analysis, the problem is simplified by only considering subsets $s_i \subseteq \mathcal{X}_i$ of the form $s_i = v_{0i} \in \mathcal{X}_i$ or $s_i = \mathcal{X}_i$. (Note that sometimes the conditions are defined reversely so that the former condition becomes $s_i = \mathcal{X}_i \setminus v_{0i}$ but the maths is still the same.) In our example, this means reducing every variable, which can take a large cardinality of possible values, to a binary random variable, so for example, age may become a binary variable indicating whether a person is aged under 18 or over (inclusive) 18 and occupational group may become an indicator variable for whether or not a person works a desk (or otherwise sedentary) job. Of course, one could remove several variables or split certain variables into 4, 8, 16 etc. variables and so the number of new binary variables (which we will denote K) does not necessarily always (although often does) equal the original number of variables p. Either way, the variables resulting from this dimensional reduction of the output space are analogous to variables indicating the presence or absence of specific retail product lines in a customer transaction at a super market (or other retail outlet), hence why this methodology is known as 'market basket' analysis. The analysis then analogously corresponds to finding product lines that are frequently bought as part of the same transaction (and - not simultaneously - trying to explicitly find product lines that are very infrequently bought together). In the case of the above example, assuming we bisect every variable space exactly once (so that $K = p$), this reduces the number of subsets for which to calculate the sample frequency to 2^p. So ultimately, market basket analysis reduces the problem down to finding subsets $\mathcal{K} \subset \{1, ..., K\}$ such that, for Z_k defined to be an indicator random variable for the inclusion

of the k^{th} random variable ($\forall k \in \mathcal{K}$),

$$\mathcal{T}(\mathcal{K}) := \mathbb{P}\left(\bigcap_{k\in\mathcal{K}} (Z_k = 1)\right) = \mathbb{P}\left(\prod_{k\in\mathcal{K}} Z_k = 1\right) > t \tag{2.1}$$

And, if the problem is a class II problem, finding partitions $\mathcal{K} = \mathcal{K}_{\text{ant}} \oplus \mathcal{K}_{\text{con}}$ such that,

$$\mathcal{C}(\mathcal{K}_{\text{ant}} \Rightarrow \mathcal{K}_{\text{con}}) := \mathbb{P}\left(\bigcap_{\substack{k_1\in\mathcal{K}_{\text{ant}}\\ k_2\in\mathcal{K}_{\text{con}}}} (Z_{k_2} = 1 \mid Z_{k_1} = 1)\right) = \mathbb{P}\left(\prod_{\substack{k_1\in\mathcal{K}_{\text{ant}}\\ k_2\in\mathcal{K}_{\text{con}}}} (Z_{k_2} \mid Z_{k_1} = 1) = 1\right) > c \tag{2.2}$$

We estimate these subsets by calculating the corresponding sample frequencies[2]:

$$\hat{\mathcal{T}}(\mathcal{K}) = \hat{\mathbb{P}}\left(\prod_{k\in\mathcal{K}} Z_k = 1\right) = \frac{1}{n}\sum_{i=1}^{n}\prod_{k\in\mathcal{K}} z_{ik} \tag{2.3}$$

$$\begin{aligned}\hat{\mathcal{C}}(\mathcal{K}_{\text{ant}} \Rightarrow \mathcal{K}_{\text{con}}) &= \hat{\mathbb{P}}\left(\prod_{\substack{k_1\in\mathcal{K}_{\text{ant}}\\ k_2\in\mathcal{K}_{\text{con}}}} (Z_{k_2} \mid Z_{k_1} = 1) = 1\right)\\ &= \frac{1}{\#\left\{\begin{array}{c} i\in\{1,\ldots,n\}:\\ \left(\prod_{k_1\in\mathcal{K}_{\text{ant}}} z_{ik_1}\right) = 1\end{array}\right\}}\left[\sum_{\substack{i\in\{1,\ldots,n\}:\\ \left(\prod_{k_1\in\mathcal{K}_{\text{ant}}} z_{ik_1}\right)=1}}\left(\prod_{k_2\in\mathcal{K}_{\text{con}}} z_{ik_2}\right)\right]\end{aligned} \tag{2.4}$$

Thus the solution will be the set of sets,

$$\{\mathcal{K} \mid \mathcal{T}(\mathcal{K}) > t\}$$

or (if it's a class II problem),

$$\{\{\mathcal{K}_{\text{ant}},\ \mathcal{K}_{\text{con}}\} \mid \mathcal{T}(\mathcal{K}_{\text{ant}} \oplus \mathcal{K}_{\text{con}}) > t,\ \mathcal{C}(\mathcal{K}_{\text{ant}} \Rightarrow \mathcal{K}_{\text{con}}) > c\}$$

For each rule mined, we refer to the total number of items in the rule (the total across both the antecedent and consequent subsets) as the *order* of the rule.

2.1.2 Association rules analysis algorithms

When constructing an algorithm to search for sufficient support/confidence regions (which we will refer to as desired regions) one may naively think to construct a brute force algorithm to search every viable region. However, there are algorithms that are much less computationally expensive, but obtain the same or similar results. One such algorithm is the apriori algorithm.

[2] z_{ik} denotes the i^{th} instance (observed value) of the indicator variable Z_k

Apriori algorithm

The apriori algorithm reduces the number of required computations by utilising a breadth-first search method. So for solving class I problems, it works thusly:

1. Decide on an appropriate value of $t \in [0, 1)$.
2. Calculate $\hat{\mathcal{T}}(\mathcal{K})$ (using equation (2.3)) for all singleton sets $\mathcal{K}$ (i.e. sets of the form $\mathcal{K} = \{k^*\}$) and record those sets for which $\hat{\mathcal{T}}(\mathcal{K}) > t$.
3. For all possible doubleton sets (i.e. sets of the form $\mathcal{K} = \{k_1^*, k_2^*\}$) that can be constructed by taking the union of sets recorded at the conclusion of step 2., calculate $\hat{\mathcal{T}}(\mathcal{K})$ and again record those sets for which $\hat{\mathcal{T}}(\mathcal{K}) > t$.
4. For all possible tripleton sets (i.e. sets of the form $\mathcal{K} = \{k_1^*, k_2^*, k_3^*\}$) that can be constructed by taking the union of sets recorded at the conclusions of the previous steps, calculate $\hat{\mathcal{T}}(\mathcal{K})$ and again record those sets for which $\hat{\mathcal{T}}(\mathcal{K}) > t$.
5. Repeat the above process iteratively for sets of cardinality 4, 5, 6, ..., n or until the set cardinality requirement is high enough that no set $\mathcal{K}$ can be constructed with $\hat{\mathcal{T}}(\mathcal{K}) > t$.
6. Return all sets $\mathcal{K}$ that were recorded at the conclusion of any of the previous steps.

In the case of a class II problem, this algorithm is adapted by additionally specifying a confidence requirement $t \in [0, 1)$ in step 1. and then applying that confidence requirement in synchrony with the application of the support requirement in steps 2. to 5., where obviously the sets in each step are substituted with pairs of subsets. However, only single item consequent sets are considered. There are also other search algorithms, such as the Eclat algorithm and FP-Growth algorithm [2]. However, those will not be explored in this report.

One of the issues with the apriori algorithm is that it is fundamentally discrete and thus attempting to find high support regions in continuous multi-dimensional space is only possible via this approach if the space is first itemised, by partitioning the space into regions, and while this is not a totally flawed approach, it does raise the question of how best to partition the space, and how to justify that partitioning in a way that isn't purely arbitrary.

Apriori example application

We now demonstrate this algorithm on, what will become, a running example throughout this report involving a simulated film review dataset[3], so we begin by summarising the dataset. This dataset was

[3]See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Datasets.zip` to view this dataset (along with all its sparsified versions).

constructed by drawing instances from the mixture distribution, [4]

$$\mathbf{X_{1:10}} = \begin{cases} \mathbf{X^1}_{\mathbf{1:10}}, & Z = 1 \\ \mathbf{X^2}_{\mathbf{1:10}}, & Z = 2 \\ \mathbf{X^3}_{\mathbf{1:10}}, & Z = 3 \end{cases}$$

where, Z is a discrete random variable taking values 1, 2 or 3 each with probability $\frac{1}{3}$ and, $\forall i \in \{1, ..., 10\}$,

$$\mathbf{X_1}_i \sim \mathcal{N}\left(5 - \tfrac{i}{3}, \tfrac{3}{2}\right) \quad \mathbf{X_2}_i \sim \mathcal{N}\left(\tfrac{3}{2} + \tfrac{i}{2}, 1\right) \quad \mathbf{X_3}_i \sim \exp\left(\tfrac{6}{11-i} + \tfrac{1}{2}\right)$$

Roughly speaking this corresponds to simulating a pseudo-realistic scenario in which there exist three populations of film viewers each with similar film interests, where each of these three populations are of equal size or at least not thought to be of different sizes (in the Bayesian paradigm), and we obtain the dataset by drawing randomly from the entire population of film viewers. Thus data obtained is meant to simulate the ratings viewers would give for each film on a sliding $1 - 5$ scale. However, to ensure we have a categorical variable in our analysis, we append to $\mathbf{X_{1:10}}$ the Boolean random variable,

$$X_{11} = \text{Bernoulli}\left(\frac{\overline{X_{1:10}}}{5}\right)$$

which is meant to represent a premium subscription indicator random variable - in other words, a random variable taking value TRUE (represented with value 1) for viewers who own a premium subscription to this hypothetical streaming service and taking value FALSE (represented with value 0) for viewers who don't. We model the probability that a given person would purchase a premium subscription as being proportional to their average satisfaction with (rating of) the films on offer, since it seems intuitive that the probability that someone would purchase a premium subscription to a film streaming service would be positively correlated with how highly they rate the available films and linear proportionality is an easy relationship to model. [5]

Figure 2.1 illustrates the distribution of the confidence and support of association rules mined from this simulated film review dataset by the apriori algorithm, when the film review variables were discretised into unit intervals (i.e. $[0, 1), [1, 2), [2, 3), [3, 4), [4, 5]$). We can see there are far more rules with low support than high support, but interestingly the same is not true of confidence; there appear to be similarly many high confidence rules as low confidence rules. Neither of these observations should surprise us; the first is because the distribution from which we simulated the values has a global minimum marginal probability density (for each variable individually) of approximately 0.0307 only about 6.5 times less than the probability density of a uniform distribution over the same interval. Hence, there cannot be very many regions of high probability density, and therefore cannot be many rules with high support. The second is due to the built in dependencies between components of the

[4] Superscripts here do *not* denote exponentiation

[5] See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Film%20review%20dataset%20simulator.R` for the code used to generate the simulated film review dataset.

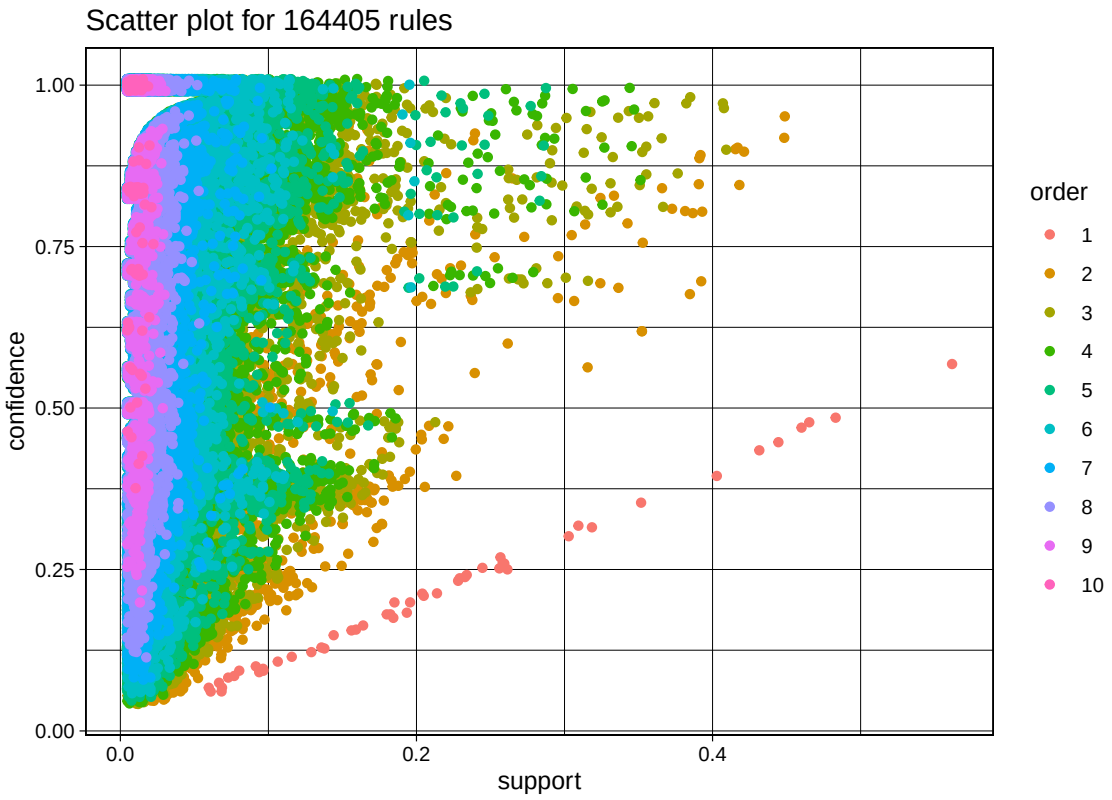

Figure 2.1: Graph showing all association rules mined from the simulated film review dataset by the apriori algorithm with support greater than 0.01 and confidence greater than 0.05.

distribution.

Another, expected phenomenon that we observe is that there are far more low order rules than high order rules and that the high order rules all have low support, whereas there are many low order rules with high support. This is due to higher order rules, by definition, describing a smaller subset of the variable space and so necessarily having lower support on average.

We also notice the odd phenomenon that there appear to be empty indents on the left hand side of the graph (most noticeably the indent in the top left). This is, again however, easily explainable because for rules with very low support (on the far left of the graph) the confidence is calculated as a fraction with a small integer denominator (and obviously an integer numerator that can be no greater than the denominator) thus reducing the granularity of confidence values and making their discrete nature more obvious to the naked eye. Another distinctive but explainable phenomenon is the high density of order 1 rules along the identity function, support = confidence. This is because for order 1 rules, the confidence is identical to the support, since for the rule to be meaningful, the one item in

the rule must be in the consequent set and thus, the rule is of the form $\{\} \Rightarrow \{k^*\}$ and so,

$$\begin{aligned}
\hat{\mathcal{C}}\left(\{\} \Rightarrow \{k^*\}\right) &= \frac{1}{\#\left\{\begin{array}{c} i \in \{1,...,n\}: \\ \left(\prod_{k_1 \in \{\}} z_{ik_1}\right) = 1 \end{array}\right\}} \left[\sum_{\substack{i \in \{1,...,n\}: \\ \left(\prod_{k_1 \in \{\}} z_{ik_1}\right)=1}} \left(\prod_{k_2 \in \{k^*\}} z_{ik_2}\right)\right] \\
&= \frac{1}{n}\left[\sum_{i \in \{1,...,n\}} \left(\prod_{k_2 \in \{k^*\}} z_{ik_2}\right)\right] \\
&= \frac{1}{n}\left[\sum_{i \in \{1,...,n\}} z_{ik^*}\right] \\
&= \hat{\mathcal{T}}\left(\{k^*\}\right) \\
&= \hat{\mathcal{T}}\left(\{\} \oplus \{k^*\}\right)
\end{aligned}$$

Of course, the results of the apriori algorithm analysis are limited in their granularity of detail because, the algorithm can necessarily only mine rules linking our choice of discretised regions of what was originally a continuous space, and so, for example, would never find a rule such as $\{\text{film\_seven} \in [3.25, 5)\} \implies \{\text{premium\_subscription} = 1\}$. Nonetheless, for the purposes of our exploration of missing value imputation methods for facilitating the application of association rules analysis methods, this association rules analysis method will suffice. Indeed, much like the substitution of the apriori algorithm, for the brute force algorithm, the discretisation of the variable space actually makes the analysis much less computationally expensive and so is more practical in a variety of practical applications with continuous variable spaces. However, the question that is naturally raised and thus the question we, for completeness, must still answer is: is there an ARA method for continuous random variables that maintains the continuity of the spaces spanned by those variables?

2.1.3 Association rules analysis via supervised learning methods

It turns out there are multiple and, although association rules analysis is a field of unsupervised machine learning, one of the continuity-preserving ARA methods involves the use of supervised machine learning methods.

In this ARA method, we seek to estimate the PDF of $\mathbf{X}$ (from which any subsequent support and confidence calculations can be made very easily). To do so, we first select a reference PDF $g_0(\mathbf{x})$ from which we generate n values via Monte Carlo methods and then by randomly combining these n values with the n values drawn from $\mathbf{X}$ we produce a mixed sample of $2n$ values that have effectively been drawn from the distribution defined by PDF $\frac{g(\mathbf{x})+g_0(\mathbf{x})}{2}$. We can then define random variable Y as an indicator random variable for observations drawn from this mean distribution; so Y takes value $Y = 1$ if the drawn value was originally drawn from $g(\mathbf{x})$ and takes value $Y = 0$ if originally drawn from

$g_0(\mathbf{x})$. We then define the function $\mu : \mathcal{X} \to \{0, 1\}$ by,

$$\mu(\mathbf{x}) := \mathbb{E}(Y \mid \mathbf{x}) = \frac{g(\mathbf{x})}{g(\mathbf{x}) + g_0(\mathbf{x})} = \frac{g(\mathbf{x})/g_0(\mathbf{x})}{g(\mathbf{x})/g_0(\mathbf{x}) + 1}$$

$$\implies \frac{g(\mathbf{x})}{g_0(\mathbf{x})} = \frac{\mu(\mathbf{x})}{1 - \mu(\mathbf{x})}$$

$$\implies \log\left(\frac{\hat{g}(\mathbf{x})}{g_0(\mathbf{x})}\right) = \log\left(\frac{\hat{\mu}(\mathbf{x})}{1 - \hat{\mu}(\mathbf{x})}\right) \quad \text{or equivalently} \quad \hat{g}(\mathbf{x}) = g_0(\mathbf{x})\frac{\hat{\mu}(\mathbf{x})}{1 - \hat{\mu}(\mathbf{x})}$$

Preliminary overview of logistic regression

Calculating a suitable estimate $\hat{\mu}(\mathbf{x})$ of $\mu(\mathbf{x})$ is an example of a classification problem. The study of such problems is known as *classification analysis* and a popular method in the field of classification analysis is logistic regression. The goal of logistic regression is to model the discrete response variable G thusly: [3]

$$\log\left(\frac{\mathbb{P}(G = k \mid \mathbf{X} = \mathbf{x})}{\mathbb{P}(G = K \mid \mathbf{X} = \mathbf{x})}\right) = \beta_{k0} + {\beta_k}^T\mathbf{x}$$

$$\text{for some coefficient matrix} \qquad \beta = \begin{pmatrix} \beta_{10} & \beta_{11} & \cdots & \beta_{1K} \\ \beta_{20} & \beta_{21} & \cdots & \beta_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{(K-1)0} & \beta_{(K-1)1} & \cdots & \beta_{(K-1)K} \end{pmatrix} \tag{2.5}$$

(where β_k is the vector formed by taking the k^{th} row of β, but omitting the 0^{th} column).
Or if we define, $\mathbf{k} := (1, ..., K-1)$, then,

$$\log\left(\frac{\mathbb{P}(G = \mathbf{k} \mid \mathbf{X} = \mathbf{x})}{\mathbb{P}(G = K \mid \mathbf{X} = \mathbf{x})}\right) = \beta^T \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \tag{2.6}$$

This corresponds to estimating the log odds, but should the probabilities be preferable it follows from the above (and the requirement that probabilities sum to 1) that,

$$\forall k \in \{1, ..., K-1\}, \qquad \mathbb{P}(G = k \mid \mathbf{X} = \mathbf{x}) = \exp\left(\beta_{k0} + \beta_k^T x\right)\mathbb{P}(G = K \mid \mathbf{X} = \mathbf{x})$$

But clearly, $\sum_{k=1}^{K}\mathbb{P}(G = k \mid \mathbf{X} = \mathbf{x}) = 1 \implies \mathbb{P}(G = k \mid \mathbf{X} = \mathbf{x}) = \dfrac{1}{1 + \sum_{k=1}^{K-1}\exp\left(\beta_{k0} + \beta_k^T x\right)}$ and thus,

$$\forall k \in \{1, ..., K-1\}, \qquad \mathbb{P}(G = k \mid \mathbf{X} = \mathbf{x}) = \frac{\exp\left(\beta_{k0} + \beta_k^T x\right)}{1 + \sum_{k=1}^{K-1}\exp\left(\beta_{k0} + \beta_k^T x\right)} \tag{2.7}$$

$$\mathbb{P}(G = K \mid \mathbf{X} = \mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{K-1}\exp\left(\beta_{k0} + \beta_k^T x\right)} \tag{2.8}$$

We then calculate an approximation for the coefficient matrix β by maximising the likelihood $l(\beta) = \prod_{i=1}^{m}\mathbb{P}(G = g_i \mid \mathbf{X} = \mathbf{x})$ for a sample containing m labelled observations, via numerical methods.

Example of association rules analysis via supervised learning methods

Consider a random variable $\mathbf{X} \sim \mathcal{N}_2\left(\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}5 & 2\\2 & 3\end{pmatrix}\right)$ from which we draw 500 observations and let us consider the problem of estimating the distribution of $\mathbf{X}$ assuming we didn't already know its distribution.

To do so we begin by generating 500 values from a reference random variable $\boldsymbol{\xi} \sim \mathcal{N}_2\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right)$ and combining them with the 500 observations from the original distribution, we produce a data matrix with row labels Y whose values are given by the indicator function $Y := 1\{\text{instance drawn from original distributio}$ Using the above method, we then produce an estimate (plotted in orange) for the original distribution (plotted in red - against the reference distribution - plotted in yellow).

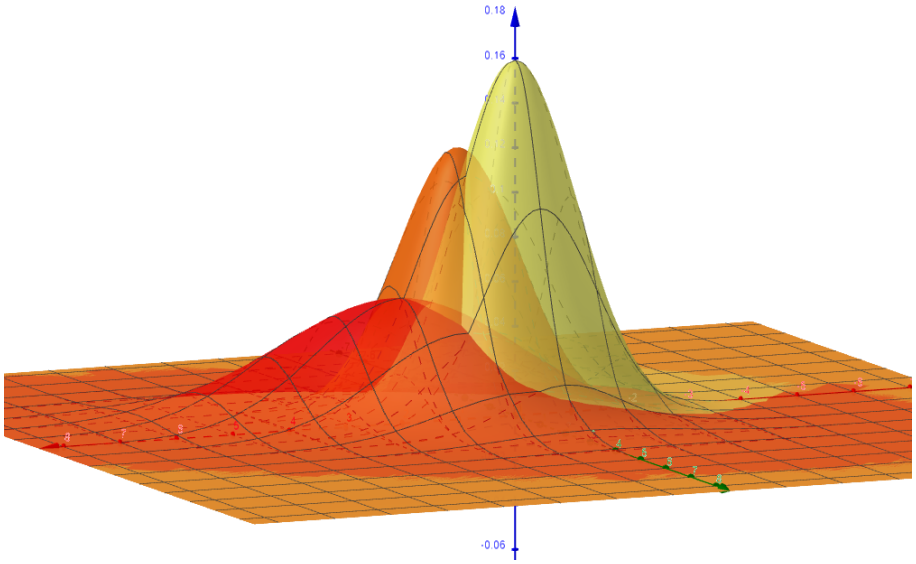

Figure 2.2: Graph illustrating association rule analysis via logistic regression on the dataset drawn from $\mathbf{X}$, with probability density functions $g(\mathbf{X})$ and $g(\boldsymbol{\xi})$ plotted in red and yellow respectively and the estimated probability density function $\hat{g}(\mathbf{X})$ plotted in orange.

$$g(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}\left(\mathbf{x} - \begin{pmatrix}2\\0\end{pmatrix}\right)^{\mathrm{T}} \begin{pmatrix}5 & 2\\2 & 3\end{pmatrix}^{-1} \left(\mathbf{x} - \begin{pmatrix}2\\0\end{pmatrix}\right)\right)}{2\pi\sqrt{\det\begin{pmatrix}5 & 2\\2 & 3\end{pmatrix}}}$$

$$g_0(\mathbf{x}) = \frac{1}{2\pi}\exp\left(-\frac{|\mathbf{x}|}{2}\right)$$

$$\hat{g}(\mathbf{x}) = g_0(\mathbf{x})\exp\left(\begin{pmatrix}-0.78190\\0.87032\\-0.40289\end{pmatrix} \cdot \begin{pmatrix}1\\x_1\\x_2\end{pmatrix}\right)$$

From this distribution, we can then calculate regions that satisfy any user-specified support and confidence requirements. For example, if we specify minimum probability density (support) values of $0.045, 0.025, 0.01$ and 0.0035, we obtain the following regions from our estimated PDF.

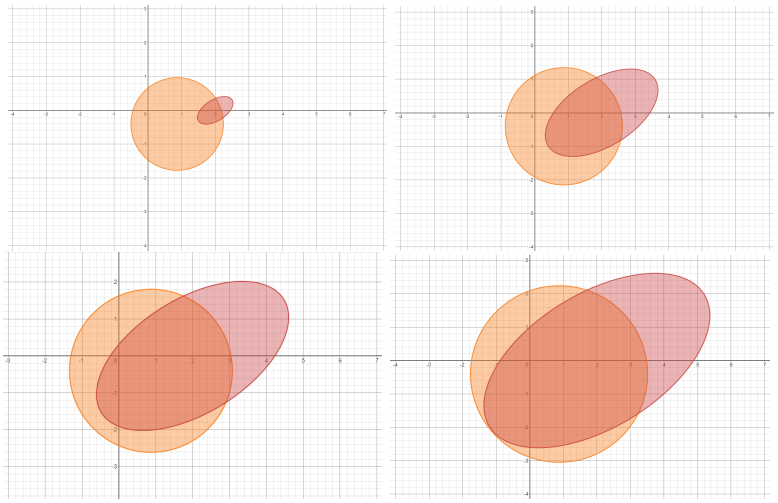

Figure 2.3: Graphs illustrating estimated sufficient support regions of $\mathbb{R}^2$ for $\mathbf{X}$ for support values $0.045, 0.025, 0.01$ and 0.0035 respectively

2.1.4 Applications of association rules analysis

So far the only non-abstract example of association rules analysis we have considered was related to market segmentation and although association rules analysis was born out of its application to market segmentation through the study of customer transaction databases, the range of applications has since broadened with, as implied earlier, major applications to medical diagnosis now having been developed, and clearly the method in the above example, which enables one to directly estimate a probability density function, has a very broad range of potential uses, due to its generality.

2.2 Cluster Analysis

[6]The second field of unsupervised machine learning we will explore is cluster analysis. At its core, clustering is the process of segmenting instances from a population into sub-populations based on some shared or similar characteristics of instances in each sup-population. This process is instinctive for human beings; every aspect of our daily lives depends on our ability to do so. All language is predicated on subconscious clustering; nouns for example, refer to clusters of instances that are in some sense similar, despite not being identical. For example, despite there being significant variation within each of the worldwide populations of dogs and cats, any human, having been given a mixed sample of dogs and cats, could successfully identify which are dogs and which are cats and thus successfully segment the sample into two clusters. The understanding and instinctive use of other elements of grammar,

[6]This introduction to cluster analysis is heavily inspired by the introduction to cluster analysis in [4]

universal to all major languages, is also an act of clustering; to observe a dog running away and subsequently make the statement "the dog ran that way!" requires a subconscious understanding of which features (and in what measure) distinguish running from walking, flying, jumping, hopping etc. and thus more abstractly requires an understanding (universal to all people) of how to segment the population of instances of locomotive actions (individual acts of walking, running etc.) into clusters (where instances in each cluster correspond to a unique form of locomotive action) [4].

Cluster analysis is the field of academic study that seeks to add mathematical rigour to this process when conducted empirically based on collected data, and the first step in adding rigour to this process is to define specifically what a cluster is. However, because the goals of cluster analysis are not always exactly the same, this proves to be surprisingly tricky. For example, consider the following two clustering problems:

1. A biologist records genetic data from a very large sample of n individual plants and wishes to determine how many unique plant species there are in this sample and to which species each plant in the sample belongs.

2. A bank wishes to value n corporations and has decided it would be most efficient to allocate similar corporations to each of their k analysts. It thus decides to segment the n corporations into k clusters - one for each analyst (possibly under the constraint that each analyst can only value a maximum of U corporations and so no cluster can be assigned more than U instances (corporations)).

In the first problem the sample is either assumed or known to be drawn from a mixture distribution and the clustering process corresponds to estimating the number of component distributions and which instances (in this case plants) belong to which component distribution (in this case species) - but importantly it's fundamentally an inference problem. However, in the latter problem the notion of clustering refers not to anything related to inference about the distribution from which the sample was drawn, but merely to the segmentation of the samples into a pre-specified number of clusters; the sample is the entire subject of interest and the population is irrelevant.
Generalising this idea we can formalise the distinction between two types of clustering problem:

Type I A sample $X_{n\times p}$ is drawn from mixture distribution $F(\mathbf{x})$ and using this sample we wish to determine the number k^* of component distributions $F_j(\mathbf{x})$ of $F(\mathbf{x})$ and which component distribution each instance $\mathbf{x_i}$ belongs to.

Type II A sample $X_{n\times p}$ is to be segmented into k clusters, such that the total dissimilarity between same-cluster instances is minimised as much as possible.

Our main focus in this report will be on Type I problems, because it is much easier with these problems to assess the performance of a given clustering method; we can effectively treat them like adapted classification problems, where the goals are to classify instances according to the component distributions they were drawn from. For Type II problems by contrast it is much harder to evaluate the performance of any given method. What distinguishes a good Type II clustering method from a

bad one? This ambiguity is the reason why, only Type I problems will be included in this report (which is aimed at critically evaluating the performance of such methods). Nonetheless, given that both types are fundamentally quite similar, it seems intuitive that any method that performs demonstrably well on Type I problems would likely also perform well on Type II problems.

However, before we proceed onto the study of methods, it is necessary to issue a warning with regard to Type I problems; they're not always solvable. For example if two multivariate normal distributions, with $\boldsymbol{\mu_1} \approx \boldsymbol{\mu_2}$ and $\det(\Sigma_1), \det(\Sigma_2) \gg 0$, are used as component distributions to form a mixture distribution, this distribution will be unimodal and due to the large overlap in high probability density regions between the two component distributions, it would be impossible for any clustering algorithm to classify instances with accuracy much greater than 0.5. It is for this reason, that only mixture distributions with reasonably distinct component distributions will be considered in this report, and when assessing the performance of methods applied to incomplete datasets, the error rate for a corresponding complete dataset will be used as the baseline rather than zero error rate (which may be impossible even in theory, let alone in practise).

2.2.1 Dissimilarity measures and proximity matrices

The notion of dissimilarity is fundamental to cluster analysis and unsurprisingly the choice of dissimilarity measure greatly affects the outcome of/solution to any clustering problem. In our exploration of options for choices of dissimilarity measures, let us begin by considering a trivial case. That is let us consider a sample of n students and record for each student which subjects they study and what exam score they obtain in each subject.

Quantitative variables

There are multiple different distance measures that could be employed for measuring dissimilarity between vectors in $\mathbb{R}^n$ or subsets of $\mathbb{R}^n$, such as $\mathbb{Z}^n$ or $\mathbb{N}^n$ - for example, Chebyschev distance ($\|\boldsymbol{\Delta}\mathbf{x}\|_\infty := \max_i \Delta x_i$), Euclidean distance $\left(\|\boldsymbol{\Delta}\mathbf{x}\|_2 := \sqrt{\sum_i {\Delta x_i}^2}\right)$ (which is often the default given that it describes distance in real 3D space) and Manhattan distance ($\|\boldsymbol{\Delta}\mathbf{x}\|_1 := \sum_i \Delta x_i$). However, often the most important consideration when choosing a dissimilarity measure is standardisation and scaling. For example, consider the student dataset described above; the variance of scores may vary greatly from subject to subject. Suppose for example that the calculated sample variance of scores in history was 25 while the calculated sample variance of scores in physics was 225. Then, a 10 point difference in history scores between student i_1 and student i_2 would arguably be indicative of a greater difference in history knowledge than a 15 point difference in physics scores between the same two students would be for their difference in physics knowledge. It depends on whether the lesser variance in scores in history is judged to be due to a lesser variance in history knowledge among students taking history or whether it is judged to be due to the history exam failing to differentiate students with differing levels of history knowledge as effectively as the physics exam differentiates students with differing levels of physics knowledge. Of course if the latter is judged to be the case, it can be corrected for by standardising all history scores by scaling down their difference from the mean by

the standard deviation of history scores. More generally if it is decided that scores must be adjusted in accordance with not only variance but also their covariances relative to each other, then the scores can be appropriately scaled by left multiplying the scores vectors (containing scores for all subjects) by the inverse of the square root of the covariance matrix, so that for example the Euclidean distance becomes $\|\mathbf{\Delta x}\| = \sqrt{\mathbf{\Delta x}^T \mathbf{S}^{-1} \mathbf{\Delta x}}$, otherwise known as the Mahalanobis distance. However, all such judgements are context and application dependent and thus there is no general rule for selecting an appropriate dissimilarity measure for quantitative variables.

Categorical variables

Categorical variables can be harder to apply dissimilarity measures to, since they are not inherently quantitative and yet we endeavour to assign quantities to their relative dissimilarities from one another. In most cases, this issue is superficially "resolved" by applying a binary dissimilarity measure mapping pairs of values of a categorical variable to 1 if the values differ and 0 if not. However, this sometimes fails to capture an accurate sense of the true dissimilarity. For example, returning to the student dataset example, consider three students - one studying maths, physics and chemistry, another studying biology, geography and economics and a third studying English, French and Spanish. By the binary dissimilarity measure described above, these three students' subject selections are equally dissimilar from each other and yet we clearly intuit this to be false, since we recognise that, although no two of the three students have any subjects in common, the second student's subjects are more similar to the first's than the third's are. To capture this dissimilarity, a network dissimilarity model can be constructed, where the dissimilarity between two subjects (nodes) is the length of the minimum length path between them. So under this model, $d(\text{maths}, \text{economics}) = 1 < d(\text{maths}, \text{French}) = 3$

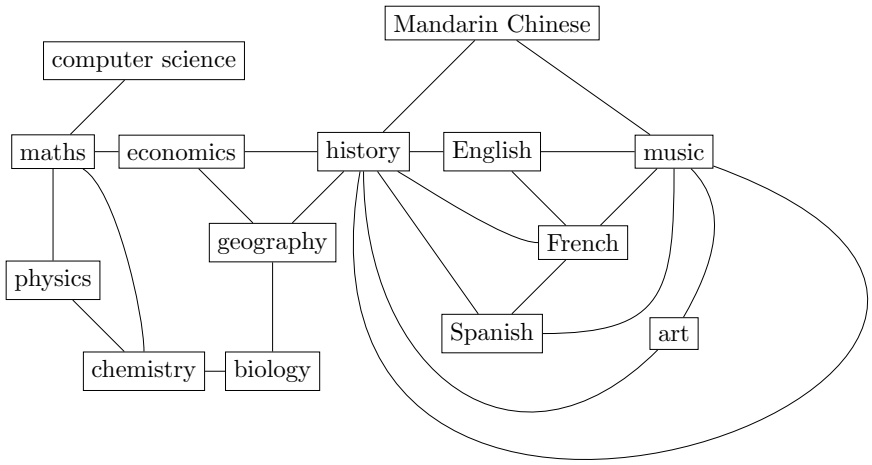

Figure 2.4: Example of a network dissimilarity model for dissimilarities between subjects

which arguably better captures the dissimilarity between the two subjects than the binary dissimilarity measure, $d(\text{maths}, \text{economics}) = 1 = d(\text{maths}, \text{French})$. Note also that we preserve symmetry - $\forall i, j \;\; d(i, j) = d(j, i)$. However, it should be noted that, when generalising the use of network dissimilarity models (or any dissimilarity measure for categorical variables) to vectors of categorical

variables, careful choice of norm is required to preserve the invariance of dissimilarity under permutation of the categorical variables. For example, if we were to treat each of the first two students' subject selections as a vector and applied the above network dissimilarity metric via the Euclidean norm we would find that, $d\left(\begin{pmatrix}\text{maths}\\\text{physics}\\\text{chemistry}\end{pmatrix},\begin{pmatrix}\text{biology}\\\text{geography}\\\text{economics}\end{pmatrix}\right) = \left\|\begin{pmatrix}2\\3\\2\end{pmatrix}\right\|_2 = \sqrt{17} \approx 4.12$ and yet, $d\left(\begin{pmatrix}\text{maths}\\\text{physics}\\\text{chemistry}\end{pmatrix},\begin{pmatrix}\text{geography}\\\text{economics}\\\text{biology}\end{pmatrix}\right) = \left\|\begin{pmatrix}2\\2\\1\end{pmatrix}\right\|_2 = 3$. This clearly renders this an unsuitable dissimilarity measure since the permutation of subjects is meaningless and so shouldn't affect the value of the output. Perhaps a more suitable dissimilarity measure would be one in which the mean of the dissimilarities of every possible pair of subjects, that can be formed by drawing a subject from each subject set, is calculated.

Of course, the major drawback of the network dissimilarity measure (in comparison with the binary dissimilarity measure) is the subjectivity resulting from the construction of the graph; namely any two people may construct different models producing different dissimilarity measures. Thus there are advantages and disadvantages to both approaches and, as with quantitative variables, contextual judgement is required to choose an appropriate dissimilarity measure.

Once we have chosen an appropriate dissimilarity measure, and implemented any necessary scaling, we can use a technique called principal component analysis to visualise the data.

2.2.2 Principal component analysis (PCA) and human visual detection of clusters

Principal component analysis is a field of unsupervised learning aimed at reducing the dimension (say from dimension p to dimension q) of a dataset while losing as little information as possible from said dataset. That corresponds to constructing a set of q ordered linear combinations of the dataset's p variables, such that variables with higher variances are given heavier (greater in absolute value) weightings and variables which are highly correlated with other variables, that already have a heavy weighting assigned to them, are given a lighter weighting.

This set of linear combinations can be represented by a matrix $V_{p\times q}$ (under the constraint - to ensure uniqueness of the matrix - that each column be a unit vector), the weightings of the columns for each data point reconstruction can be represented by the q-entry vector $\boldsymbol{\lambda}$ and the mean of all the data points can be represented by the p-entry vector $\boldsymbol{\mu}$. The problem of calculating the principal components, for an $n \times p$ dataset, is then reduced to solving the optimisation problem, [3]

$$\underset{\boldsymbol{\mu},V_{p\times q},\boldsymbol{\lambda}}{\arg\min} \sum_{i=1}^{n} \|\mathbf{x_i} - \boldsymbol{\mu} - V_{p\times q}\boldsymbol{\lambda}\|^2 \tag{2.9}$$

We can partially optimise $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ analytically: [3]

$$\hat{\boldsymbol{\mu}} = \overline{\mathbf{x}} \tag{2.10}$$

$$\hat{\boldsymbol{\lambda}} = V_{p\times q}\left(\mathbf{x_i} - \overline{\mathbf{x}}\right) \tag{2.11}$$

However, $V_{p\times q}$ must be optimised numerically.
Principal component analysis is particularly useful for clustering, because it can often enable visual identification of clusters in high-dimensional datasets, by plotting the first two principal components against each other, you can often visualise a significant proportion of the variance in a 2-dimensional plane and thus identify patterns and trends, such as clusters.

2.2.3 Optimisation clustering algorithms

Nonetheless, although PCA can be very useful for identifying trends, in very large datasets sometimes the first two principal components don't explain enough of the variance to enable visual identification of clusters and thus more rigorous methods are required. One such class of methods are optimisation clustering algorithms, which - as the name suggests - approach the problem of clustering by iteratively optimising a given expression.

k-Means

Arguably the most well-known optimisation clustering algorithm is the k-means clustering algorithm, which seeks to solve the optimisation problem, [3]

$$C^* = \left(C^*, \{m_\kappa^*\}_1^k\right)_1 = \underset{C,\{m_\kappa\}_1^k}{\arg\min} \sum_{\kappa=1}^{k} N_\kappa \sum_{C(i)=\kappa} \|x_i - m_\kappa\|^2 \tag{2.12}$$

where $\{m_\kappa^*\}_1^k$ denotes the optimal choice of cluster means and $(\cdot)_1$ denotes the extraction of the first element in a set.
To optimise this expression, we first note that, for any set of observations $S = \{x_1, ..., x_n\}$,

$$\bar{x} = \underset{m}{\arg\min} \sum_{i=1}^{n} \|x_i - m\|^2 \tag{2.13}$$

And thus deduce that,

$$C^* = \underset{C}{\arg\min} \sum_{\kappa=1}^{k} N_\kappa \sum_{C(i)=\kappa} \|x_i - \bar{x}_\kappa\|^2 \tag{2.14}$$

Of course these latter two formulae only partially optimise (2.12), but alternating implementation of them, enables us to converge towards a clustering assignment that increasingly (totally) optimises (2.12). This is the logical basis of the k-means clustering algorithm.

k-Means algorithm

1. A random set $\{m_1, ..., m_k\}$ of cluster means are selected from the space of values in $\mathcal{X}$.

2. Given the current set $\{m_1, ..., m_k\}$ of cluster means, (2.12) is minimised by assigning each observation to the closest current cluster mean. That is, $C(i) = \arg\min_{\kappa\in\{1,...,k\}} \|x_i - m_\kappa\|^2$.

3. Given the current cluster assignment $C : \{1, ..., n\} \to \{1, ..., k\}$, (2.12) is minimised with respect to the cluster means $\{m_1, ..., m_k\}$ by taking $\{m_1, ..., m_k\} = \{\bar{x}_1, ..., \bar{x}_k\}$.

4. Steps 2. and 3. are iterated until the cluster assignment does not change.

5. Return the final cluster assignment.

Note that the initial selection of cluster means could potentially effect the returned cluster assignment because the above algorithm only minimises 2.12 locally not necessarily globally; thus, in practice, the above algorithm is usually run several times with different initialisations and if multiple distinct assignments are returned, then whichever assignment results in the least total cluster variance (i.e. minimises 2.12) is estimated to be the global minimum.

k-Means example application

Applying the k-means algorithm to the simulated film review dataset, we then hope to identify three distinct clusters (due to there being three component distributions), where hopefully the cluster assignments of as large a proportion of the instances (film viewers) as possible correspond to the component distributions from which those instances were drawn.[7] We illustrate (using principal component analysis as described previously) the results produced by the algorithm in Figure 2.5, where the colour coding represents the cluster assignment calculated by the k-means algorithm. Comparing the calculated cluster assignments with the (known) true assignments, we can then calculate that the proportion of instances assigned to the wrong cluster is approximately 0.015, meaning that the k means algorithm has performed very well, although this assumes that we knew there were three component distributions. Aside from measuring the proportion of instances incorrectly assigned, we can also see signs of the efficacy of k-means clustering from the clustering graphs themselves. We see in the graph of the first two principal components against each other that the instances form three clearly separated clusters and that the cluster assignments (represented through colour coding) resulting from the application of the k-means algorithm closely follow this clustering pattern. We even observe that the third principal component provides significant evidence of clustering and that the assignments produced by the k-means algorithm are again consistent with this clustering pattern.

k-Medoids

The k-means algorithm is not the only widely used optimisation clustering algorithm; another such algorithm is the k-medoids algorithm. In k-medoids clustering, we seek to solve the similar optimisation problem, [3]

$$C^* = \min_{C, \{m_\kappa\}_1^k} \sum_{\kappa=1}^{k} N_\kappa \sum_{C(i)=\kappa} \|x_i - m_\kappa\|^2, \text{ for some } \{m_\kappa\}_1^k \subset \{x_1, ..., x_n\} \tag{2.15}$$

[7] See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/k-means%20clustering.R` for the code used to generate this dataset.

Thus k-medoids works very similarly to k-means, with the only difference being that k-medoids only selects cluster centroids from the space of instances, not the means of instances and hence the k-medoids algorithm's cluster centroid optimisation step works by exhaustion (computing the errors for every possible selection of centroid for the set of instances in each cluster).

k-Medoids algorithm

1. A random set $\{m_1, ..., m_k\}$ of cluster medoids are selected from the set S of observed values.
2. Given the current set $\{m_1, ..., m_k\}$ of cluster medoids, 2.15 is minimised by assigning each observation to the closest current cluster medoid. That is, $C(i) = \arg\min_{\kappa \in \{1,...,k\}} \|x_i - m_\kappa\|^2$.
3. Given the current cluster assignment $C : \{1, ..., n\} \to \{1, ..., k\}$, 2.12 is minimised with respect to the cluster medoids $\{m_1, ..., m_k\}$ by taking $\{m_1, ..., m_k\} = \{\arg\min_{m_1} \sum_{C(i)=1} \|x_i - m_1\|^2, ..., \arg\min_{m_k} \sum_{C(i)=k} \|x_i - m_k\|^2\}$.
4. Steps 2. and 3. are iterated until the cluster assignment does not change.
5. Return the final cluster assignment.

Because, for any given cluster mean/medoid m_j, computing $\arg\min_{m_1} \sum_{C(i)=1} \|x_i - m_1\|^2$ is far more computationally expensive than computing $\bar{x}_j$, the k-medoids algorithm is more computationally expensive to run than the k-means algorithm. But note also that for the k-medoids algorithm, all relevant dissimilarity information can be stored in a (symmetric) proximity matrix, where the $(i, j)^{\text{th}}$ entry of the matrix takes the value of the dissimilarity between the i^{th} and j^{th} instances. This is due to the fact that the k-medoids algorithm only requires dissimilarity calculations between instances. So there is one benefit to k-medoids over k-means, but for our purposes we will focus on the use of k-means.

Of course, k-means and k-medoids by no means form a comprehensive list of actively used clustering algorithms and there are a number of radically different approaches to clustering outside of optimisation clustering, such as hierarchical clustering and Bayesian mixture modelling. Moreover, the range of potential applications of clustering extends far beyond just market segmentation (such as film viewer clustering used as the example above) into medical applications, such as differentiating types of tissue in PET scans, and other biological applications such as genetic analysis of sub-populations of humans and other organisms, as well as applications in several other fields such as finance, with the emergence of methods such as pairs trading. However, for the purpose of this report it will not be necessary to explore alternative clustering methods and applications, and we progress in the next section onto the issue of how to apply ARA and CA methods to incomplete datasets.

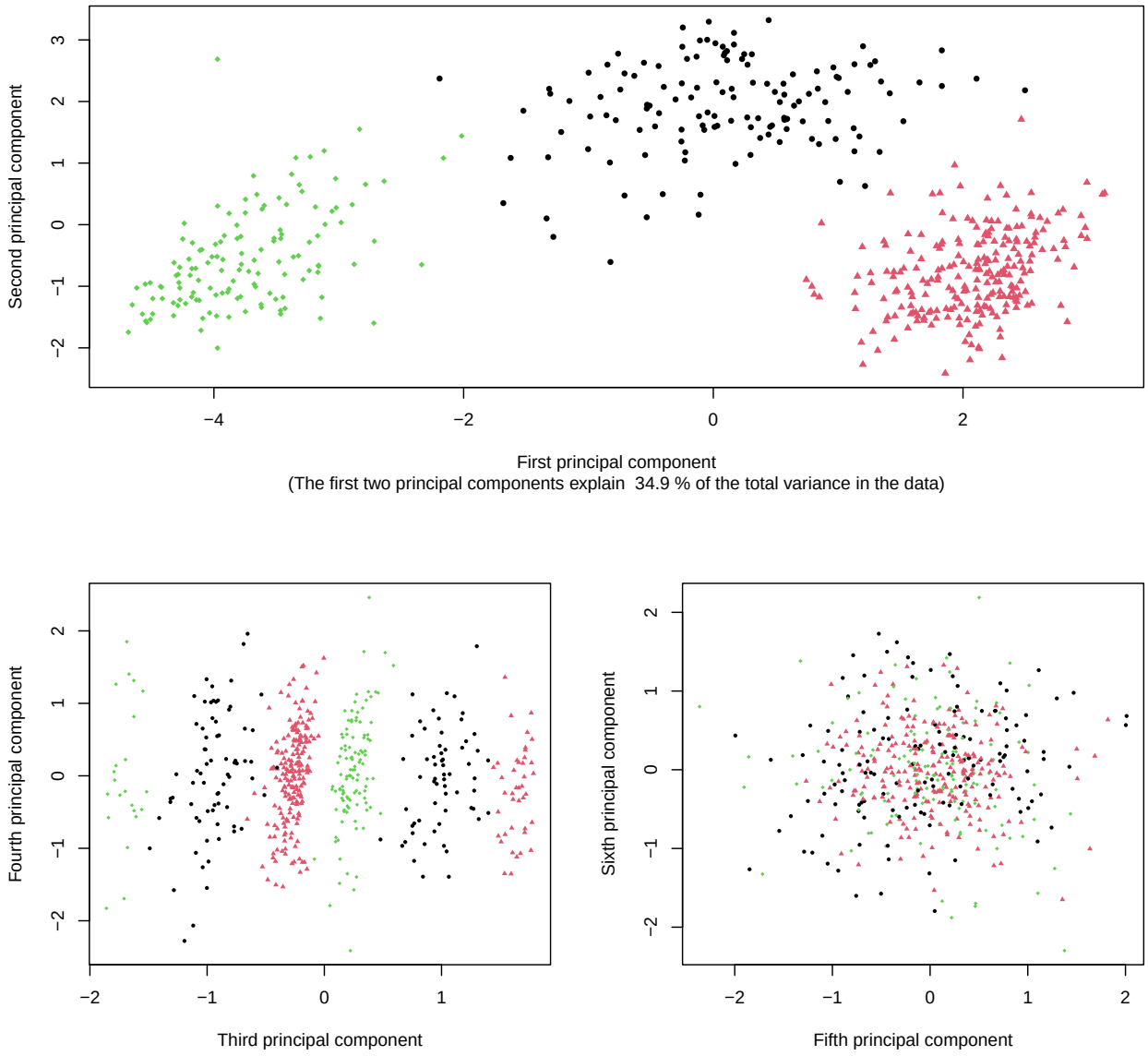

Figure 2.5: Graphs illustrating k-means clustering on the simulated film review dataset

CHAPTER 3

Non-Imputation and Single Imputation based Methods

When applying clustering methods to incomplete datasets, there are a number of quick-fix solutions that may spring to mind. Before proceeding to more complex methods, it is necessary to first explore these simplistic methods and gain a comprehensive understanding of their shortcomings so that we have some motivation for the study of alternative methods and also so that we can compare the performance of the more complex methods against the performance of the simplistic methods to justify that they do indeed perform significantly better.

In order to assess the effectiveness of these methods, we must construct an appropriate error rate function, so that we can quantify the quality of their performance. And since the purpose of this project has been to explore methods for imputation of missing values as prerequisites for association rules and cluster analysis, any error rate function should directly measure the error in these subsequent analyses rather than the error of the imputations themselves. As such for cluster analysis, a natural error rate function to use would be the proportion of instances which are wrongly assigned (i.e. assigned to a cluster to which they do not belong). Of course, each cluster is only defined by its member instances and so in writing the code implement this, one must be cautious to ensure that the permutation of clusters that minimises such error is the one used to calculate the error[1]. For association rules analysis, developing a good error rate function is somewhat more subjective, since there are in some sense two competing factors; we wish on one hand to reward (i.e. reduce error rate for) correctly discovered association rules (i.e. ones which were discovered by the same association rules mining algorithm on the original/unsparsified dataset) but we wish also to punish (i.e. increase error rate for) misdiscovered false association rules. The question then is how to weight these competing factors; should the error rate function penalise wrongful discovery less or more harshly than it rewards correct discovery? The answer to this question inevitably depends on how the results of the association rules

[1]See 'centroid_match' and 'check_against_true_assignment' functions in the 'k-means clustering' file for my coded solution to this problem

analysis are intended to be used and so no definitive answer can be given in general. As such, for our purposes we will in some sense compromise by weighting them equally and so we could measure the error rate by subtracting the misdiscovered rues from the correctly discovered rules and dividing the result by the total number of correct rules (i.e. ones deduced from the original unsparsified dataset). Except we require one further adjustment; any output that has more misdiscovered rules than correctly discovered ones is deemed useless under our function and so distinguishing between the error rates of these entirely useless outputs is pointless and so our error rate function may as well be capped at value one.

3.1 Complete case analysis

The first method for dealing with missing data is complete case analysis; that is simply remove instances for which data is missing (and thus conduct the analysis on only the complete instances/cases). Clearly for cluster analysis, this has the major flaw that those incomplete instances (instances for which some values are missing) cannot be assigned to a cluster and thus in the case of a Type II problem it is impossible to estimate which component distribution they were drawn from. However, let us merely consider the accuracy of the clustering method on the complete instances.

We see from Figure 3.1 that even for the remaining instances, this method performs worse than with the full dataset. This is because the other instances would have provided information on the position of the cluster centroids and thus would have indirectly enabled a more accurate allocation of complete instances to clusters. Hence, complete case analysis is a very poor and very wasteful approach to handling the issue of missing values; its only advantages are how easy it is to implement and how easy it is for someone from a non-mathematical background to understand.

3.2 Adaptation of Dissimilarity Measures

Given the demonstrable wastefulness of simply deleting incomplete rows, one may speculate whether it is possible to adapt the unsupervised learning methods studied in the previous chapter to make them applicable directly to a dataset without prior incomplete row removal. In the case of cluster analysis, that would require an NA aware dissimilarity measure (one that can be applied to pairs of incomplete instances). Naturally, one may consider applying one of the standard dissimilarity measures (Chebyschev, Euclidean and Manhattan) to the common (non-missing in both vectors) elements of each vector - in this case we'll consider the Euclidean dissimilarity measure, although every issue we're about to uncover applies equally to the other two measures also.

Besides the obvious problem that this dissimilarity measure is only well-defined for pairs of vectors with at least one common element, there is a more fundamental problem with this measure. To demonstrate, consider that for any instances $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ which have values for entries indexed by values in $\mathcal{I}, \mathcal{J}, \mathcal{K}$ respectively,

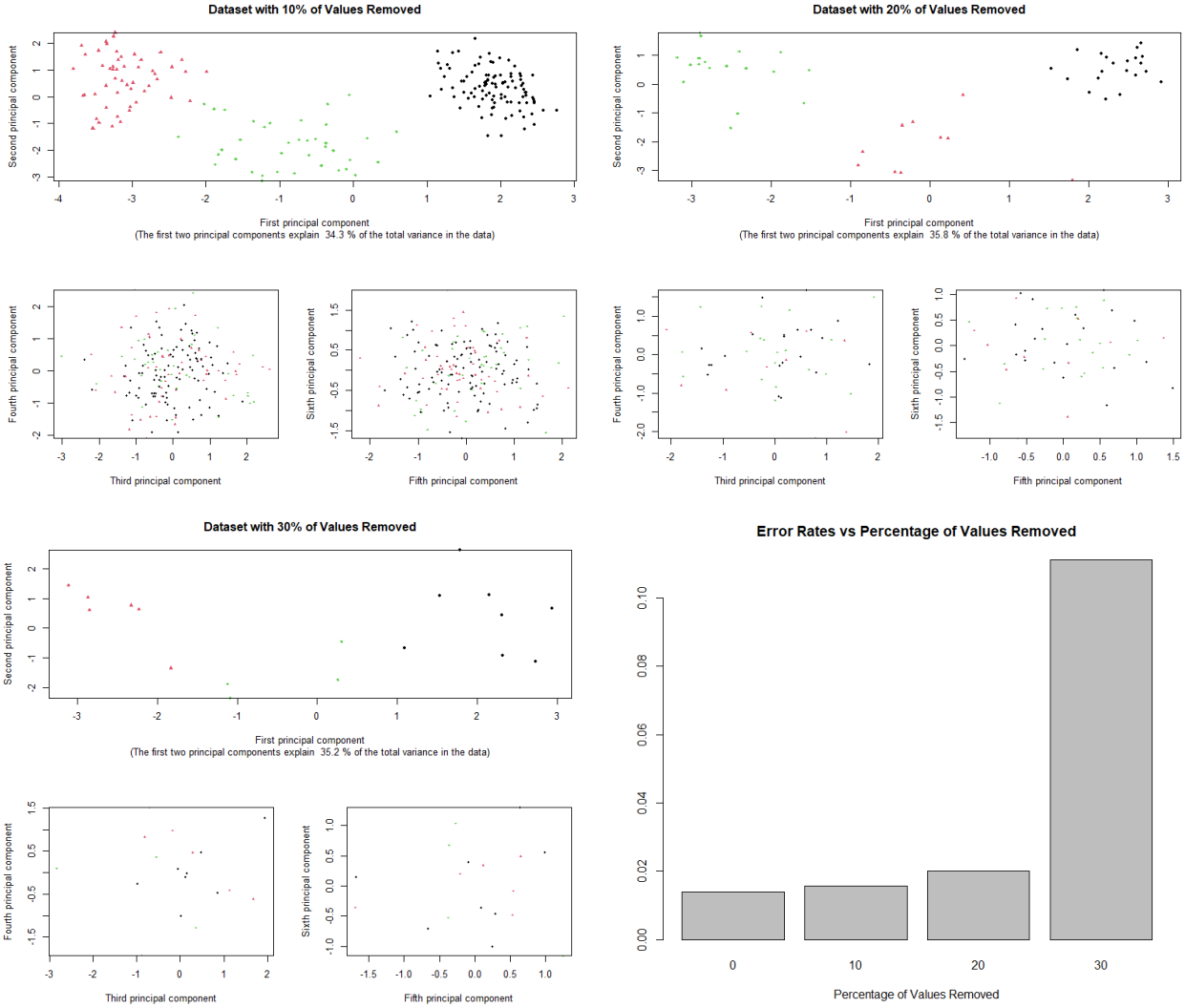

Figure 3.1: Graphs illustrating the effectiveness of complete case analysis on randomly sparsified versions of the simulated film review dataset prior to k-means clustering of the simulated film review dataset

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{n \in \mathcal{I} \cap \mathcal{J}} ((\mathbf{x}_i)_n - (\mathbf{x}_j)_n)^2}$$

$$d(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{\sum_{n \in \mathcal{I} \cap \mathcal{K}} ((\mathbf{x}_i)_n - (\mathbf{x}_k)_n)^2}$$

$$d(\mathbf{x}_j, \mathbf{x}_k) = \sqrt{\sum_{n \in \mathcal{J} \cap \mathcal{K}} ((\mathbf{x}_j)_n - (\mathbf{x}_k)_n)^2}$$

and hence, for the following 3 tri-element vectors,

$$\begin{pmatrix} 0 \\ 0 \\ \text{NA} \end{pmatrix}, \quad \begin{pmatrix} 1 \\ \text{NA} \\ 0 \end{pmatrix}, \quad \begin{pmatrix} \text{NA} \\ 2 \\ 0 \end{pmatrix}$$

we have that,

$$d\left(\begin{pmatrix}0\\0\\\text{NA}\end{pmatrix},\begin{pmatrix}1\\\text{NA}\\0\end{pmatrix}\right)=1,\quad d\left(\begin{pmatrix}0\\0\\\text{NA}\end{pmatrix},\begin{pmatrix}\text{NA}\\2\\0\end{pmatrix}\right)=2,\quad d\left(\begin{pmatrix}1\\\text{NA}\\0\end{pmatrix},\begin{pmatrix}\text{NA}\\2\\0\end{pmatrix}\right)=0$$

$$\implies d\left(\begin{pmatrix}0\\0\\\text{NA}\end{pmatrix},\begin{pmatrix}\text{NA}\\2\\0\end{pmatrix}\right)>d\left(\begin{pmatrix}0\\0\\\text{NA}\end{pmatrix},\begin{pmatrix}1\\\text{NA}\\0\end{pmatrix}\right)+d\left(\begin{pmatrix}1\\\text{NA}\\0\end{pmatrix},\begin{pmatrix}\text{NA}\\2\\0\end{pmatrix}\right)$$

which violates the triangle inequality, and hence our intuitive notion of dissimilarity for incomplete datasets is actually not a valid dissimilarity measure and so this approach is fundamentally flawed.

3.3 Average and Novel Category based Imputation of missing values

Given the flaws of removing incomplete instances or attempting to adapt methods to avoid having to impute missing values, it is clear that there is a strong case for imputation to ensure that both; none of the non-missing values are wasted, and that we are able to construct a valid and meaningful dissimilarity measure. The most simple approach to doing this is to impute missing values for numerical variables with some form of average (of the non-missing column values) - typically the (arithmetic) mean or median - and to impute missing values for categorical variables with the modal category or with a novel category (i.e. treat NA as a categorical value).

3.4 Random Sampling and Posterior Predictive Sampling for Imputation of missing values

The issue with imputing values using notions of averages is that it distorts (typically artificially reduces) the column variance. An alternative approach which typically doesn't have such a strong distortive effect is random sampling. So for example, if a column of people's heights in inches drawn from,[2]

$$x_i \sim \frac{\mathcal{N}(64, 3.25) + \mathcal{N}(69, 3.5)}{2}$$

has values as follows (in vector form),

$$(63, 69, ..., 68, ..., 67, ..., 73, 69, 68, ..., 59, 64, 65, 61, 64, ..., 69, 67, 66, ..., 64, ..., 63, 79, ..., ..., 66, 72, 62, 67,$$

$$57, 66, ..., 69)$$

we can impute the missing values by randomly sampling from those values that are not missing. However, in the above example that would mean, for example imputing with value 64 with probability

[2]This distribution has been used because online sources indicate that the distributions of heights of men and women follow roughly (although sources don't agree exactly) this distribution, with these stated means and standard deviations.

$\frac{3}{25}$, imputing with value 66 with probability $\frac{3}{25}$ and yet imputing with value 65 with only probability $\frac{1}{25}$. Similarly there is a non-zero probability of imputing with value 79 and yet zero probability of imputing with any value from 74 to 78. Thus, this approach will clearly still have somewhat of a distortive effect, although that effect is greatly minimised for datasets with large numbers of values. An alternative approach, which will minimise this distortive effect even for datasets with smaller numbers of instances, is Bayesian posterior predictive sampling; that is we begin by proposing a parametric form of the sampling distribution and prior distributions for those parameters, based on our contextual understanding of the column variable and then apply Bayes' theorem to compute the posterior predictive distribution which we then sample from using Monte Carlo methods. So in this example, if we propose a normal sampling distribution with the following prior distributions[3],

$$\mu \sim \mathcal{N}\left(\mu_0 = 67, \frac{\sigma^2}{\tau_0} = \frac{\sigma^2}{2}\right), \quad \sigma^2 \sim \text{IG}\left(\alpha_0 = 50, \beta_0 = 200\right)$$

we can apply Bayes' Theorem and obtain our joint posterior distribution for μ and σ thusly:

$$\pi\left(\mu, \sigma^2 | \mathbf{x_{1:n}}\right) \propto f_{\mathbf{x_{1:n}}}\left(\mathbf{x_{1:n}} | \mu, \sigma^2\right) \pi_{\mu,\sigma^2}\left(\mu, \sigma^2\right) \overset{iid}{=} \prod_{i=1}^{n} \mathcal{N}\left(x_i | \mu, \sigma^2\right) \mathcal{N}\left(\mu | \mu_0, \frac{\sigma^2}{\tau_0}\right) \text{IG}\left(\sigma^2 | \alpha_0, \beta_0\right)$$

where $\mathbf{x_{1:n}}$ denotes the first n observations. Evaluation of the latter product then yields,

$$\pi\left(\mu, \sigma^2 | \mathbf{x_{1:n}}\right) \propto \mathcal{N}\left(\mu | \mu_n, \frac{\sigma^2}{\tau_n}\right) \text{IG}\left(\sigma^2 | \alpha_n, \beta_n\right)$$

$$\implies \mu | \mathbf{x_{1:n}}, \sigma^2 \sim \mathcal{N}\left(\mu_n, \frac{\sigma^2}{\tau_n}\right), \quad \sigma^2 | \mathbf{x_{1:n}} \sim \text{IG}\left(\alpha_n, \beta_n\right)$$

with,

$$\mu_n = \frac{n\overline{x_n} + \tau_0\mu_0}{n + \tau_0}, \quad \tau_n = \tau_0 + n, \quad \alpha_n = \alpha_0 + n, \quad \beta_n = \beta_0 + \frac{1}{2}ns_n^2 + \frac{1}{2}\frac{\tau_0 n\left(\mu_0 - \overline{x_n}\right)^2}{n + \tau_0}$$

So, (although we will leave substitution of values till the end) we have, in our example, that,

$$\mu_n = \frac{199}{3}, \quad \tau_n = 27, \quad \alpha_n = 75, \quad \beta_n \approx 461.4$$

We can then derive the posterior predictive distribution thusly,

$$f_{x_{n+1}}\left(x_{n+1} | \mathbf{x_{1:n}}\right) = \int_0^\infty \int_{\mathbb{R}} f_{x_{n+1}}\left(x_{n+1} | \mu, \sigma^2\right) \pi_\mu\left(\mu | \mathbf{x_{1:n}}, \sigma^2\right) \pi_{\sigma^2}\left(\sigma^2 | \mathbf{x_{1:n}}\right) \, d\mu \, d\sigma^2$$

$$\implies f_{x_{n+1}}\left(x_{n+1} | \mathbf{x_{1:n}}\right) = \int_0^\infty \text{IG}\left(\sigma^2 | \alpha_n, \beta_n\right) \int_{\mathbb{R}} \mathcal{N}\left(x_i | \mu, \sigma^2\right) \mathcal{N}\left(\mu | \mu_n, \frac{\sigma^2}{\tau_n}\right) \, d\mu \, d\sigma^2$$

[3] $\text{IG}(\cdot)$ denotes the inverse gamma distribution.

which evaluates to,

$$x_{n+1}|\mathbf{x_{1:n}} \sim \mathrm{T}\left(\mu_n, \frac{\beta_n}{\alpha_n}\left(1+\frac{1}{\tau_n}\right), 2\alpha_n\right)$$

$$\implies x_{n+1}|\mathbf{x_{1:n}} \dot{\sim} \mathrm{T}\left(66.3, 6.40, 150\right)$$

We see from Figure 3.2 that the posterior predictive distribution does roughly approximate the

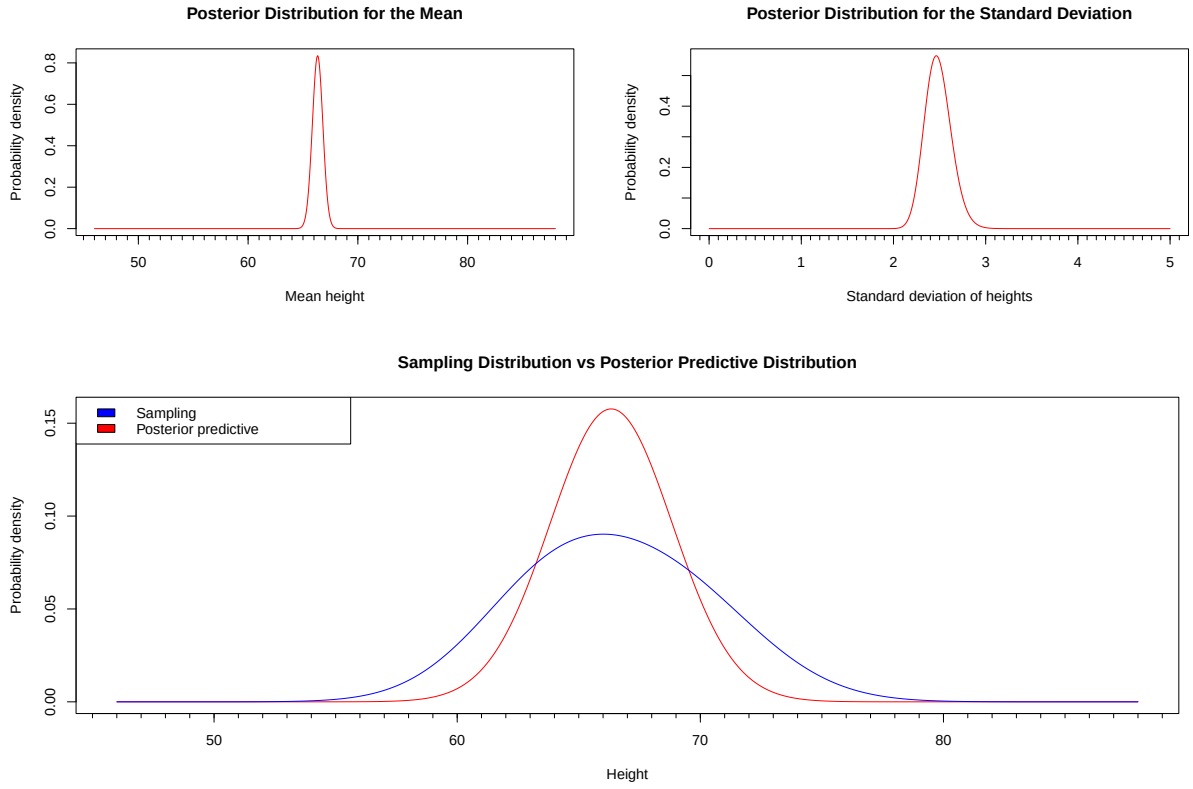

Figure 3.2: Graphs illustrating the posterior parametric distributions and the posterior predictive distribution against the sampling distribution to visualise the accuracy of its use for simulating values to impute those values which are missing

sampling distribution although has a lesser variance.

Using the posterior predictive distribution, we impute the missing values thusly:

$$(63, 69, 67, 68, 60, 67, 68, 73, 69, 68, 67, 59, 64, 65, 61, 64, 64, 69, 67, 66, 60, 64, 65, 63, 79, 66, 62, 66, 72, 62,$$

$$67, 57, 66, 71, 69)$$

Of course, this imputation is based on the assumption that those values that are missing are missing independently of their true values, and thus their true values were drawn from the same distribution as the ones not missing. We will explore later the issue of data missing through other mechanisms.

But first, let us explore the application of random sampling and posterior predictive sampling based imputation to the randomly sparsified versions of simulated film review dataset as prerequisites

for association rules analysis and cluster analysis. To apply the latter we must assume a distribution for the film review dataset and select an appropriate prior distribution. We must do this without using our knowledge of the true distribution that the data was simulated from, since that would defeat the purpose of its use as an imputation method and render it useless for any real world application. Thus, the distribution we select must be a distribution that someone may likely select without having seen any data. An obvious choice of such distribution therefore is the joint distribution formed by IID normal distributions for each film review and a Bernoulli distribution for the premium subscription Boolean variable - again independent of the distributions for each film review. The selection of normal and Bernoulli distributions hardly requires justification since they are often the default modelling choices for the distributions of continuous and binary random variables respectively. However, the independence assumptions are clearly less reasonable, since one would clearly expect films of a similar genre for example to have correlated reviews and for the premium subscription random variable to be positively correlated with the average of all the film reviews. Nonetheless, we justify the independence assumption on the basis that it massively reduces the complexity of our model; refusing to impose any independence assumptions would require us to introduce another $11C2 = 55$ parameters to our model (one for the correlation between any pair formed from each of the 11 variables).

Having assumed a normal distribution for each film review, it is then reasonable to assume conjugate prior distributions for the mean and variance of each film review - normal and inverse gamma respectively. Manual adjustment of the hyper-parameters using graphs of the prior distributions then helps us to obtain a reasonable selection; in this case we arrive at $\mu_{i0} = 3, \tau_{i0} = 2, \alpha_0 = 10, \beta_0 = 10$. We then use the result for the posterior predictive distribution derived in the above heights example:

$$x_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right), \quad \mu_i \sim \mathcal{N}\left(\mu_{i_0} = 3, \frac{\sigma_i^2}{\tau_0} = \frac{\sigma_i^2}{2}\right), \quad \sigma_i^2 \sim \text{IG}\left(\alpha_0 = 10, \beta_0 = 10\right) \quad \forall i \in \{1, ..., 11\}$$

$$\implies x_{in+1} | \mathbf{x_{i1:n}} \sim \text{T}\left(\mu_n, \frac{\beta_n}{\alpha_n}\left(1 + \frac{1}{\tau_n}\right), 2\alpha_n\right)$$

For the Bernoulli distribution it is again reasonable to assume the conjugate prior distribution for the success probability parameter - the beta distribution. We will not derive here the posterior predictive distribution for the premium subscription Boolean variable resulting from this choice of prior, since this derivation follows a very similar argument to the one shown in the heights example. However, such a derivation will yield the following result:

$$x_{11} \sim \text{Bernoulli}\left(p\right), \quad p \sim \text{Beta}\left(\gamma_0, \delta_0\right)$$

$$\implies x_{11n+1} | \mathbf{x_{111:n}} \sim \text{Bernoulli}\left(\frac{\gamma_n}{\gamma_n + \delta_n}\right)$$

with $\gamma_n := \gamma_0 + \sum_{i=1}^{n} x_{11i} - 1$ and $\delta_n := \delta_0 + n - \sum_{i=1}^{n} x_{11i} - 1$. We again choose the hyperparameters by manually adjusting them to match our intuition (or rather what we expect would be the intuition of a person who hadn't seen the data or distribution from which the data was simulated) and in this case arrive at values $\gamma_0 = 12$, $\delta_0 = 8$.

Figure 3.3 illustrates the performance of random sampling and posterior predictive sampling (us-

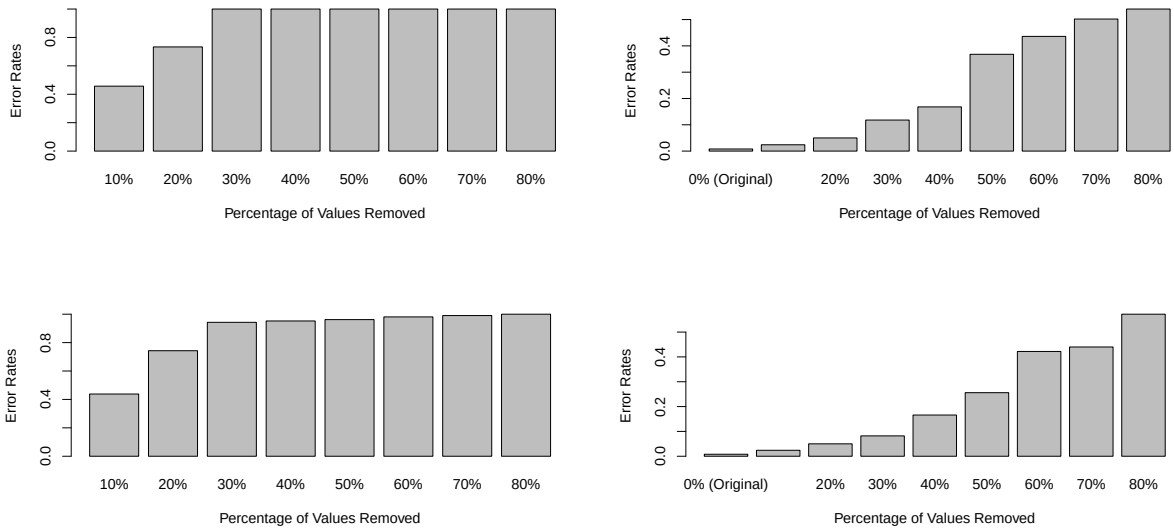

Figure 3.3: Graphs showing the association rules analysis error rates (left) and cluster analysis error rates (right) of random sampling from observed values (top) and random posterior predictive sampling (bottom) on the simulated film review dataset with MCAR values

ing the prior distributions stated above) for prerequisite imputation before ARA and clustering on the simulated film review dataset, and as can be seen, posterior predictive sampling does not perform significantly better than random sampling.[4] In the case of association rules analysis, this poor performance is largely due to the association rules that are implied by the chosen prior distribution translating over to and influencing the posterior distribution and thus misdiscovering several false association rules; these rules are punished by the error rate metric as harshly as the discovery of true rules are rewarded. This apparently minimal benefit associated with using posterior predictive sampling as opposed random sampling is therefore arguably more a flaw of our choice of prior distribution as opposed to a flaw of the use of posterior predictive sampling as a method. Looking at the right side graphs illustrating the performance of each method for prerequisite imputation before cluster analysis, we see that the difference in performance between random sampling and posterior predictive sampling is marginally greater than for association rules analysis - especially for larger proportions of data missing - although overall the difference is still not substantial. The increase in difference in performance between the two sampling based imputation methods for larger proportions of data missing can be explained by the increase in distortive effects described above when randomly sampling from the (fewer) observed values; as previously explained, this type of distortion does not occur with posterior predictive sampling.

Nonetheless, overall the increase in performance when switching from random sampling to posterior predictive sampling is disappointingly small and suggests that column-wise sampling based imputation

[4]See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Posterior%20predictive%20sampling.R` for the code used to generate to impute the simulated film review dataset using posterior predictive sampling.

methods, in which dependencies between variables are ignored, are greatly limited in their effectiveness. Hence, in the next few sections, we progress to exploring regression and classification based imputation methods.

3.5 Linear Regression

3.5.1 Overview of regression analysis

Regression analysis is the analysis of supervised learning methods to predict the value of a continuous random variable Y based on the values of certain predictor variables $\mathbf{X}$. Regression analysis is very useful for missing data imputation purposes because as previosuly eluded to it enables the imputer to take advantage of dependencies between variables in a dataset when imputing missing values. So for example, if a dataset's i^{th} column contains missing values, but all the other columns are complete then a regression model for the i^{th} variable using all other variables as predictors could be used to effectively impute the missing values.

Linear regression

Linear regression is a form of regression analysis which seeks to construct a linear model $Y = \mathbf{X}^T\boldsymbol{\beta} + \epsilon$ for the prediction of Y. The derivation of formulae for the distributions of the estimates for $\boldsymbol{\beta}$ and ϵ are premised on the assumption that $\epsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma)$, and so for linear regression to be useful, there has to be evidence that this assumption is correct.

3.5.2 Linear regression based imputation

We now seek to apply linear regression to the simulated film review dataset for the purpose of imputation, to enable subsequent association rules and cluster analysis. Clearly we have in this case missing values across all columns meaning that whichever variable we choose to impute first will not be regressable on the remaining variables since those remaining variables will still contain missing values. So to implement linear regression (or indeed any form of regression), we must first use a sampling based imputation method to impute the missing values with placeholder values. Although we have now developed the machinery to implement posterior predictive sampling for this step, we aim in this section to assess the effectiveness of linear regression based imputation and so we need only use random sampling as long as that is maintained across all other regression based imputation methods, to ensure a fair comparison can be made. Another imputation step that is required to complete the film review dataset is the imputation of missing values for the premium subscription Boolean variable. For this we will use logistic regression as described in section 2.1.3, and again will maintain this as a control for all future regression methods we investigate, to ensure a fair comparison.[5]

[5] See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Posterior%20predictive%20sampling.R` for the code used to impute the simulated film review dataset using both logistic regression and linear regression.

As can be seen from Figure 3.4 this method of imputation produces a completed dataset that is far more useful (in the sense that is produces more accurate results) for association rules analysis and cluster analysis than the two previously investigated sampling based imputation methods.

3.6 k-NN Classification and Regression

An alternative regression-based approach is to impute values not with a global variable average but with an average of the corresponding variable values for its k nearest neighbours, where k is some user-specified value - for our purposes, we will use $k := \sqrt{10n - n_{\text{NA}}}$ (i.e. the square root of the number of observed values in the dataset). However, the k nearest neighbours would have to be calculated using some NA aware measure, which, as demonstrated above, does not necessarily obey the triangle inequality and hence will not necessarily conform to our intuitive axiomatic notions of how a dissimilarity measure should behave.

Nonetheless, according to Jadhav et al., k-NN imputation (for numerical values) still outperforms mean imputation, median imputation, predictive mean matching, Bayesian linear regression, non-Bayesian linear regression, and random sampling [5]. Thus, despite its apparently flawed underpinnings, it should not be discarded so quickly, since empirical evidence of good performance is arguably sufficient justification for use in its own right. However, although k-NN imputation can also be used for imputing missing values for categorical variables, we will not explore this option, because the empirical evidence for the effectiveness of k-NN was specifically for k-NN regression, not k-NN classification, which compounded with the mathematically problematic basis for this method renders it unattractive.

Nonetheless, applying k-NN regression in the same manner as done with linear regression[6] (i.e. with random sampling used to generate placeholder values and logistic regression used for the imputation of the Boolean variable) as a prerequisite for clustering and association rules analysis produces results seemingly at odds with Jadhav et al.'s conclusions; kNN regression appears to perform worse than linear regression. Of course Jadhav et al. measured the effectiveness of each imputation method based on the accuracy of imputations made, whereas our interest is specifically in which method will enable more accurate results in subsequent cluster analysis or association rules analysis. Thus, our conclusions are not necessarily at odds. However, it must be noted that while most professional publications assess the performance of methods based on analyses of multiple datasets, we have used only one (due to the constraints of having to present all results and all analyses in a greatly restricted number of pages) and therefore our conclusions are not completely reliable and so in particular the disappointing performance of k-NN regression on this dataset may be a mere coincidence, not representative of its performance in general.

[6] See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Logistic%20regression%20and%20kNN%20regression.R` for the code used to generate to impute the simulated film review dataset using both logistic regression and k-NN regression.

3.7 Predictive Mean Matching

The final regression-based imputation method we will explore is predictive mean matching which works by building a regression model for the variable to be imputed (using all other variables as predictor variables) and then using this model to find associated predicted values of all missing and observed values of the variable to be imputed. Subsequently, for each missing value, the set of the k observed values whose predicted values are closest to the predicted value of this missing value can be randomly sampled from, or their weighted mean calculated [6]. For our purposes, the regression model we will use for predictive mean matching will be linear regression and the k predicted values will be averaged (arithmetic mean).[7]

We again demonstrate the application of predictive mean matching to association rules and cluster analysis of the simulated film review dataset, by using it as described previously, in conjunction with column-wise random sampling and logistic regression and illustrate the error rates of each procedure in Figure 3.6.

Figure 3.6 again demonstrates the limitations of analysing methods using just one dataset, since it presents an example of the paradoxical result in which the increases in proportion of values missing from 50% to 60% and then 70% produce successive decreases in error rates; this can only be coincidence, because no imputation method would in general perform better using fewer observed values. We also observe interestingly that predictive mean matching performs (at least on our dataset) better than linear regression and k-NN regression for association rules purposes but worse for clustering purposes.

3.8 Discriminant Analysis

[8]So far in assessing potential methods for imputation of missing values, we have investigated three regression based methods, but have each time taken by default logistic regression to be the method for imputing any categorical variables. Of course, fixing the categorical variable imputation method, enables a fair comparison of the regression based imputation methods used in parallel, but alone it doesn't enable for a full exploration of imputation methods so we now proceed to explore an alternative method of classification that can be also be employed for categorical variable imputation - discriminant analysis. Discriminant analysis, is premised on the assumption that if, for all the instances with the same true value for the variable to be imputed, random vectors are formed by omitting their entry for this variable, then those random vectors are all multivariate normally distributed. Hence, if we are imputing the j^{th} variable (which must be a discrete variable), which can take values in $\mathcal{L} = \{l_1, ..., l_L\}$ and thus can take $\#\mathcal{L} = L$ possible values, we assume that the distribution of the instances with common value l for this variable - minus this variable - follows a normal PDF,

[7]See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Logistic%20regression%20and%20predictive%20mean%20matching.R` for the code used to generate to impute the simulated film review dataset using both logistic regression and linear regression.

[8]This section is heavily inspired by the explanation of discriminant analysis in [3]

$$f_l(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu_l}, \boldsymbol{\Sigma_l}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma_l}|^{\frac{p}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu_l})^T \boldsymbol{\Sigma_l}^{-1}(\mathbf{x} - \boldsymbol{\mu_l})\right)$$

If we label the j^{th} variable G, and denote the prior probability of variable G taking value l_m by π_{l_m} for all $l_m \in \mathcal{L}$, we then have by Bayes' theorem that,

$$\mathbb{P}(G = l_m | \mathbf{X} = \mathbf{x}) = \frac{f_{l_m}(\mathbf{x})\pi_{l_m}}{\sum_{l=l_1}^{l_L} f_l(\mathbf{x})\pi_l}$$

From here there are two approaches: linear discriminant analysis and quadratic discriminant analysis. Linear discriminant analysis works on the assumption that $\forall l_{m_1}, l_{m_2} \in \mathcal{L},\ \boldsymbol{\Sigma_{l_{m_1}}} = \boldsymbol{\Sigma_{l_{m_2}}} =: \boldsymbol{\Sigma_l}$, to achieve the following simplification ($\forall l_{m_1}, l_{m_2} \in \mathcal{L}$) [3]:

$$\log \frac{\mathbb{P}(G = l_{m_1}|\mathbf{X} = \mathbf{x})}{\mathbb{P}(G = l_{m_2}|\mathbf{X} = \mathbf{x})} = \log \frac{\pi_{l_{m_1}}}{\pi_{l_{m_2}}} - \frac{1}{2}\left(\boldsymbol{\mu_{l_{m_1}}} + \boldsymbol{\mu_{l_{m_2}}}\right)^T \boldsymbol{\Sigma_l}^{-1}\left(\boldsymbol{\mu_{l_{m_1}}} - \boldsymbol{\mu_{l_{m_2}}}\right) + \mathbf{x}^T \boldsymbol{\Sigma_l}^{-1}\left(\boldsymbol{\mu_{l_{m_1}}} - \boldsymbol{\mu_{l_{m_2}}}\right)$$

Given this is linear in $\mathbf{x}$, we can construct a maximum likelihood classifier,

$$G(\mathbf{x}) = \underset{l_m \in \mathcal{L}}{\arg\max}\left(\log \pi_{l_m} - \frac{1}{2}\boldsymbol{\mu_{l_m}}^T \boldsymbol{\Sigma_l}^{-1} \boldsymbol{\mu_{l_m}} + \mathbf{x}^T \boldsymbol{\Sigma_l}^{-1} \boldsymbol{\mu_{l_m}}\right)$$

Of course, in practice, many of these values are not known so we user the sample estimates,

$$\widehat{\pi_{l_m}} = \frac{n_{l_m}}{n},\ \widehat{\boldsymbol{\mu_{l_m}}} = \sum_{g_i = l_m} \frac{\mathbf{x_i}}{n_{l_m}},\ \widehat{\boldsymbol{\Sigma_l}} = \frac{\sum_{m=1}^{L} \sum_{g_i = l_m} (\mathbf{x_i} - \boldsymbol{\mu_{l_m}})(\mathbf{x_i} - \boldsymbol{\mu_{l_m}})^T}{n - L}$$

where n_{l_m} represents the number of instances with value l_m for the j^{th} value. This closed form expression is clearly very convenient for classification, but we must not be naive of the great limitation imposed by the two very specific assumptions on which this formula is premised. Of course, in practice these assumptions are rarely true but often we use the method anyway, so long as they are approximately true.

As before, we assess the performance of LDA (when used in synchrony with each of linear regression, k-NN regression and PMM) as an imputation method used prior to association rules analysis or cluster analysis of sparsified versions of the simulated film review dataset[9], and we illustrate the results in figure 3.7. Comparing figure 3.7 to figures 3.4, 3.5 and 3.6 we see that there is no visually discernible difference in the performance of the imputation methods as prerequisites for association rules analysis between the use of LDA and logistic regression for imputation of the Boolean variable, when either linear regression or k-NN regression is used for the imputation of the continuous variables, but that logistic regression slightly outperforms LDA in this regard when predictive mean matching is used for the imputation of the continuous variables. As far as performance for clustering purposes is concerned,

[9] See the three logistic regression imputation files available at `https://github.com/CJCharlton/Project-III-Report-Code` for the code used to impute the simulated film review dataset using LDA and each of the three described regression methods.

we again see negligible difference in performance when either linear regression or k-NN regression is co-implemented and again see that logistic regression very marginally (noticing differences for 20% removed and 40% removed but negligible difference otherwise) outperforms LDA when PMM is co-implemented. Thus, we overall see that the difference in performance between logistic regression and LDA is minimal (although that is expected since only one of the eleven variables is categorical), but that where there is a noticeable difference, logistic regression outperforms LDA.

Figure 3.7 also provides an opportunity for us to conduct a second comparison of each regression-based imputation method, this time using LDA as the control. Doing so, we observe that k-NN regression performs worse than both linear regression and PMM for both association rules analysis and performs worse than linear regression but better than PMM for cluster analysis. We observe also that PMM outperforms linear regression for association rule analysis, but that linear regression outperforms PMM for cluster analysis. These observations, are all consistent with observed performance when using logistic regression as the control, hence reaffirming our previous conclusions that linear regression is optimal when we seek to cluster and PMM is optimal when we seek to mine association rules.

3.9 Overview of single imputation methods

We have now covered a variety of single imputation methods and these methods can be classified two ways. They can be categorised firstly according to whether they impute categorical or numerical variables (for the purpose of succinctness, ordinal variables have been ignored) and secondly according to whether they impute univariately (based only on the known values for the variable to be imputed) or multivariately (using classification and regression to exploit the relations between variables). The methods covered thus far are categorised according to this 2×2 categorisation in table 3.1.

The question that is naturally raised at the conclusion of this chapter is, how can we take advantage

Imputation methods	
Univariate categorical imputation	**Univariate numerical imputation**
Mode imputation	Mean imputation
Novel category imputation	Median imputation
Random Sampling	Random Sampling
Posterior Predictive Sampling	Posterior Predictive Sampling
Classification imputation	**Regression imputation**
Logistic regression	Linear regression
Discriminant analysis	Predictive mean matching
	k-NN regression

Table 3.1: A non-exhaustive list of single imputation methods

both of the distribution of each variable and of the relations between variables when imputing missing values for datasets with missing values across multiple variables? This question is the subject of the next chapter.

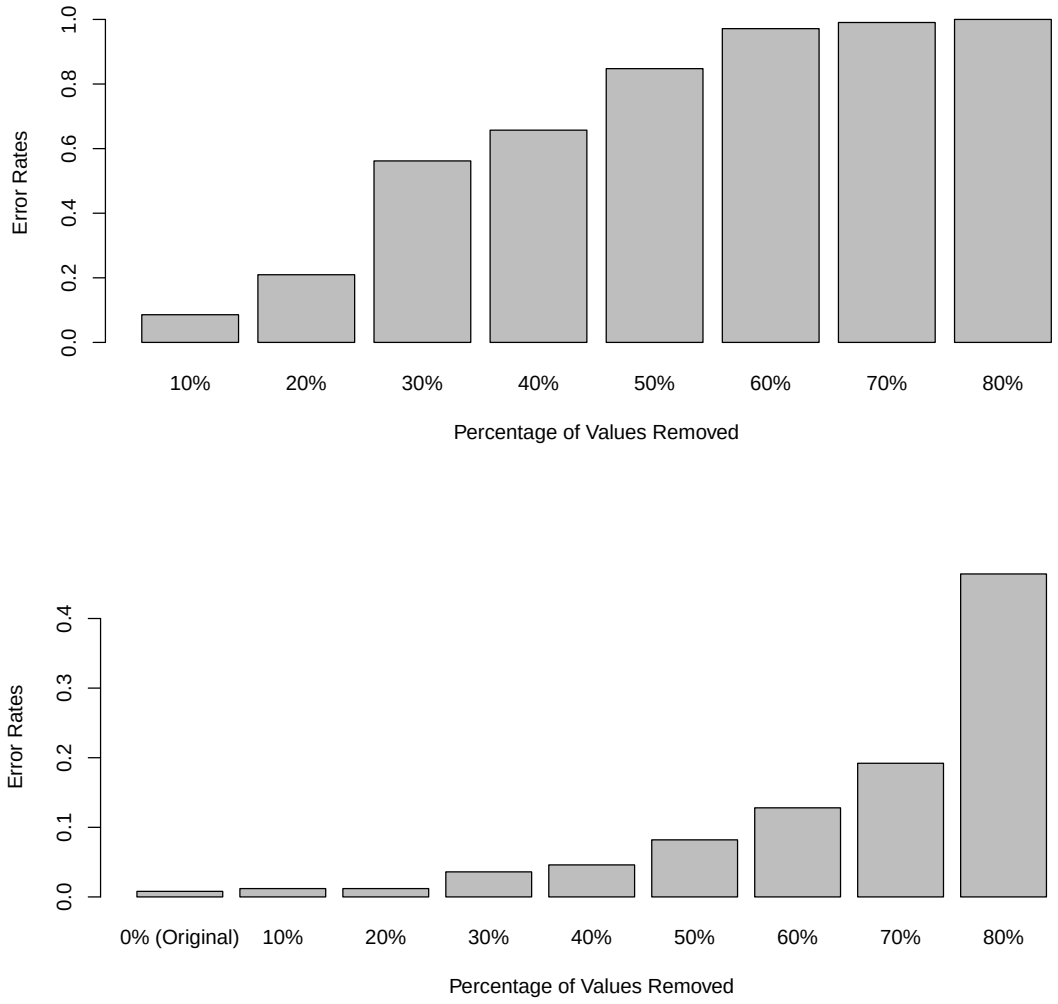

Figure 3.4: Graphs showing the association rules analysis error rates (top) and cluster analysis error rates (bottom) of joint linear regression and logistic regression based imputation (with previous random sampling for the generation of placeholder values) on the simulated film review dataset with MCAR values

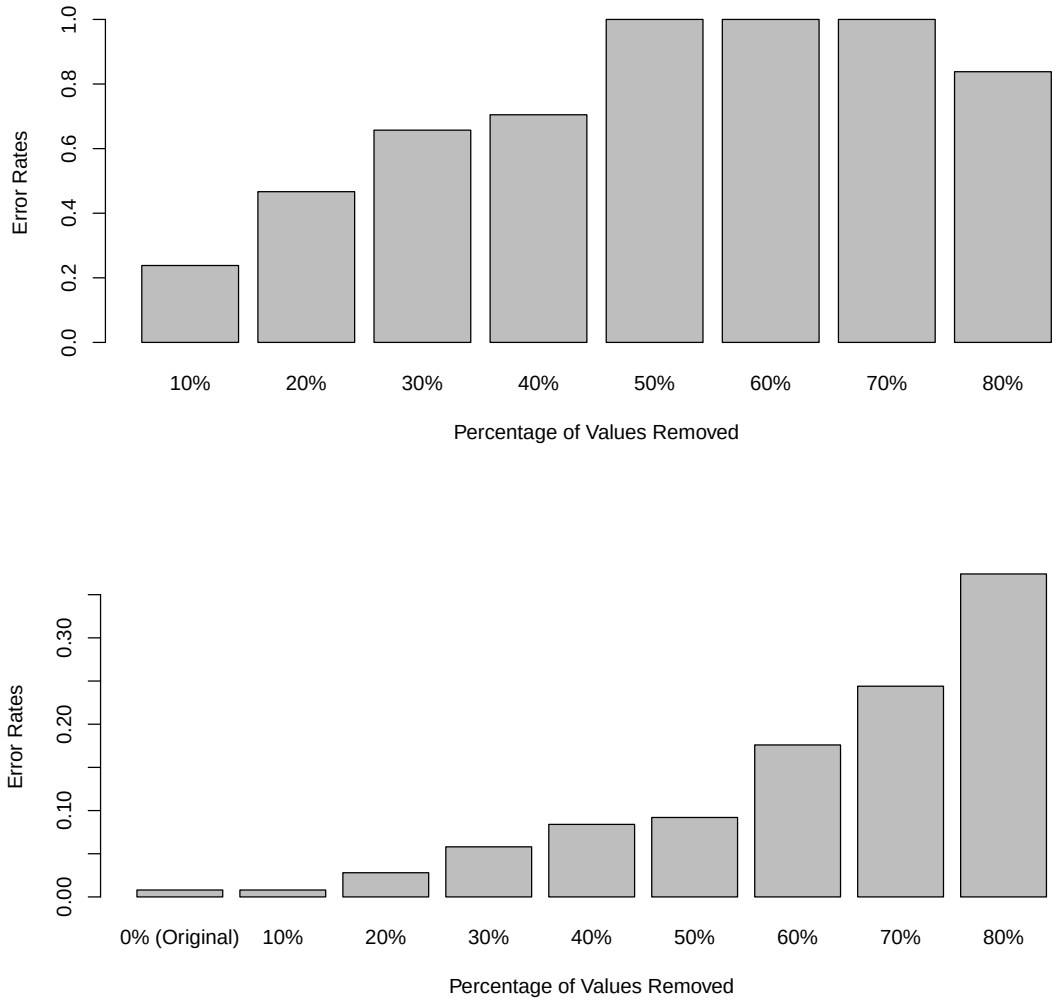

Figure 3.5: Graphs showing the association rules analysis error rates (top) and cluster analysis error rates (bottom) of joint kNN regression and logistic regression based imputation (with previous random sampling for the generation of placeholder values) on the simulated film review dataset with MCAR values

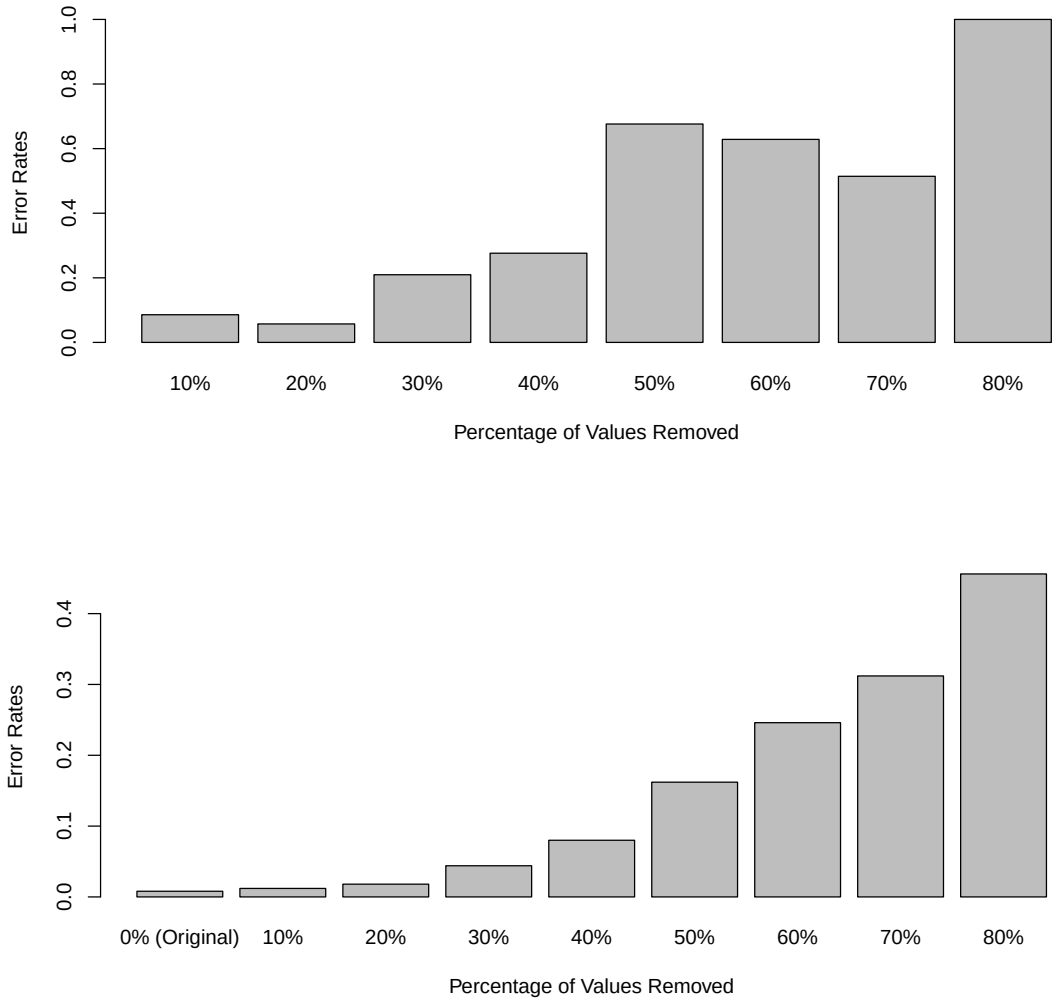

Figure 3.6: Graphs showing the association rules analysis error rates (top) and cluster analysis error rates (bottom) of joint predictive mean matching and logistic regression based imputation (with previous random sampling for the generation of placeholder values) on the simulated film review dataset with MCAR values

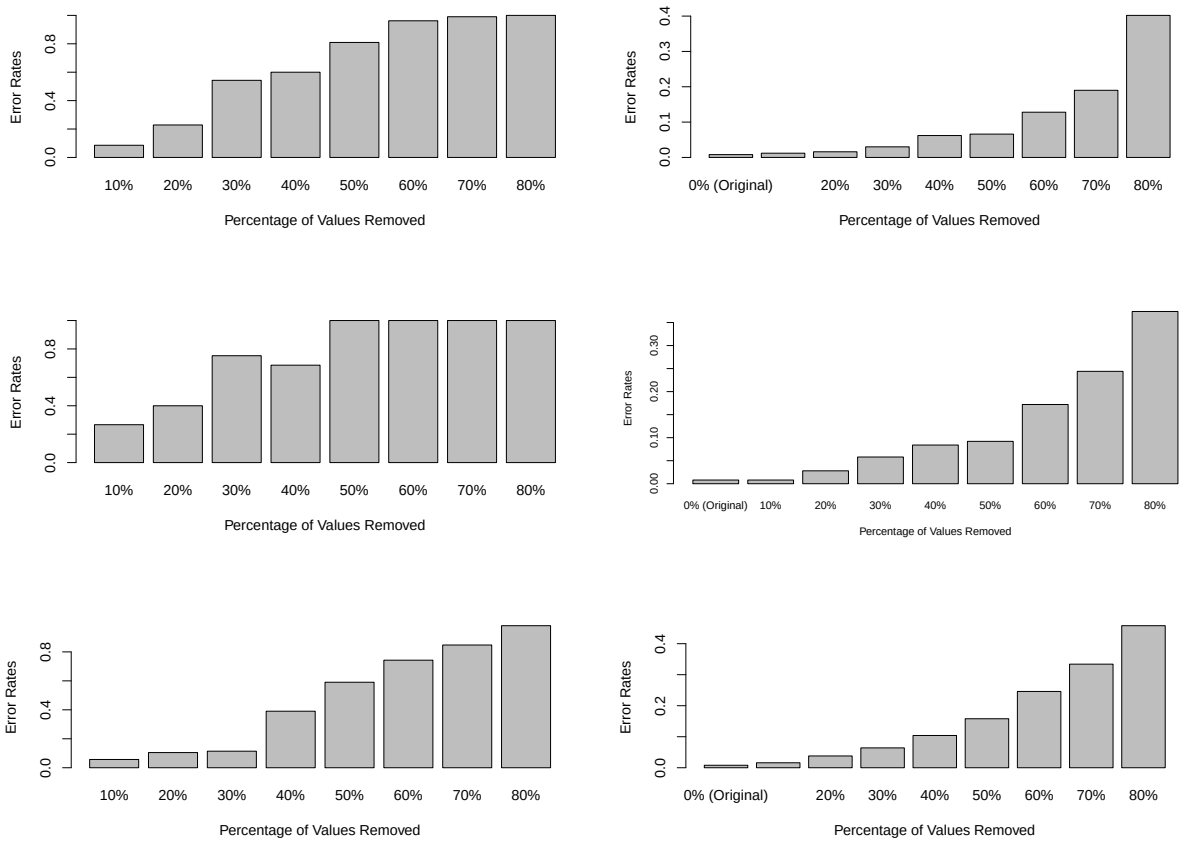

Figure 3.7: Graphs showing the association rules analysis error rates (left) and cluster analysis error rates (right) of joint linear regression and linear discriminant analysis based imputation (top), joint k-NN regression and linear discriminant analysis based imputation (vertically middle) and joint predictive mean matching and linear discriminant analysis based imputation (bottom) (all with previous random sampling for the generation of placeholder values) on the simulated film review dataset with MCAR values

CHAPTER 4

Multiple Imputation based Methods

4.1 Multiple Imputation

In the previous chapter, we looked at a range of single-imputation methods - methods in which the data is imputed once. In this section we look at multiple imputation methods - methods in which missing data is imputed multiple times. All multiple imputation methods are of the form:

1. Impute all missing values m times using single imputation methods each time. At the end of this step there will be m completed datasets.
2. On each of these m datasets, perform the analysis that would have been performed on the original dataset had there not been any missing values. In our case, this will mean perform the cluster analysis or association rules analysis on each completed dataset.
3. Pool together the results of the analyses of these m completed datasets.

Based on this methodology, one would probably expect that multiple imputation methods usually perform better than single imputation methods (in the sense that they produce more accurate results, although obviously at the cost of being more computationally expensive), and in this section we will demonstrate that this is indeed the case, but firstly let us address the issue of how to pool the results of multiple analyses. For clustering, we do this by assigning each instance to its modal assignment - that is assign it to whichever cluster it was assigned to in the greatest number of the cluster analyses (following the imputations). For association rules mining, we pool results by only including rules which were mined in at least three of the ten analyses (thus retaining most of the rules that have been mined but excluding rules that are less likely to be correct to minimise error resulting from misdiscovery of false rules). Of course there are many other possible ways to pool the results; for example, with association rules one could take all rules that are mined in at least two analyses or at least four analyses etc., depending on how strongly we weight the risk of retaining a false rule as opposed to the risk of rejecting a true one.

4.2 Multiple Imputation by Chained Equations (MICE)

Within the field of multiple imputation, one method that is widely used is multiple imputation by chained equations (MICE), which uses successive regression and/or classification (cycling through multiple times) to produce each imputation.

4.2.1 MICE Algorithm

We present this method by considering a dataset with n observations of p variables, with missing values for the q variables $\{p_{\sigma_1}, ..., p_{\sigma_q}\}$. The true values of these missing variables can be imputed as follows in the MICE method is as follows:

1. Impute all missing values using a column-wise single imputation method; we refer to the values imputed at this step as the placeholder values.
2. Reset the placeholder values for the ${\sigma_1}^{\text{th}}$ variable to NA (or missing).
3. Build a regression/classification model for the ${\sigma_1}^{\text{th}}$ variable, using the other $p-1$ variables as predictor variables, to predict and impute the missing values for the ${\sigma_1}^{\text{th}}$ variable.
4. Repeat steps 2. and 3. for the ${\sigma_2}^{\text{th}}$, ${\sigma_3}^{\text{th}}$, ..., ${\sigma_{q-1}}^{\text{th}}$ and ${\sigma_q}^{\text{th}}$ variables.
5. Continue to cycle through this omission-regression cycle until the values assigned in place of the initially missing values stabilise.
6. The process thus far has produced one completed dataset, but this is a multiple imputation based method, so if the desired number of completed datasets is C, we now repeat steps 1. to 5. $C-1$ times to yield C completed datasets.

Note then that either the choice of initial column-wise imputation method or the choice of regression/classification method must be random (or both), because if both are deterministic (e.g. using mean imputation, linear regression and logistic regression as the choices of imputation methods for these imputation steps), the resulting completed datasets will all be identical, thus defeating the point of multiple imputation. So to begin with, we will use random sampling for the first single imputation (to ensure uniqueness of each of the C completed datasets) and then either linear regression (for cluster analysis) or predictive mean matching (for association rules analysis) and logistic regression for regression and classification based imputation respectively thereafter. The reason for these choices of combinations of regression and classification based imputation is that they produced the lowest errors when used for single imputation, so it seems intuitive that they should be the most effective methods when used for multiple imputation. The appropriate numbers of cycles and imputations have largely been determined empirically and are well established in the literature; according to Raghunathan et al., 10 cycles is generally appropriate [7] (so we will use 10 cycles in our analyses) and 5-10 imputed datasets is sufficient [8] although, according to Graham et al., increasing that to as many as 40 imputed datasets continues to improve power [9], so we will stick to the upper end of the range that is considered sufficient and use 10 imputed datasets.

4.2.2 Application of MICE to the simulated film review dataset

In Figures 4.1 and 4.2 we see the results of k-means clustering of the first two imputed datasets when this MICE method is used for prerequisite imputation of missing values in the 10% and 30% randomly sparsified versions of the simulated film review dataset.[1] We see that for 10% of values missing, multiple imputation is unlikely to perform significantly better than single imputation since the first two imputations in the multiple imputation result in literally identical (every instance assigned to the same cluster in each case) clustering assignments. However, for 30% of values missing, multiple imputation is likely to perform better than single imputation because the cluster assignments from the first two imputations do differ and so the modal assignment across all ten imputations is likely to differ from the assignment from any single imputation.

Figure 4.3 illustrates the errors resulting from association rules analysis and cluster analysis when using MICE for prior imputation of missing values in the simulated film review dataset and, as can be seen, MICE does as expected perform significantly better than single imputation methods.

4.3 Missingness of Data

[2]So far we have not considered in any detail the missing data mechanism - in other words, the probabilistic mechanism that determines whether each data value is missing or not. However, this mechanism is clearly very relevant when it comes to imputing data. To demonstrate this idea and elaborate on what it means for data imputation, we must first outline the three data missingness definitions as presented by Rubin [10].

Data is said to be *missing completely at random (MCAR)* if values are missing independently, not only of the true value of the variable in question, but also of the true values of all other variables. So for example, in a film review dataset, that is generated by asking people to submit reviews online after watching each film, if some people don't submit their reviews, because the survey website goes down for a period of time, then the missing review data for those people is said to be missing completely at random. Data is said to be *missing at random (MAR)* if values are missing independently of the true value of the variable in question, but not necessarily of the true values of all other variables. So if some of the responses are missing because people who didn't like film 1 were unlikely to watch and review film 2, then those missing responses for film 2 are missing at random. Data is said to be *missing not at random (MNAR)*, if the probability that a value for the variable in question is missing is dependent on the true value of that variable. So if data is missing for a given film because only those people who strongly liked or disliked the film were likely to fill out the review survey, then the missing responses for those people who did not have extreme views is said to be missing not at random.

[1]See `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Random%20sampling%20with%20logistic%20regression%20and%20PMM%20MICE.R` for the code used to generate to impute the simulated film review dataset using MICE with random sampling, logistic regression and PMM and `https://github.com/CJCharlton/Project-III-Report-Code/blob/main/Random%20sampling%20with%20logistic%20regression%20and%20linear%20regression%20MICE.R` for the code used to generate to impute the simulated film review dataset using MICE with random sampling, logistic regression and linear regression.

[2]This section is heavily inspired by [6]

Clearly, in the case of MNAR data, imputation of values using the distribution of non-missing data is unlikely to accurately replicate what those true values would have been and instead the effect of the missing data mechanism must be taken into account when imputing missing values. This naturally raises the question: in what scenarios can we impute the missing values without taking into direct consideration the missing data mechanism? To answer this question we must introduce some new notation. Firstly, we will denote by X the full dataset, including the observed values and the true (unobserved) values of the missing data, denote by X_{obs} the set of observed values and denote by X_{mis} the set of missing values. We then define a matrix R of equal dimension to be the binary-valued (each entry takes value 0 or 1) matrix indicating whether the value of each entry in X is missing (value 0) or observed (value 1). If we then assume parametric distributions (not a very limiting assumption) for X and R, we can denote the parameters for X and R by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ respectively. We then have by Bayes' theorem that,

$$\pi_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\boldsymbol{\theta},\boldsymbol{\phi} \mid X_{\text{obs}}, R\right) = f_{X_{\text{obs}},R}\left(X_{\text{obs}}, R \mid \boldsymbol{\theta},\boldsymbol{\phi}\right)\pi_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\boldsymbol{\theta},\boldsymbol{\phi}\right)$$

Clearly, for us to be able to ignore the influence of the missing data mechanism when imputing, we require that the posterior distribution for $\boldsymbol{\theta}$ be independent of the posterior distribution for $\boldsymbol{\phi}$, or equivalently that their joint posterior distribution factorises into the product of the marginal distributions. For that to be the case, the joint prior distribution must also factorise into the product of the corresponding marginal distributions and the likelihood function for (X_{obs}, R) must also factorise into the product of the marginal likelihoods of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. So to determine when the latter requirement is satisfied, we decompose the likelihood expression thusly,

$$f_{X_{\text{obs}},R}\left(X_{\text{obs}}, R \mid \boldsymbol{\theta},\boldsymbol{\phi}\right) = f_{X_{\text{obs}}}\left(X_{\text{obs}} \mid R,\boldsymbol{\theta},\boldsymbol{\phi}\right) f_R\left(R \mid \boldsymbol{\theta},\boldsymbol{\phi}\right)$$

But once, we know which values are missing, $\boldsymbol{\phi}$ provides no further information and the distribution of R is independent of $\boldsymbol{\theta}$ so,

$$f_{X_{\text{obs}},R}\left(X_{\text{obs}}, R \mid \boldsymbol{\theta},\boldsymbol{\phi}\right) = f_{X_{\text{obs}}}\left(X_{\text{obs}} \mid R,\boldsymbol{\theta}\right) f_R\left(R \mid \boldsymbol{\phi}\right)$$

And under the assumption that missing data is MAR, we have $X_{\text{obs}} \perp\!\!\!\perp R$ and so,

$$f_{X_{\text{obs}},R}\left(X_{\text{obs}}, R \mid \boldsymbol{\theta},\boldsymbol{\phi}\right) = f_{X_{\text{obs}}}\left(X_{\text{obs}} \mid \boldsymbol{\theta}\right) f_R\left(R \mid \boldsymbol{\phi}\right)$$

So in summary, two requirements that are jointly sufficient for us to ignore the missing data mechanism (as we have done in all methods explored thus far) are:

1. Values are missing at random (MAR)
2. $\pi_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\boldsymbol{\theta},\boldsymbol{\phi}\right) = \pi_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}\right)\pi_{\boldsymbol{\phi}}\left(\boldsymbol{\phi}\right)$

Therefore, none of the single imputation methods we have explored would be appropriate for imputation of MNAR values. However, a method very similar to MICE has been proposed by Galimard et al.

as a suitable means of imputing MNAR values [11]; the only significant difference between the MICE method outlined in this chapter and the one proposed by Galimard et al. is that Galimard et al.'s algorithm uses not only the observed data but also the missingness indicator matrix for the variables with MNAR values to impute MAR and MCAR values in the dataset. Another recently proposed multiple imputation method is multiple imputation using denoising autoencoders, proposed by Gondara and Wang [12], which counter-intuitively introduces noise to imputed values only to denoise those values and in the process produce more accurate imputations. However, both these methods have been proposed quite recently (in terms of the timescale that most areas of maths develop) in publications in 2016 and 2017 respectively, and since the proposal of Galimard et al's miceMNAR algorithm, they have received some pushback with Shanahan demonstrating that for the US 2018 National Survey of Childrens Health, Galimard et al.'s algorithm produced "very large standard error estimates compared to both complete case analysis and MICE and demonstrated difficulty in providing accurate parameter estimates under MNAR conditions" [13]. Thus the field of MNAR value imputation is a fast moving field and consensus on the best methods for imputing MNAR values has not been reached. However, given many (arguably all) of the, so far most popular, methods for imputing MNAR values have been multiple imputation based methods, it seems there is consensus that single imputation methods in general cannot reliably be used to impute MNAR values.

4.4 Overview of multiple imputation

Thus, overall we have seen that multiple imputation methods do typically produce more accurate results than single imputation methods when used to impute MCAR values prior to association rules mining or clustering and that the single imputation methods we have explored are insufficient to accurately impute MNAR values, while there is evidence that there are some multiple imputation methods that are sufficient to do so. However, multiple imputation methods are clearly much more computationally expensive to implement and so in some scenarios single imputation or even complete case analysis are still preferable, for example if the proportion of values missing is so low that even these latter methods still produce quite accurate results.

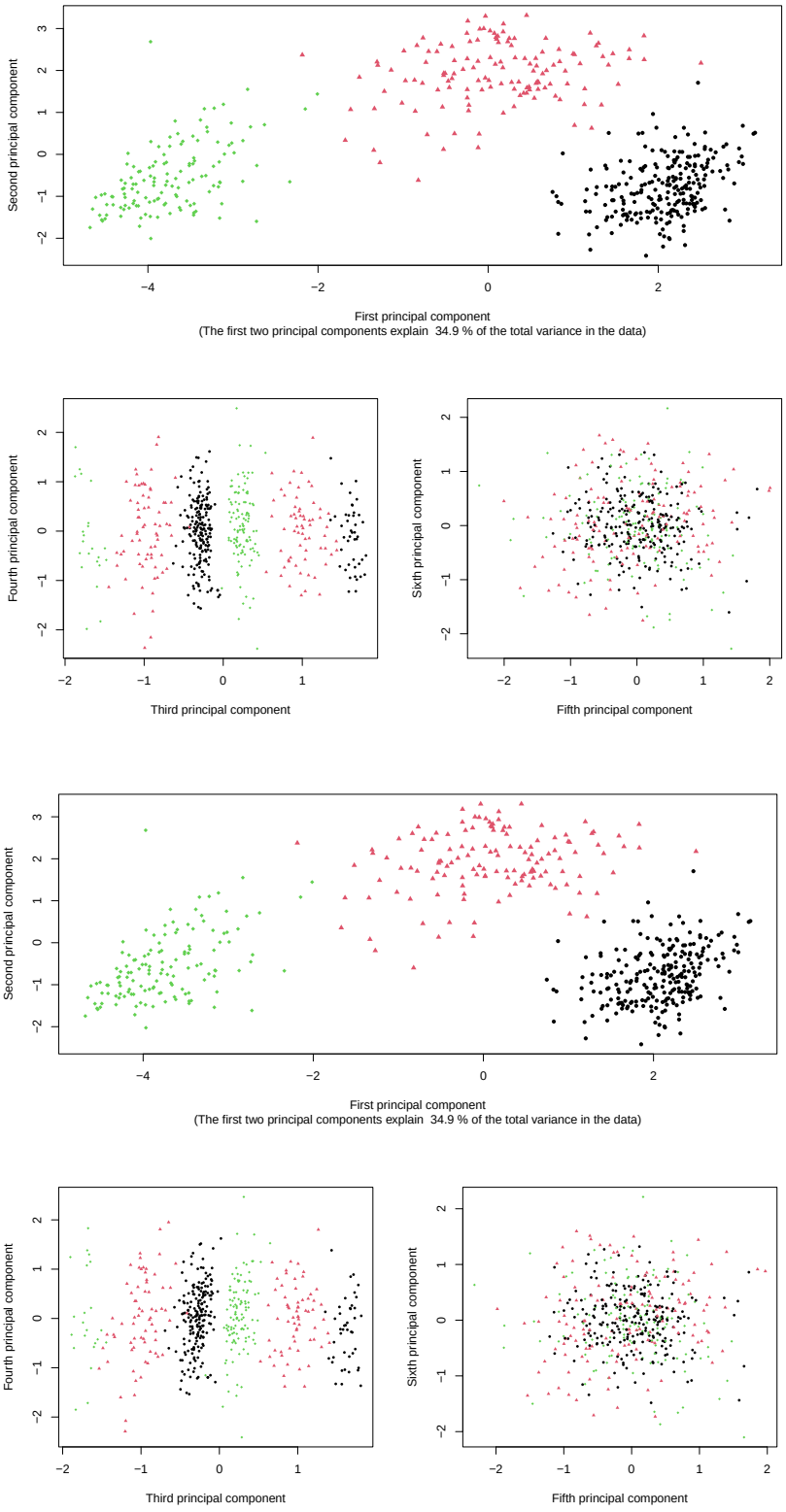

Figure 4.1: Graphs depicting the k-means clustering of the first two imputations of the simulated film review dataset, with 10% of values MCAR, by the MICE algorithm

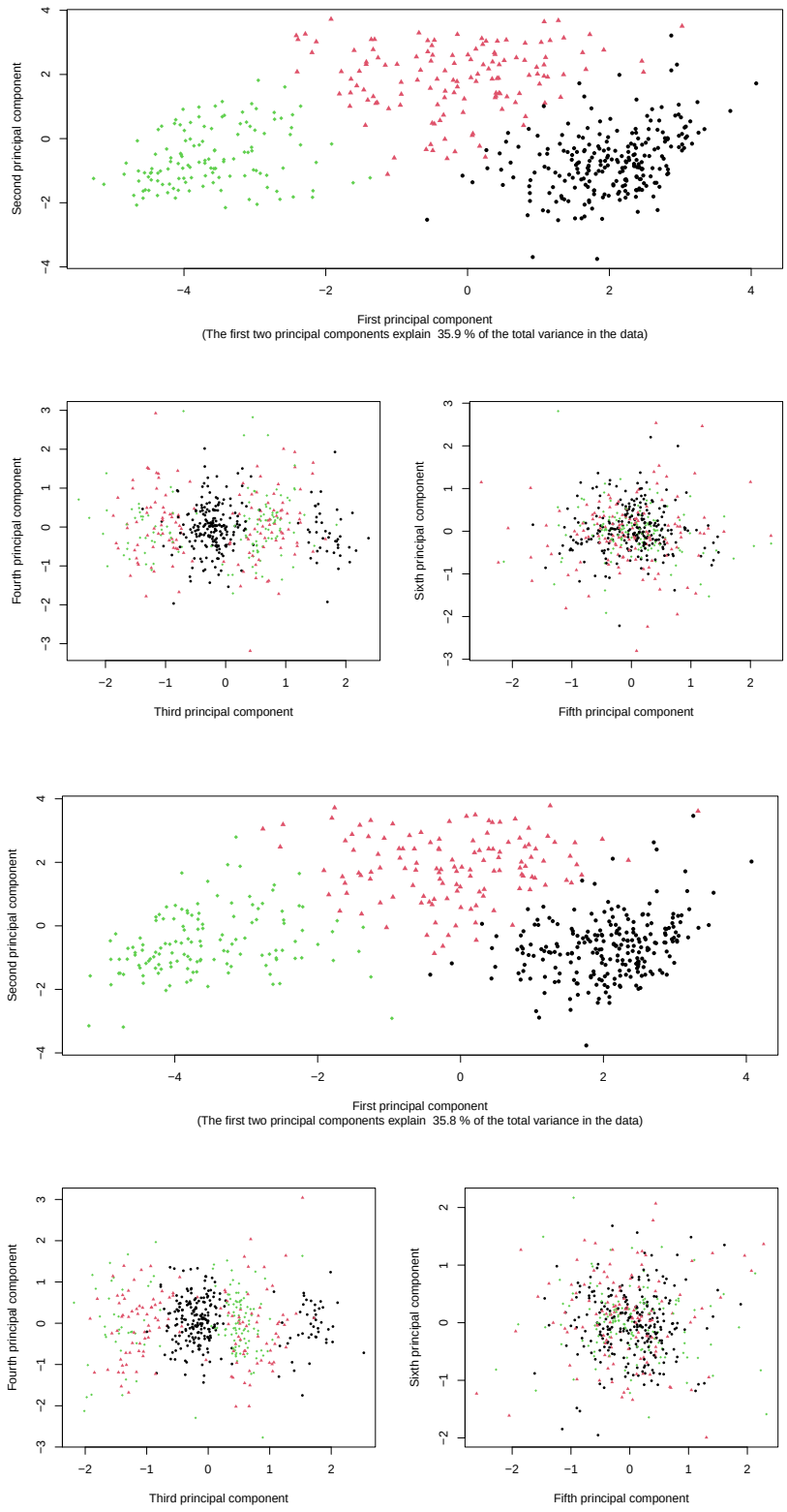

Figure 4.2: Graphs depicting the k-means clustering of the first two imputations of the simulated film review dataset, with 30% of values MCAR, by the MICE algorithm

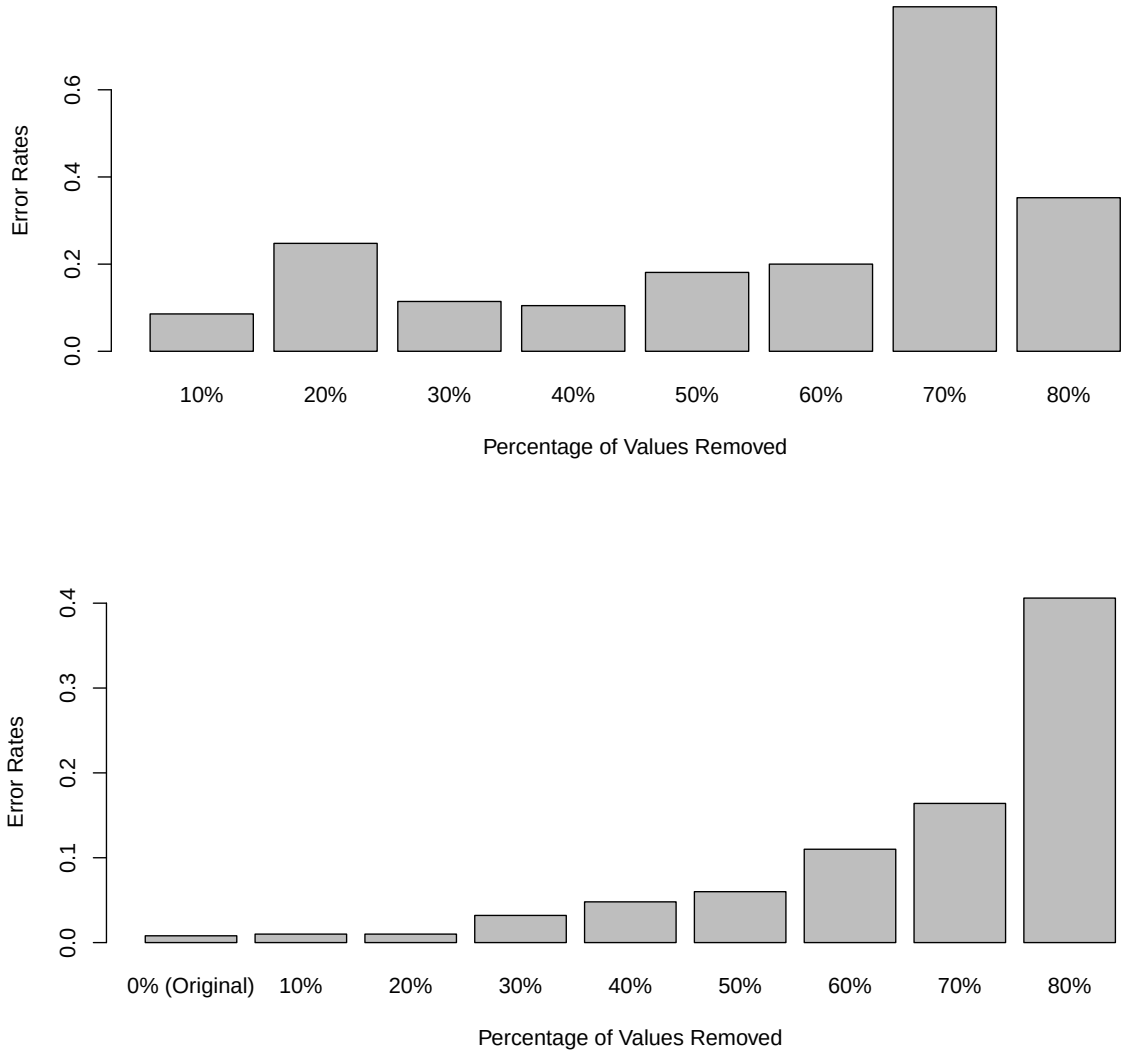

Figure 4.3: Graph showing the association rules analysis error rates (top) and cluster analysis error rates (bottom) of MICE imputation (with previous random sampling for the generation of placeholder values) on datasets with MCAR values

CHAPTER 5

Conclusion

Overall, the field of unsupervised machine learning is vast and at this point in the report, it is important for us to reflect on how narrow our focus has become as we have progressed through this report. We started by defining what machine learning is and how it can be subcategorised into supervised and unsupervised. We then progressed into exploring in depth just two fields of unsupervised learning - association rules analysis and cluster analysis - but there are many other such fields, such as self-organising maps and non-linear generalisations of principal component analysis to curves and surfaces [3] to name just two.

Within association rules analysis, we explored just two potential methods of clustering - the apriori algorithm and an adapted supervised learning approach - but this leaves several methods unstudied, such as the popular eclat and FP-growth algorithms. Within cluster analysis, we again explored two algorithmic methods (as well as exploring the less rigorous method of visual identification with the assistance of principal component analysis) - k-means and k-medoids. But we again emphasise that there are many other clustering methods, including both other optimisation clustering methods as well as clustering methods in fundamentally different fields of cluster analysis, such as hierarchical clustering.

When exploring imputation methods, we specifically sought to investigate how well methods worked as prerequisites for association rules analysis and cluster analysis and not just for any generic or unspecified statistical analysis, as most literature on the topic assumes. But of course, we were limited in this endeavour by the fact that we have only scratched the surface on the range of potential methods for cluster analysis and association rules analysis. Due to report length limitations, we then had to restrict ourselves further, by testing imputation methods in conjunction with just one analysis method, in each case - the apriori algorithm for association rules analysis and the k-means algorithm for clustering. It is important then to be aware how narrow our focus is in the latter chapters of this report.

In chapter 3, we explored several non-imputation and single imputation methods. We concluded

that non-imputation methods are largely ineffective - in particular the error rates of complete case analysis explode rapidly (much more rapidly than all explored imputation methods) as the proportion of values missing increases, and hence is only appropriate when the proportion of values missing is extremely low, when it could (depending the context of application and cost of error) be argued that any potential error is insignificant enough that the effort required to impute values is unjustified. As for our brief discussion of the adaptation of the unsupervised learning methods to be directly applicable to incomplete data, we concluded that any such adaptation is built on logically spurious foundations and so should not be used.

In our subsequent investigation of imputation methods we explored what could be considered four different classes of imputation methods: average based imputation, sampling based single imputation, regression and classification based single imputation and multiple imputation. We conclude that random sampling is always preferable to any form of average based imputation, due to the ease with which random sampling can be conducted and its tendency to less strongly bias the variance of imputed values. We also argued that, when used in isolation, there is little value in using posterior predictive sampling over random sampling (from the observed data) due to it being significantly more computationally involved and not producing better enough results to justify this greater burden of effort on behalf of the person conducting the imputation. We explored three regression based imputation methods and two classification based imputation methods and concluded that the best combination for association rules analysis was predictive mean matching for imputation of continuous variables and logistic regression for imputation of categorical variables, whilst the best combination for cluster analysis was linear regression and logistic regression respectively.

Progressing onto multiple imputation methods, we then tested the use of MICE with logistic regression and linear regression (when applied to cluster analysis) or predictive mean matching (when applied to association rules analysis) for the same purpose and again found it to produce better results than all previously explored methods. We then discussed the additional benefit of multiple imputation when working with MNAR values and demonstrated that the use of single imputation when handling a dataset with a significant proportion of MNAR values, would likely produce inaccurate results.

Thus in conclusion, this report has found that (ignoring the computational cost and the comprehensibility for those from less mathematical backgrounds) multiple imputation is the best approach to imputing missing values prior to both association rules analysis and cluster analysis. Therefore to justify using single imputation methods or complete case analysis, we must know that the error resulting from the use of these methods is likely to be low enough to argue that the greater computational cost and inaccessibilty of the model to people from less mathematical backgrounds is not justified. For that to be the case, we must know that the data is missing MCAR (or at least MAR) and we must ensure that the portion of values missing is low enough [1].

[1] See error graphs throughout the report for specific values for what proportion of mising values is considered low enough.

Bibliography

[1] K. Murphy, *Probabilistic Machine Learning An Introduction*. 2022. 1.1, 1

[2] J. Heaton, "Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms," in *SoutheastCon 2016*, pp. 1–7, IEEE, 3 2016. 2.1.2

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. 2017. 2.1.3, 2.2.2, 2.2.2, 2.2.3, 2.2.3, 8, 3.8, 5

[4] B. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*. 2011. 6, 2.2

[5] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of Performance of Data Imputation Methods for Numeric Dataset," *Applied Artificial Intelligence*, vol. 33, pp. 913–933, 8 2019. 3.6

[6] S. v. Buuren, *Flexible Imputation of Missing Data*. 2 ed., 2018. 3.7, 2

[7] T. Raghunathan, P. Solenberger, P. Berglund, and J. Van Hoewyk, "IVEware: Imputation and Variance Estimation Software (Version 0.3) 1," tech. rep. 4.2.1

[8] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: What is it and how does it work?," *International Journal of Methods in Psychiatric Research*, vol. 20, pp. 40–49, 3 2011. 4.2.1

[9] J. W. Graham, A. E. Olchowski, and T. D. Gilreath, "How many imputations are really needed? Some practical clarifications of multiple imputation theory," *Prevention Science*, vol. 8, pp. 206–213, 9 2007. 4.2.1

[10] D. B. Rubin, "Inference and Missing Data," vol. 63, no. 3, pp. 581–592, 1976. 4.3

[11] J. E. Galimard, S. Chevret, C. Protopopescu, and M. Resche-Rigon, "A multiple imputation approach for MNAR mechanisms compatible with Heckman's model," *Statistics in Medicine*, vol. 35, pp. 2907–2920, 7 2016. 4.3

[12] L. Gondara and K. Wang, "MIDA: Multiple Imputation using Denoising Autoencoders," 5 2017. 4.3

[13] B. Shanahan, *Evidence for a multiple imputation approach to MNAR mechanisms*. PhD thesis, Ball State University, Muncie, 7 2021. 4.3