Joint Cohort Detection & Predictive Modelling alongside Safe Model Updating

SAMUEL EMERSON

Joint Cohort Detection & Predictive Modelling alongside Safe Model Updating

Samuel R. Emerson

A Thesis presented for the degree of
Doctor of Philosophy

Department of Mathematical Sciences
Durham University
United Kingdom
October 2023

Abstract

A common objective provided by stakeholders, given a supervised dataset, is to construct a predictive model of the response given the covariates. If a clustering structure is suspected (such that different clusters interact with the response in different ways) then an additional objective may be given to detect these clusters, or cohorts, such that interventions based on the predictive model can be adapted for each group.

The solution to this problem requires a balanced handling of both objectives through a joint cohort detection and predictive modelling method. Previous solutions to this issue often favour one objective over the other. Indeed, cohort detection takes prevalence for unsupervised clustering methods such as K-means (which are followed by cluster-specific models for prediction), whereas accurate prediction takes prevalence for supervised clustering methods such as mixture models (which use clustering solely as a tool for more accurate modelling).

This thesis aims to provide a method that focuses on cohort detection by providing a non-probabilistic partitioning of the data whilst simultaneously focusing on accurate predictive modelling by allowing the Bayesian evidence of the model to dictate the partition. A graphical representation of the data is constructed to ensure the partitioning both respects the structure in covariate space and reduces the number of possible partitions (and hence models) one would have to consider. The latter point is particularly important as the Bayesian evidence is determined through Sequential Monte Carlo, a computationally expensive but necessary process used to ensure the estimated measure that selects the partition is accurate. This method has an associated `R` package (`UNCOVER`) for implementation.

Finally, a separate contribution is discussed in this thesis surrounding the topic of safe modelling updating. Specifically, this refers to the use of hold-out sets when updating a model to avoid interventions negatively impacting model quality. Contributions to this field are: a method of locating the minimum hold-out set size through Gaussian process emulation of a total cost function and a discussion on the impacts of clustering in this setting.

Declaration

The work in this thesis is based on research carried out at the Department of Mathematical Sciences, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

First and foremost I would like to express my gratitude to my supervisor Louis Aslett. His dedication, be that through extended meetings lasting much longer than I should reasonably expect or by helping me through the technical difficulties that constantly arose through lockdown and beyond is truly appreciated. Most importantly however is his willingness to bounce around ideas and encourage exploration of new avenues of research, no matter how trivial.

Louis's great effort in encouraging me to collaborate with institutions other than Durham University resulted in me meeting the next person I would like to thank — James Liley. From working on the SPARRA project together through to the various spin-off papers that grew as a consequence, collaborating with James has always been a pleasure. Indeed, along with Louis his help and guidance with writing papers for publication was incredibly important, and allowed me to greatly improve the standard of my own research.

Outside of academia I would like to thank my partner Chloe Gordon for her patience throughout my exploration into a new research environment and her support with the inevitable stresses that followed. I would also like to give thanks to my parents, Jane Emerson and Jason Emerson for their collective belief in me and my aspirations. Friends, made both prior to and during the doctorate, are also deserving of recognition. Whilst there are many people who have helped a great deal, I want to give particular thanks to Matthew Foskett for his helpful distractions away from the PhD and to Muhammad Hasan as well as Qasem Tawhari for their encouragement and friendliness when in the department. Additionally, I would like to give thanks to Neill Johnstone for his part in proofreading this thesis.

Finally, I would like to round off by thanking my second supervisor, Jochen Einbeck, and the entire Maths Department for the inclusive and warm atmosphere during my time at Durham, and to EPSRC for giving me the funding — without which my postgraduate journey would not be possible.

Contents

List of Figures

List of Tables

Dedication

Dedicated to, and in memory of, my wonderful and kind-hearted fetherio: Jason Emerson

CHAPTER 1

Introduction

1.1 Motivation

Clustering observations into similar groups for inference has long been used in the fields of medicine [1], finance [2], psychology [3] and beyond. As clustering techniques develop in academia, one would assume that these new methods follow through to adoption in practical settings outside of academia; however, this tends not to be the case.

Applications that require the clustering of data have the propensity to rely on unsupervised methods such as K-means [4], even if the initial task is one of prediction with respect to a response. As a consequence, the inference made on the relationship between the response and the covariate data can be adversely affected by a possible random partitioning of the observations.

Inclusion of the response within the process of developing clusters is well established, with methods ranging from finite mixture of regressions [5] to mixture of experts [6]. As well as response incorporation, development of clustering techniques typically have favoured incorporation of uncertainty through soft clustering — the probabilistic assignment of observations to clusters. If the initial task is prediction,

some methods bypass the need for cluster assignment entirely. These supervised methods represent a loss in interpretability with respect to cluster assignment, which explains the favouring of unsupervised methods in practice.

One may liken this scenario to the trade-off between interpretability and predictive power often present in statistical learning algorithms. This concept is seldom discussed in the clustering setting, with the consequence being a heavy imbalance towards either: interpretability (for applications of clustering methods) or predictive power (for the theoretical advancement of clustering techniques).

The development of methods which provide a more balanced trade-off are the main focus of this thesis. Interpretability is considered through the assignment of a sole cluster to each observation, known as hard clustering, with the clusters themselves containing observations which have similarities in certain attributes. To further aid interpretability, these attributes may be selected by the stakeholder (the person for whom interpretability has the most importance). Aspects of predictive power are implemented by: including the response when determining the clusters and framing the problem in a Bayesian setting, which incorporates uncertainty in the parameters used to infer the relationship between the covariates and the response.

Examples of where a more balanced method could be desirable are scenarios where cluster assignment is given equal importance to the prediction of a response. Such a situation can arise in medical settings, where cluster assignment directly results in patients being grouped into cohorts. Here one could imagine a scenario where a data scientist is given the task of developing a model that predicts the risk of a contracting a disease, but is given the additional information that experts suspect that different cohorts of patients depend on their health data differently for determining disease presence. Here it is crucial not only that the model can predict the risk accurately, but also that we can determine patient cohorts accurately in order to devise separate prevention techniques. Similar scenarios emerge in finance, where devising group specific loan schemes is coupled with prediction of whether an individual is likely to default on a loan.

Utilising the prediction model in a more abstract way, one could use such a balanced method as an initial piece of inference to uncover different cohorts, and

then use that information to ensure a representative hold-out set is selected. Hold-out sets can be particularly useful for generalisability or to prevent intervention effects when sequentially updating a model.

Devising a method to provide a solution to these types of problem is the primary focus of this research.

1.2 SPARRA: Scottish Population At Risk of Readmission and Admission

The motivating example behind this thesis is a particular project run by NHS Scotland — SPARRA [7]. The current version of SPARRA, SPARRA v4, utilises the majority of the Scottish population's electronic health records in order to construct an ensemble of various machine learning models, which aim to predict (for the Scottish public) the risk of emergency admission within the next year. The response recorded was whether the patient had an emergency admission to hospital within the following year of their health data being recorded, and therefore was binary. Currently in the fourth version (v4) of the model, the third version (v3) was the first to incorporate the majority of the Scottish population. The data used to train the v3 and v4 models were people's recorded health data, therefore a member of the public is only represented in the training data if they have had a prior interaction with the Scottish health system during the time period that the data was collected.

Specifically SPARRA v3 was designed to be a collection of three logistic regression models for three different cohorts, derived by medical professionals, which accounted for all of the training data. These were [8]:

1. Frail Elderly — patients aged 75 and over
2. Long Term Conditions — patients aged from 16 to 74
3. Younger Emergency Department — patients aged from 16 to 55 and have attended an emergency department within the last year

It is important to note these cohorts did not completely partition the data; it is possible to encounter overlap for the long-term conditions and younger emergency

department cohorts. In terms of prediction, if a patient could be represented by two cohorts then the maximum of the two 'risk scores' (predictions) given by the two logistic regression models was taken as that patients risk score.

This overlap in cohorts raises an interesting query: if separate action plans to reduce emergency admission (utilising the predictive models created) were developed for each of the cohorts, how would one assign an action plan to a patient represented in two cohorts? Assignment to the plan whose model gave the maximum score is less suitable here as the aim now is not simply to identify high-risk patients. Indeed, if a patient is assigned a higher risk score than needed through this maximisation policy when just identifying high-risk patients, the damage to the patient being misidentified is minimal as the result is just potentially more attention being focused on the patient. However, if the goal is to derive action plans per cohort, then the actual score being accurate does now play a large role, with incorrect cohort assignment leading to a potentially ineffective plan being applied to a patient.

Therefore if cohort-tailored interventions are considered, redefining the cohorts may be a legitimate requirement. When redefining these cohorts, one could allow the data to act as the determiner, creating cohorts that are: interpretable to medical professionals (in the manner of SPARRA v3) and able to produce accurate risk scores by capturing cohorts differing covariate–response relationships. This is exactly the scenario detailed in section 1.1. Therefore, the SPARRA project offers a setting where development of a new model that balances interpretability and predictive power has real practical use, along with a direct competitor (v3) in which to compare.

1.3 Contributions Overview

1. A new methodology (named UNCOVER) for the detection of cohorts whilst simultaneously creating an accurate predictive model (Section 4.5), with several distinct features including:

   (a) Separate processes for the modelling of the data and the derivation of the possible partitions of the data, allowing different selections of covariates for different stages of the algorithm. This gives stakeholders greater con-

trol of the cohort structure, whilst still ensuring the detection of cohorts is data driven (Section 4.1.1).

(b) An incorporation of Bayesian evidence generation into Chopin's Iterated Batch Importance Sampling scheme [9] (Section 4.2).

(c) A corrective algorithm which can combine cohorts as well as split them, which attempts to combat the possible drawbacks greedy algorithms encounter (Section 4.3).

(d) An expansive list of 'deforestation' criteria which combine clusters to increase the outputted cohort's generalisability to new data, as well as meeting potential stakeholder demands (Section 4.4).

2. An application of function memoisation [10] which provides Sequential Monte Carlo algorithms with more convenient initial distributions (Section 5.1).

3. An adaptation of Iterated Batch Importance Sampling that removes observations from the posterior instead of adding them, which requires a bias correction technique (Section 5.2).

4. An `R` package `UNCOVER` [11], for ease of use when implementing the UNCOVER method (Section 5.5).

5. Applying the technique of expected improvement using a Gaussian process emulator to the field of optimal hold-out sets, which formed a component of the paper [12] detailing the use of hold-out sets as a solution to the issue of model updating [13] (Section 7.3).

6. An examination into the effects clustering may have on the problem of safe model updating and the specific role clustering plays when selecting an optimal hold-out set size (Section 7.4).

1.4 Outline

The structure of the thesis as well as a brief introduction to each chapter is given below. Note that Chapters 2 and 3 form the literature review, with Chapters 4 through

7 detailing the numerous research contributions from this PhD. Finally, Chapter 8 provides a summary of the work together with avenues for future research.

Chapter 2 — Supervised & Unsupervised Clustering This chapter provides a literature review to introduce the reader to the existing methods that can be used to tackle joint cohort detection and predictive modelling, and the shortcomings of these methods for this particular problem.

Chapter 3 — Bayesian Frameworks & Graphical Representations of Data A secondary literature review chapter is given, detailing the methodologies behind the novel UNCOVER model discussed in Chapter 4. A Bayesian framework is compared to a frequentist approach in the context of clustering, and basic graph theory together with its existing use for clustering problems is introduced.

Chapter 4 — UNCOVER: Utilising Normalisation Constant Optimisation Via Edge Removal This chapter provides the main methodological contribution of the thesis, detailing a novel modelling framework which tackles the issue of joint cohort detection and predictive modelling directly. This is done through graphically representing covariate data, which provides a mechanism for creating clusters by considering both the structure of the covariate data itself as well as the consequences forming particular clusters has on model quality. Additional contributions which provide methods of improving generalisability to new data are also detailed.

Chapter 5 — Implementation of UNCOVER The practical aspects of implementing UNCOVER are discussed in this chapter. After identifying potential bottlenecks, computational solutions are derived and integrated into UNCOVER. This requires the introduction of novel methods for integration of existing practices (such as memoisation), as well as the development of new contributions which extend existing SMC methodology. The chapter concludes by detailing a software contribution in the form of the `UNCOVER` package, developed in `R`.

Chapter 6 — Application of UNCOVER In this chapter various aspects of UNCOVER are tested on both synthetic and real-world datasets. These datasets cover both settings; when the clustering structure is known a priori and when it is not. Additionally, a careful examination of UNCOVER is made when clusters overlap, a common issue when considering noisy datasets.

Chapter 7 — Optimal Hold-out Sets: An Application in Updating Risk Scores The final contribution to this thesis addresses a separate topic: safe model updating. Deviating away from joint cohort detection and predictive modelling, safe model updating provides a solution to the dangers of naïve model updating through the use of hold-out sets. Selection of an optimal hold-out set with respect to size is detailed, and the relation of this work to the UNCOVER method is discussed. The chapter describes the authors own contributions to a wider project on this topic.

CHAPTER 2

Supervised & Unsupervised Clustering

This chapter serves as an introduction into clustering techniques that are either used in recent applications or recently developed. The distinction between application and theory here, in general, can be divided into unsupervised and supervised clustering. Unsupervised methods focus on providing interpretable clusters which is appealing in practice, while supervised methods focus on generating a model that offers generalisability through uncertainty quantification, which is more appealing from a theoretical standpoint.

There is naturally some overlap between supervised and unsupervised methods with respect to interpretability and generalisabilty. As a result it must be noted that the popularity of a method, either for theoretical or applied purposes, does not represent an automatic appeal for the goals of joint cohort generation and predictive modelling. All methods showcased in this chapter contain useful and harmful aspects for joint modelling, with extraction of positive design elements a challenge detailed in subsequent chapters.

2.1 Unsupervised Clustering

In the unsupervised setting we are able to access covariate data, which we can represent as a matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T \in \mathbb{R}^{n \times p}$ formed of n observations on p variables. However, in contrast to a supervised or semi-supervised setting, for each of these observations we do not have access to (or there does not exist) an associated response. It is also not uncommon to simply ignore a response even if it is accessible. Trivially this lack of response indicates that cohort or cluster detection in such settings is attempted through a partitioning of the covariate data, which can be represented as a partition of an index set into K subsets

$$\mathfrak{V} = \{1, \ldots, n\} = \bigcup_{k=1}^{K} \mathfrak{V}_k \tag{2.1}$$

where $\mathfrak{V}_k \neq \emptyset \ \forall \ k = 1, \ldots, K$ and $\mathfrak{V}_k \cap \mathfrak{V}_l = \emptyset \ \forall \ k \neq l$.

The lack of response, and the tendency to partition the data as opposed to the covariate space where the data lies, leads to unsupervised methods having somewhat of a self contained output. Observations provided are assigned a cluster, however, separate additional methods are required to assign new observations to a cluster, many of which utilise similar techniques to that of the algorithm that produced the initial clustering.

Unsupervised methods can produce either a hard clustering or a soft clustering output. Whilst soft clustering (probabilistic assignment of clusters to observations) unsupervised methods often provide supervised counterparts[1], the main focus of this section will be on hard clustering (where each observation is assigned to one and only one cluster), as the interpretability of these outputs is crucial to their popularity in applied settings.

Finally, we note that although unsupervised methods are performed without knowledge of a response, one could still produce a predictive model in a sequential manner using unsupervised techniques. There are various ways in which this could

[1]For example finite mixture models for $\pi(X)$ being altered to finite mixture models for $\pi(Y|X)$ to include the response.

be done, with a general overview being that a predictive model is generated *given* knowledge of the clusters (i.e. the clustering information is deduced beforehand using either a hard or a soft clustering unsupervised method). The viability of this approach is discussed in section 2.1.3.

2.1.1 K-means Clustering

A technique first introduced in the 1960s by James MacQueen [4], K-means clustering still features prominently in recent practical applications [14,15]. The algorithm can be initialised by selecting K (where K is predetermined by the user) observations at random to represent the initial value for the centroids $\mathbf{c}_1, \ldots, \mathbf{c}_K$, which then form a centroid matrix $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_K)^T$. Observations which have a common nearest centroid, with 'nearest' defined by the Euclidean distance, are defined as being in the same cluster[2]. Using this cluster assignment, the centroids are recalculated as the mean value of the observations that are closest to them, and then the process repeats until convergence. The repetition here is crucial as the centroids dictate the clustering, and so by updating the centroids we ensure that they are more representative of the observations assigned to them. However, updating the centroids then by design requires the updating of the cluster assignment, and so repetition is key to ensure stability in the final output. This procedure is showcased in algorithm 26, found in appendix A.1, along with a description of the regions of covariate space that are formed through K-means clustering.

In the finite data setting the output of a K-means algorithm will have the property that any two clusters are linearly separable in covariate space[3].

Definition 2.1.1 (Linearly Separable Clusters)**.** *Let* $\mathbf{a} \in \mathbb{R}^p$ *and* $b \in \mathbb{R}$ *be constant. Two clusters,* $\mathfrak{V}_k$ *and* $\mathfrak{V}_l$*, are linearly separable if there exist* $\mathbf{a}, b$ *defining a hyperplane* $\mathbf{a}^T\mathbf{x} = b$ *such that* $\mathbf{a}^T\mathbf{x}_i < b \ \forall \ i \in \mathfrak{V}_k$ *and* $\mathbf{a}^T\mathbf{x}_j \geq b \ \forall \ j \in \mathfrak{V}_l$*. Such a hyperplane is known as a separating hyperplane.*

[2]If an observation has two or more nearest centroids, then typically one centroid is selected at random to be that observation's nearest centroid.

[3]Provided every observation in $\mathbf{X}$ is unique. If not then it is entirely possible for a point in covariate space to be assigned to two different clusters, and therefore the clusters are not separable.

This is clear by considering a hyperplane $\mathcal{H}$ that contains the centroids of both the clusters in question. The hyperplane that is perpendicular to $\mathcal{H}$ and contains the midpoint of both the centroids is a separating hyperplane[4].

This linear separability property that exists for clusters formed using K-means is an implicit assumption the user makes when selecting this algorithm — an assumption that may not be valid for a wide range of datasets. Indeed, seldom seen in practical applications of K-means is the justification for this linear separability, which becomes less trivial as the dimensionality of the data becomes larger. Furthermore, the random initialisation method leads to variability in the cluster assignments. This can be remedied through multiple runs of the algorithm at different starting points and selection of the best output, measured by the within-cluster sum of squares WCSS $= \sum_{k=1}^{K} \sum_{i \in \mathfrak{V}_k} \|\mathbf{x}_i - \mathbf{c}_k\|_2^2$, but as the algorithm can only be run finitely many times the drawback is uncertainty in whether our final output is a local or global minimum of WCSS. In a healthcare scenario this could have severe negative consequences for patients near a decision boundary. Finally, the requirement to have complete certainty in the number of clusters prior to the running of the algorithm becomes problematic if there is indeed uncertainty in the number of clusters for the desired output. There are methods that can assist in the selection of K, such as the silhouette method [16], however, there is not a theoretical justification for using this metric. Another possible metric to base selection of K on, which does have some theoretical justification, is the gap statistic method [17]. This method is detailed in appendix A.3.

Despite these drawbacks one cannot dispute the popularity of K-means clustering, as under the criteria of interpretability and ease of use the algorithm performs well. The hard clustering output is guaranteed with this method and the partitioning of the covariate space (see appendix A.1) lends itself to the notion of observations in the same cluster having similar attributes through the connectivity of the regions. These advantages are coveted by stakeholders that require information on

[4]Some separating hyperplanes of this design can be obtained through extension of the decision boundaries of the Voronoi diagram produced from the K-means algorithm (see appendix A.1 and figure A.1).

the cohorts, even if the primary function of the data is prediction of a response.

2.1.2 Hierarchical Clustering

If the linear separability property present for K-means clusters is not representative of the data, then one may be inclined to resort to another unsupervised method — hierarchical clustering. Hierarchical clustering groups observations through a greedy process to create a hierarchy of clusterings, which can be expressed visually through a dendrogram (see appendix A.2 for details). Initially one must choose a particular form of hierarchical clustering:

1. **Agglomerative**: Initialises with a cluster per observation and then combines two clusters into one at each iteration.

2. **Divisive**: Initialises with one cluster and then splits one cluster into two at each iteration.

In addition to this, a linkage method must also be specified along with a distance metric (the Euclidean distance is the typical choice).

Definition 2.1.2 (Linkage Method)**.** *Given a distance metric* $d : \mathbb{R}^p \times \mathbb{R}^p \to [0, \infty)$*, covariate data* $\mathbf{X}$ *and observation index sets* $\mathfrak{V}_k$*,* $\mathfrak{V}_l$*, a linkage method is a function* $f(d, \mathbf{X}, \mathfrak{V}_k, \mathfrak{V}_l)$ *which gives a measure of distance between the clusters defined by* $\mathfrak{V}_k$ *and* $\mathfrak{V}_l$*.*

For agglomerative clustering the two clusters which minimise f are combined. For divisive clustering, for each current cluster we consider all possible splits into two clusters[5], taking the split that maximises f as the optimal split for that particular cluster. The actual split taken then comes from the maximal optimal split across current clusters.

[5]Letting n be the number of observations currently assigned to the cluster in question, there are $2^{n-1} - 1$ unique possible ways to split the cluster into two non-empty separate clusters.

Common linkage methods include

$$\text{Single Linkage [18]} : f(d, \mathbf{X}, \mathfrak{V}_k, \mathfrak{V}_l) = \min_{i \in \mathfrak{V}_k, j \in \mathfrak{V}_l} \{d(\mathbf{x}_i, \mathbf{x}_j)\} \tag{2.2}$$

$$\text{Complete Linkage [19]} : f(d, \mathbf{X}, \mathfrak{V}_k, \mathfrak{V}_l) = \max_{i \in \mathfrak{V}_k, j \in \mathfrak{V}_l} \{d(\mathbf{x}_i, \mathbf{x}_j)\} \tag{2.3}$$

$$\text{Average Linkage [20]} : f(d, \mathbf{X}, \mathfrak{V}_k, \mathfrak{V}_l) = \frac{1}{|\mathfrak{V}_k| \times |\mathfrak{V}_l|} \sum_{i \in \mathfrak{V}_k} \sum_{j \in \mathfrak{V}_l} d(\mathbf{x}_i, \mathbf{x}_j) \tag{2.4}$$

with the choice of linkage method affecting the topology of the resulting clusters. For example complete linkage has a tendency to produce compact clusters (i.e. for a given cluster all observations in that cluster are close to each other with regards to the distance metric chosen) whereas single linkage produces chained clusters which tend not to be compact but can be highly non-linear (i.e. observations in single linkage clusters tend to be 'close' to only a small number of other observations in their cluster, creating a chain-like structure; this structure does not have any properties such as linear separability enforced[6]). Algorithm 27 details the general procedure and can be found in appendix A.2.

Similar to K-means, there are significant disadvantages to this method. The predominant issue is paradoxically the main advantage hierarchical clustering has over K-means: the flexibility in cluster topology through specification of the linkage method. The problem with this choice is that it requires the user to have knowledge on the shape of the true clusters a priori. For example, if the true clusters are compact then selecting a single linkage algorithm will likely result in a misleading output. One may try to infer the topology of the clusters through data exploration, however, this becomes more challenging as the dimensionality of the data increases. Furthermore, it is important to note that even when a suitable linkage method has been established, the resulting clusters for a fixed K is effected by the choice of agglomerative clustering or divisive clustering, with divisive detecting more global

[6]Note that single linkage is the only method mentioned that does not take into consideration other between-cluster distances. Complete linkage takes into consideration other distances by taking the maximum (consider observation indices $a, i \in \mathcal{V}_k$ and $b, j \in \mathcal{V}_l$, if we combined the clusters defined by $\mathcal{V}_k$ and $\mathcal{V}_l$ based on $d(\mathbf{x}_i, \mathbf{x}_j)$, then because we are stating that the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is sufficiently small we also by definition state that the distance between $\mathbf{x}_a$ and $\mathbf{x}_b$ is sufficiently small as well), as does average linkage by taking the mean.

patterns of clustering within the data and agglomerative detecting local clustering patterns. Given that these choices are in addition to selecting K (which has been shown in section 2.1.1 to be a non-trivial task), selection of an inappropriate hierarchical clustering method is highly plausible.

An increase in dimensionality also presents issues with visualisation. As hierarchical clustering only requires the distances between observations, there is no dimension reduction mechanism attached to the algorithm, resulting in no clear way of visualising the data in a lower dimension whilst still maintaining the sense of connected clusters. Nevertheless, whilst the required specification of a linkage method reduces the ease of use with this algorithm, the hard clustering output does still lend some interest with stakeholders due to the interpretability of the resulting cohorts.

2.1.3 Sequential Predictive Modelling

The necessity for unsupervised clustering without any further modelling has limited use with regards to the context studied in this thesis. Whilst interpretable cohort information is indeed useful when paired with a predictive model, obtaining information just about groups of observations with similar characteristics does not allow for intervention aimed at preventing or achieving a desired outcome. Indeed, covariate data is typically provided with an associated response in which the task is to predict the response given the covariates. Despite this, unsupervised methods are still popular in real-world settings, due to the aforementioned interpretability properties, and so prediction is accommodated through sequential predictive modelling [15].

Sequential Predictive Modelling can be viewed as a greedy two-stage model. First clusters are obtained through an unsupervised method, and then that cluster information is utilised in a second-stage predictive model (which is supervised but does not attempt to cluster the observations further). The specifics of how the cluster information is used depend on the type of clustering that the unsupervised method produced. If the output is a hard clustering of the observations (i.e. each observation is assigned to one and only one cluster), such as K-means or hierarchical clustering, then typically we would build K predictive models for each of the K outputted clusters (i.e. M_k for $k = 1, \ldots, K$). Observations whose associated index is an

element of $\mathfrak{V}_k$, $k \in \{1, \ldots, K\}$, would be viewed as training data for model $\mathcal{M}_k$. For prediction of a response given a new observation, $\mathbf{x}_{n+1}$, the new observation would first be assigned to a cluster $\mathfrak{V}_k$ through a procedure incorporating the aspects of the unsupervised method[7], and then the response for $\mathbf{x}_{n+1}$ would be predicted soley by $\mathcal{M}_k$. If the output is a soft clustering of the observations (i.e. each observation is assigned to every cluster with a certain probability) then there are numerous ways in which cluster information could be incorporated. One example is to produce one predictive model with cluster assignment probabilities included as covariates. Another would be to produce K models ($\mathcal{M}_k$ for $k = 1, \ldots, K$), with model $\mathcal{M}_k$ utilising all observations. However, observations for this model are weighted, with the weight for a particular observation being the probability that observation belongs to cluster k, given by the soft clustering (i.e. a soft clustering output gives matrix P with elements $p_{ik} = \pi(i \in \mathfrak{V}_k \mid \mathbf{X})$ for $i = 1, \ldots, n$, $k = 1, \ldots, K$. Model $\mathcal{M}_k$ can then be defined by the likelihood $\pi(\mathbf{y} \mid \mathbf{X}, P_{\cdot k})$). For a new observation $\mathbf{x}_{n+1}$ the methods of prediction again can vary, but typically will require the unsupervised method to produce probabilities of $\mathbf{x}_{n+1}$ belonging to each cluster (i.e. $\pi(n+1 \in \mathfrak{V}_k \mid \mathbf{x}_{n+1}, \mathbf{X})$). For a one-model system one could use this information along with $\mathbf{x}_{n+1}$ to obtain a predicted response. Alternatively, with the probabilities of cluster assignment for $\mathbf{x}_{n+1}$ one could use a weighted average from each of the models predicted responses to obtain an overall predicted response, i.e. a prediction from the model $\sum_{k=1}^{K} \pi(n+1 \in \mathfrak{V}_k \mid \mathbf{x}_{n+1}, \mathbf{X}) \mathbb{E}_{\pi(Y \mid \mathbf{X}, P_{\cdot k})}(Y \mid \mathbf{x}_{n+1})$.

This framework for constructing predictive models sequentially, sometimes referred to as 'cluster-then-predict' [15], has gained popularity through the strong preference for interpretable clusters; however, the greedy methodology only produces a suitable output when the clustering structure in X is synonymous with the clustering structure in $Y \mid X$. Consider data generated from two Gaussian distributions that are well separated in covariate space as showcased in figure 2.1. Here in the first stage an unsupervised method such as K-means would be able to successfully distinguish the two clusters in covariate space. It is also a possibility that

[7]For K-means this assignment would be to the nearest centroid and for hierarchical clustering this would be $\arg\min_{\mathfrak{V}_k \in \mathfrak{V}} \{f(d, \mathbf{X}, \{n+1\}, \mathfrak{V}_k)\}$.

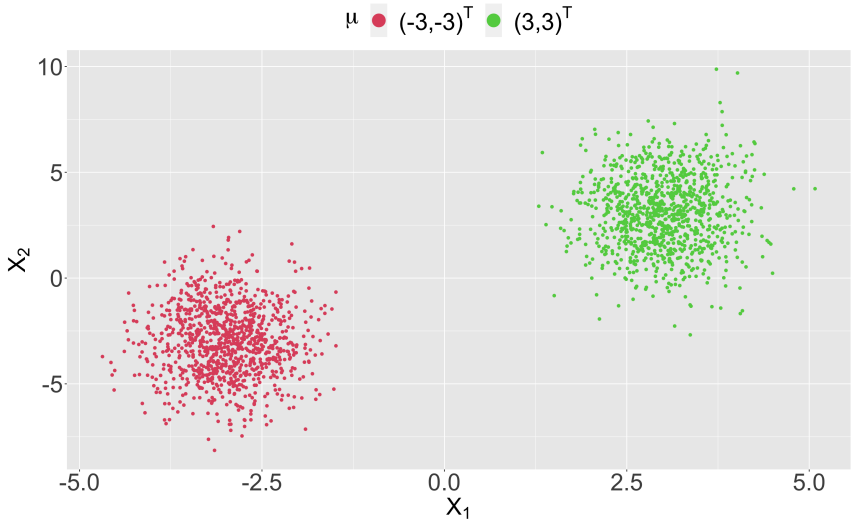


Figure 2.1: Covariate data generated from $\mathcal{N}_2((-3,-3)^T, \mathcal{I}_2)$ (red points) and $\mathcal{N}_2((3,3)^T, \mathcal{I}_2)$ (green points).

when a response is introduced the true clustering for $Y \mid X$ matches that of the clustering structure apparent in X. In more detail, letting $\mathfrak{V}_1$ represent the observations indices in the red cluster and $\mathfrak{V}_2$ represent the observations indices in the green cluster, the introduction of $\mathbf{y}$ could have the property that (assuming a binary response)

$$\mathbb{P}(y_i = 1) = \frac{1}{1 + \exp\{-\beta_0 - x_{i1}\beta_1 \mathbb{1}\{i \in \mathfrak{V}_1\} - x_{i2}\beta_2 \mathbb{1}\{i \in \mathfrak{V}_2\}\}} \tag{2.5}$$

such that the response does not depend on X_2 in the red cluster and does not depend on X_1 in the green cluster. In this scenario, a cluster-then-predict methodology would produce reasonable results. However, it also remains a possibility that introduction of a response produces a clustering structure in $Y \mid X$ at odds with the clustering structure in X. For example, assume $\mathbf{y}$ now has the property

$$\mathbb{P}(y_i = 1) = \frac{1}{1 + \exp\{-\beta_0 - x_{i2}\beta_1 \mathbb{1}\{x_{i1} > x_{i2}\} - x_{i2}\beta_2 \mathbb{1}\{x_{i1} \leq x_{i2}\}\}} \tag{2.6}$$

such that now the response never explicitly depends on X_1, and depends on X_2 differently depending on whether $X_1 > X_2$. This example is visualised in figure 2.2, assuming logistic regression models were built on these two pre-defined clusters (in a similar procedure to SPARRA). Furthermore, assume that new observations are

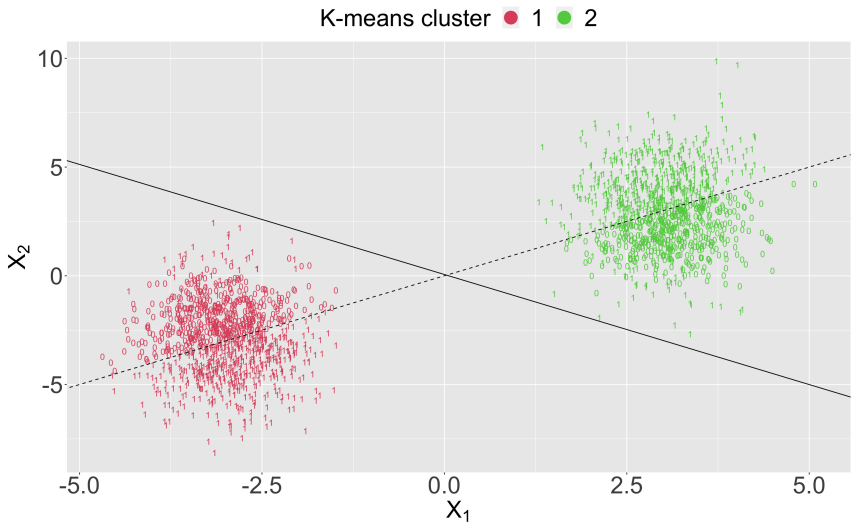


Figure 2.2: Covariate data from figure 2.1 with response values added as labels, along with; K-means separating hyperplane (solid black line) and true separating hyperplane for $Y \mid X$ (dashed black line).

assigned to the cluster associated with their closest K-means centroid, and then have their response predicted from their cluster's associated logistic regression model. In this scenario, the non-synonymous clustering structure results in models which will perform poorly as they fail to capture the relationship between the response and the covariates. Indeed, given the clustering structure provided by K-means, a linear combination of covariates and regression coefficients does not appear to be a suitable modelling assumption. In reality, the linearity assumption is perfectly valid for the true clustering, and a model that detects the true separating line $X_1 = X_2$ will have a high predictive performance with logistic regression models. Perrakis et al. [21] highlight this sequential modelling problem, with their solution being to introduce a latent cluster allocation variable to a joint model of X and Y.

2.1.4 Summary

Through examination of some of the most widely used unsupervised methods it is clear that the ease of use combined with interpretable outputs are key factors in the popularity of these methods with stakeholders. Indeed, for determining cohort information basic unsupervised procedures have highly desirable qualities for a non-statistician audience, even if the implicit cluster constraints are less desirable.

This popularity, however, unfortunately transitions into sequential predictive modelling when prediction of a response is required, which is a cause for concern given that it is unknown whether the cluster structure for $Y \mid X$ (the clustering structure most useful for prediction and therefore the most useful to stakeholders) is synonymous with the clustering structure in X.

2.2 Supervised Clustering

In the supervised setting, as well as covariate data we also have access to an associated response, $\mathbf{y}$. We assume from this point on that we have a binary response, i.e. $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$[8].

Typically the inclusion of a response indicates the requirement of a predictive model from the stakeholder, which can affect the clustering procedure in two ways. Either the clustering aspect of the model is used simply as a tool to provide a more accurate model or the clustering of observations is required as an output alongside the predictive model. Our main concern is with the latter, however, understanding of the former is important in order to review the benefits of such models.

Supervised methods naturally focus primarily on predictive modelling, and as such have incorporated methods for predicting the response of new data. However, the cluster assignment for new data is much more model-dependent and can even be removed from the process. If included, the cluster assignment portion of the methods typically fall under the category of soft clustering. This is due to these methods offering much more uncertainty quantification than the unsupervised methods mentioned previously which manifests itself in a probabilistic or score based interpretation of cluster assignment. Soft clustering is not extremely problematic in terms of interpretability, as a soft clustering assignment can be transformed into a hard clustering assignment through sampling or maximum score attribution (i.e. observation i is assignment to cluster k such that $k = \max_{k \in \{1,\ldots,K\}} \mathbb{P}(i \in \mathfrak{V}_k)$).

In this section we will discuss some popular supervised clustering methods and

[8]In many cases this can be trivially extended to multi-class responses, as well as (with slight alterations of the methods discussed below) continuous responses.

their attempts to provide an interpretable clustering output. It is worth considering at this point that these models were constructed to predict a response given covariate data and not necessarily to partition the data (and by extension the covariate space) and so a lack of attention on the cohort generation aspect of the model is to be expected.

2.2.1 Finite Mixtures of Logistic Regressions

Finite Mixtures of Logistic Regression (FMLR) models are a by-product of mixture modelling practices that date back over many years, with some of the earliest work being a representation of covariate data as a mixture of Gaussian distributions by Pearson [22].

Noting that a logistic regression model with parameters $\boldsymbol{\beta}$ has likelihood

$$\pi(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}) = \prod_{i=1}^{n} (1 + e^{-\mathbf{x}_i^T \boldsymbol{\beta}})^{-y_i} (1 + e^{\mathbf{x}_i^T \boldsymbol{\beta}})^{-(1-y_i)} \tag{2.7}$$

we can represent a FMLR model as

$$\pi(\mathbf{y} \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\tau}) = \prod_{i=1}^{n} \sum_{l=1}^{K} \tau_l \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l) \tag{2.8}$$

where $\mathcal{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)^T$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K)^T$ and τ_l represents the contribution of sub-model l to the overall model and has the condition that $\sum_{l=1}^{K} \tau_l = 1$. This representation, however, does not allow for the estimation of the model parameters as the maximum likelihood equations produced are intractable. Therefore we introduce unknown latent vectors, $V_1, \ldots, V_n$, which indicate cluster assignment — $V_i = (V_{i1}, \ldots, V_{iK})$ where $V_{ik} \in \{0, 1\}$. If V was known, we could express this as a binary $n \times K$ matrix, $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)^T$, where $v_{ik} = 1$ indicates observation i belongs to cluster k. Assuming $\mathbf{v}_i \sim \text{Mult}(1, \boldsymbol{\tau})$, this would give the following

likelihood:

$$\begin{aligned}\pi(\mathbf{y}, \mathbf{v}_1, \ldots, \mathbf{v}_n \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\tau}) &= \pi(\mathbf{y} \mid \mathbf{v}_1, \ldots, \mathbf{v}_n, \mathbf{X}, \mathcal{B}, \boldsymbol{\tau})\pi(\mathbf{v}_1, \ldots, \mathbf{v}_n \mid \boldsymbol{\tau}) \\ &= \prod_{i=1}^{n}\prod_{k=1}^{K}[\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)]^{v_{ik}} \times \prod_{i=1}^{n}\prod_{k=1}^{K}[\tau_k]^{v_{ik}} \\ &= \prod_{i=1}^{n}\prod_{k=1}^{K}[\tau_k\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)]^{v_{ik}} \end{aligned} \tag{2.9}$$

Reverting back to setting where $V_1, \ldots, V_n$ are unknown, through a combination of equations (2.8) and (2.9) we can obtain the conditional distribution for $V_1, \ldots, V_n$:

$$\begin{aligned}\pi(V_1, \ldots, V_n \mid \mathbf{y}, \mathbf{X}, \mathcal{B}, \boldsymbol{\tau}) &= \frac{\pi(\mathbf{y}, \mathbf{v}_1, \ldots, \mathbf{v}_n \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\tau})}{\pi(\mathbf{y} \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\tau})} \\ &= \prod_{i=1}^{n}\prod_{k=1}^{K}\left[\frac{\tau_k\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)}{\sum_{l=1}^{K}\tau_l\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l)}\right]^{V_{ik}} \end{aligned} \tag{2.10}$$

noting that

$$\sum_{l=1}^{K}\tau_l\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l) = \prod_{k=1}^{K}\left[\sum_{l=1}^{K}\tau_l\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l)\right]^{v_{ik}} \tag{2.11}$$

as $v_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} v_{ik} = 1 \; \forall \; i$.

With this representation, we can then proceed to apply the Expectation–Maximisation (EM) algorithm [23, 24] in order to obtain model parameter estimates. For the expectation step we have, for iteration t,

$$\begin{aligned}Q((\boldsymbol{\tau}, \mathcal{B}) \mid (\boldsymbol{\tau}^{(t)}, \mathcal{B}^{(t)})) &= \mathbb{E}_{\pi(V_1,\ldots,V_n|\mathbf{y},\mathbf{X},\mathcal{B}^{(t)},\boldsymbol{\tau}^{(t)})}\{\log(\pi(\mathbf{y}, \mathbf{v}_1, \ldots, \mathbf{v}_n \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\tau}))\} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}_{\pi(V_{ik}|\mathbf{y},\mathbf{X},\mathcal{B}^{(t)},\boldsymbol{\tau}^{(t)})}(V_{ik})\left[\log(\tau_k\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k))\right] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\frac{\tau_k^{(t)}\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k^{(t)})}{\sum_{l=1}^{K}\tau_l^{(t)}\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l^{(t)})}\left[\log(\tau_k\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k))\right] \end{aligned} \tag{2.12}$$

Letting

$$v_{ik}^{(t)} = \frac{\tau_k^{(t)}\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k^{(t)})}{\sum_{l=1}^{K}\tau_l^{(t)}\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l^{(t)})} \tag{2.13}$$

for the maximisation step we must obtain

$$\begin{aligned}
\boldsymbol{\tau}^{(t+1)} &= \arg\max_{\boldsymbol{\tau}} Q((\boldsymbol{\tau}, \mathcal{B}) \mid (\boldsymbol{\tau}^{(t)}, \mathcal{B}^{(t)})) \\
&= \arg\max_{\boldsymbol{\tau}} \sum_{k=1}^{K} \log(\tau_k) \sum_{i=1}^{n} v_{ik}^{(t)} \quad \left(\text{subject to } \sum_{k=1}^{K} \tau_k = 1 \right) \qquad (2.14) \\
\boldsymbol{\beta}_k^{(t+1)} &= \arg\max_{\boldsymbol{\beta}_k} Q((\boldsymbol{\tau}, \mathcal{B}) \mid (\boldsymbol{\tau}^{(t)}, \mathcal{B}^{(t)})) \\
&= \arg\max_{\boldsymbol{\beta}_k} \sum_{i=1}^{n} v_{ik}^{(t)} \log(\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)) \qquad (2.15)
\end{aligned}$$

Equation (2.14) is equivalent to the maximum likelihood estimator of a multinomial distribution, leading to

$$\tau_k^{(t+1)} = \frac{\sum_{i=1}^{n} v_{ik}^{(t)}}{n} \qquad (2.16)$$

Equation (2.15) is equivalent to obtaining the coefficients of a weighted logistic regression, and so methods such as Iteratively Reweighted Least Squares (IRLS) [25] can be applied to obtain a solution.

Use of the EM method results in algorithm 1, which is initialised with a random start for $\boldsymbol{\tau}$, $\mathcal{B}$ and $\mathbf{V}$ (noting that both $\boldsymbol{\tau}$ and the rows of $\mathbf{V}$ must still sum to 1). Our stopping criterion, or convergence indicator, is represented by η and is typically a small value.

In terms of prediction, for a new observation $\mathbf{x}_{n+1}$, the expected response would be a weighted sum of the probability of success for $\mathbf{x}_{n+1}$ under each of the K individual logistic regression models (i.e. $\hat{y}_{n+1} = \sum_{k=1}^{K} \hat{\tau}_k (1 + e^{-\mathbf{x}_{n+1}^T \hat{\boldsymbol{\beta}}_k})^{-1}$ where $\hat{\boldsymbol{\tau}} = (\hat{\tau}_1, \ldots, \hat{\tau}_K)^T$ and $\hat{\mathcal{B}} = (\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_K)^T$ are outputs from algorithm 1). The expected clustering assignment, $\mathbf{v}_{n+1}$, can also obtained through evaluation of the marginal distribution

$$\pi(\mathbf{v}_{n+1} \mid \mathbf{x}_{n+1}, \hat{\mathcal{B}}, \hat{\boldsymbol{\tau}}) = \sum_{i=0}^{1} \prod_{k=1}^{K} \left[\hat{\tau}_k \pi(y_{n+1} = i \mid \mathbf{x}_{n+1}, \hat{\boldsymbol{\beta}}_k) \right]^{v_{n+1,k}} \qquad (2.17)$$

leading to individual elements predicted as

$$\hat{\mathbb{P}}(V_{n+1,k} = 1 \mid \mathbf{x}_{n+1}, \hat{\mathcal{B}}, \hat{\boldsymbol{\tau}}) = \hat{\tau}_k \sum_{i=0}^{1} \pi(y_{n+1} = i \mid \mathbf{x}_{n+1}, \hat{\boldsymbol{\beta}}_k) = \hat{\tau}_k \qquad (2.18)$$

Algorithm 1: EM algorithm for FMLR models

Input : *Covariate Matrix* — $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$,
Response Vector — $\mathbf{y} = (y_1, \ldots, y_n)^T$, *Convergence Threshold* — $\eta > 0$,
Proportion Vector — $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K)^T$,
Regression Coefficent Matrix — $\mathcal{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)^T$,
Latent Cluster Assignment Matrix — $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)^T$

Step 1 : Let $\tilde{\mathbf{V}}$ be a matrix with elements

$$\tilde{v}_{ik} = \frac{\tau_k \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)}{\sum_{l=1}^{K} \tau_l \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l)}$$

Step 2 : Let $\tilde{\boldsymbol{\tau}}$ such that

$$\tilde{\tau}_k = \frac{\sum_{i=1}^{n} \tilde{v}_{ik}}{n}$$

Step 3 : **for** $k = 1, \ldots, K$ **do**
  Obtain $\tilde{\boldsymbol{\beta}}_k$ through IRLS with log likelihood $\sum_{i=1}^{n} \tilde{v}_{ik} \log(\pi(y_i \mid \mathbf{x}_i, \tilde{\boldsymbol{\beta}}_k))$
end
Let $\tilde{\mathcal{B}} = (\tilde{\boldsymbol{\beta}}_1, \ldots, \tilde{\boldsymbol{\beta}}_K)^T$.
Step 4 : **if** $\log(\pi(\mathbf{y} \mid \mathbf{X}, \tilde{\mathcal{B}}, \tilde{\boldsymbol{\tau}})) - \log(\pi(\mathbf{y} \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\tau})) < \eta$ **then**
  Let $\mathbf{V} = \tilde{\mathbf{V}}$, $\boldsymbol{\tau} = \tilde{\boldsymbol{\tau}}$ and $\mathcal{B} = \tilde{\mathcal{B}}$. Stop.
else
  Let $\mathbf{V} = \tilde{\mathbf{V}}$, $\boldsymbol{\tau} = \tilde{\boldsymbol{\tau}}$ and $\mathcal{B} = \tilde{\mathcal{B}}$. Go to step 1.
end
Result : $\mathbf{V}, \boldsymbol{\tau}, \mathcal{B}$

From the method showcased in algorithm 1 it is clear that the clustering of observations is merely used as a tool for a more flexible model. Whilst it is indeed possible to obtain a soft clustering estimate of new observations through equation (2.18), and this in turn can produce a hard clustering assignment, the use of that cluster assignment is meaningless as the estimated probability vector of belonging to clusters $1, \ldots, K$ for *any* new observation is always $\hat{\boldsymbol{\tau}}$. In addition to this, the response prediction does not utilise any hard or soft cluster assignment in its calculation of $\hat{y}$.

From an interpretability perspective, for the training data the response is known and so using the conditional distribution $\pi(\mathbf{v}_i \mid \mathbf{x}_i, y_i, \hat{\mathcal{B}}, \hat{\boldsymbol{\tau}})$ gives a soft cluster assignment of the observations which is dependent on the covariate data, i.e.

$$\hat{\mathbb{P}}(V_{ik} = 1 \mid \mathbf{x}_i, y_i, \hat{\mathcal{B}}, \hat{\boldsymbol{\tau}}) = \frac{\tau_k \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)}{\sum_{l=1}^{K} \tau_l \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l)} \quad \text{for } i = 1, \ldots, n \tag{2.19}$$

However, in this case the treatment of the cluster assignment offers too much flexibility to provide a clear visual interpretation of the cohorts formed. This is in stark contrast to unsupervised methods such as K-means, which produced regions of the covariate space too restrictive in their topological constraints (see section 2.1.1). Indeed, here we have the opposite problem, there are no restrictions on the cohorts formed from a hard clustering assignment of the data and as a result the cohorts typically appear disconnected and lacking similarity in covariate space. An example of this can be seen in figure 2.3, where even with synonymous clustering structure in X and $Y \mid X$ the hard clustering output of a FMLR model cannot provide a clear separation of the two cohorts in covariate space[9]. We can attribute this to the

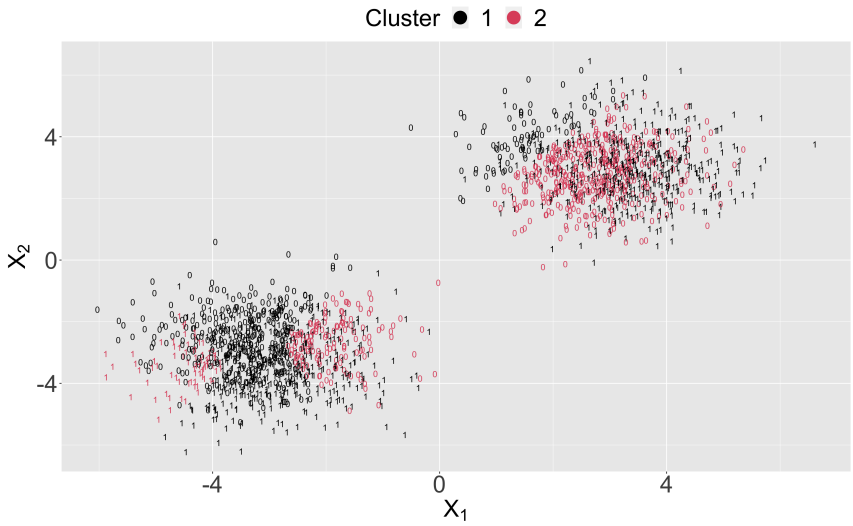


Figure 2.3: FMLR clustering assignment output for data generated from two Gaussian's, Gaussian 1: $\mathcal{N}((-3,-3)^T, \mathcal{I}_2)$ and Gaussian 2: $\mathcal{N}((3,3)^T, \mathcal{I}_2)$, each with a differing relationship to the associated response (i.e. $\boldsymbol{\beta}_1 = (3,0,1)^T$ and $\boldsymbol{\beta}_2 = (3,-1,0)^T$). Observation labels are their associated response and observation colours relate to their assigned cluster.

fact that the cluster assignment variable was introduced to provide a method for obtaining the model coefficients and therefore the output from this latent variable was not intended to be of use.

In summary, the stakeholder's requirement for a cohort producing mechanism explains the apparent lack of popularity in applications. In addition to this there is

[9]Here observations were given a hard clustering assignment by selecting cluster k for observation i if $k = \arg\max_{k=1,\dots,K}\{\mathbb{P}(v_{ik} = 1 \mid \mathbf{x}_i, y_i, \hat{\mathcal{B}}, \hat{\boldsymbol{\tau}})\}$.

the common issue of requiring the number of clusters (or mixture components) K to be known a priori[10]. Nevertheless, there is still the clear advantage here that the predictive power of this model in general will be greater than unsupervised methods, due to the inclusion of the response.

2.2.2 Mixture of Experts

First introduced in 1991 [6], Mixture of Experts (MoE) models provide a supervised clustering method that gives more consideration to the assignment of covariates to a cluster (or expert) than FMLR models. MoE models are intrinsically linked to FMLR models when the experts are logistic regression models, as we can represent a MoE model as

$$\pi(\mathbf{y} \mid \mathbf{X}, \mathcal{B}, \mathbf{\Lambda}) = \prod_{i=1}^{n} \sum_{k=1}^{K} g_k(\mathbf{x}_i \mid \mathbf{\Lambda}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k) \tag{2.20}$$

where $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K)^T$ and

$$g_k(\mathbf{x} \mid \mathbf{\Lambda}) = \frac{e^{\mathbf{x}^T \boldsymbol{\lambda}_k}}{\sum_{l=1}^{K} e^{\mathbf{x}^T \boldsymbol{\lambda}_l}} \tag{2.21}$$

Comparison of equation (2.8) to equation (2.20) highlights the similarity of the two approaches, with both models having the same structure. The difference in the two models is crucial, however, as instead of a general model proportion parameter τ_k we have a covariate dependent softmax function with unknown parameters.

Definition 2.2.1 (Softmax function)**.** *Given a real-valued vector* $\mathbf{x}$*, a vector function* $\mathbf{g} : \mathbb{R}^K \to (0,1)^K$ *is said to be a softmax function if* $\sum_{k=1}^{K} g_k(\mathbf{x}) = 1$*.*

Specification of $\mathbf{g}$, also known as the gating network, then allows for a meaningful soft clustering assignment of the observations, as given parameters $\mathbf{\Lambda}$ any observation can be given a score $g_k(\mathbf{x} \mid \mathbf{\Lambda})$ for belonging in cluster k.

Estimation of the model parameters mirrors the estimation procedure for FMLR models, where we introduce latent variables $V_1, \ldots, V_n$. The caveat here is that now

[10] Selection of K is further discussed in section 3.2.

$V_i \sim \text{Mult}(1, \mathbf{g}(\mathbf{x}_i \mid \boldsymbol{\Lambda}))$, which gives the following:

$$\pi(\mathbf{y}, \mathbf{v}_1, \ldots, \mathbf{v}_n \mid \mathbf{X}, \mathcal{B}, \boldsymbol{\Lambda}) = \prod_{i=1}^{n} \prod_{k=1}^{K} [g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)]^{v_{ik}} \tag{2.22}$$

$$\pi(V_1, \ldots, V_n \mid \mathbf{y}, \mathbf{X}, \mathcal{B}, \boldsymbol{\Lambda}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left[\frac{g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)}{\sum_{l=1}^{K} g_l(\mathbf{x}_i \mid \boldsymbol{\Lambda}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l)} \right]^{V_{ik}} \tag{2.23}$$

For iteration t this then gives our expectation step as

$$Q((\boldsymbol{\Lambda}, \mathcal{B}) \mid (\boldsymbol{\Lambda}^{(t)}, \mathcal{B}^{(t)})) = \sum_{i=1}^{n} \sum_{k=1}^{K} v_{ik}^{(t)} \log(g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)) \tag{2.24}$$

where

$$v_{ik}^{(t)} = \frac{g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}^{(t)}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k^{(t)})}{\sum_{l=1}^{K} g_l(\mathbf{x}_i \mid \boldsymbol{\Lambda}^{(t)}) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l^{(t)})} \tag{2.25}$$

The maximisation step as before can be done separately for the parameters $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$, whereas $\boldsymbol{\Lambda}$ requires joint estimation:

$$\begin{aligned}
\boldsymbol{\Lambda}^{(t+1)} &= \arg\max_{\boldsymbol{\Lambda}} Q((\boldsymbol{\Lambda}, \mathcal{B}) \mid (\boldsymbol{\Lambda}^{(t)}, \mathcal{B}^{(t)})) \\
&= \arg\max_{\boldsymbol{\Lambda}} \sum_{i=1}^{n} \sum_{k=1}^{K} v_{ik}^{(t)} \log(g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda})) \\
&= \arg\max_{\boldsymbol{\Lambda}} \sum_{i=1}^{n} \left[\left(\sum_{k=1}^{K} v_{ik}^{(t)} \mathbf{x}_i^T \boldsymbol{\lambda}_k \right) - \log \left(\sum_{l=1}^{K} e^{\mathbf{x}_i^T \boldsymbol{\lambda}_l} \right) \right] \qquad (2.26) \\
\boldsymbol{\beta}_k^{(t+1)} &= \arg\max_{\boldsymbol{\beta}_k} Q((\boldsymbol{\Lambda}, \mathcal{B}) \mid (\boldsymbol{\Lambda}^{(t)}, \mathcal{B}^{(t)})) \\
&= \arg\max_{\boldsymbol{\beta}_k} \sum_{i=1}^{n} v_{ik}^{(t)} \log(\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)) \qquad (2.27)
\end{aligned}$$

All parameters can be estimated through the IRLS algorithm [26]. The EM algorithm therefore for MoE models is similar to that of algorithm 1, with the only alterations being that $\boldsymbol{\tau}$ is replaced with $\mathbf{g}(\mathbf{x} \mid \boldsymbol{\Lambda})$ and step 2 is replaced with a IRLS algorithm for obtaining $\tilde{\boldsymbol{\Lambda}}$.

With the latent variables $V_1, \ldots, V_n$ in this setting now dependent on the covariates, it is unsurprising that when considering prediction of cluster assignment for

$\mathbf{x}_{n+1}$ we have

$$\hat{\mathbb{P}}(V_{n+1,k} = 1) = g_k(\mathbf{x}_{n+1} \mid \hat{\mathbf{\Lambda}}) \tag{2.28}$$

This method therefore offers a significant improvement over previous methods for a stakeholder whose aims are cohort detection combined with predictive modelling. Clearly the MoE model targets accurate prediction of the response through the individual 'experts'[11], but the model also provides cohort detection through the gating network $\mathbf{g}(\mathbf{x} \mid \mathbf{\Lambda})$.

Whilst the gating network provides a soft clustering, we can obtain a hard clustering for observation $n+1$ by assigning it to cluster $k = \arg\max_{l=1,\dots,K}\{g_k(\mathbf{x}_{n+1} \mid \hat{\mathbf{\Lambda}})\}$. This method creates $\binom{K}{2}$ linear decision boundaries for the experts, namely:

$$\{\mathbf{x} : g_k(\mathbf{x} \mid \hat{\mathbf{\Lambda}}) = g_l(\mathbf{x} \mid \hat{\mathbf{\Lambda}})\} \quad \text{for } k, l \in \{1, \dots, K\},\ k \neq l \tag{2.29}$$

Intersection of these boundaries then define connected regions of the covariate space similar to that of a Voronoi diagram produced by K-means clustering (see appendix A.1). This aspect of MoE models is highly desirable, as we have a mechanism for generating cohorts which includes the response whilst additionally defining regions of the covariate space such that observations with very dissimilar attributes are not likely to be in the same cohort. Referring back to the example showcased in figure 2.3, a two-cluster MoE model can produce a meaningful clustering that also considers the response for this dataset, as shown in figure 2.4.

The beneficial dual output of MoE models has perhaps led to a higher level of application [27, 28] when compared to FMLR models, however, there are still interpretability issues with this method. The main issue being that the gating network restricts the decision boundary between clusters to be linear, giving linearly separable clusters which may not accurately reflect the true clustering structure[12]. Of course there is more flexibility to this method as one could specify another non-linear softmax function $\mathbf{g}$, however, the specific structure of the data that would aid

[11] For observation i expert k would be the model expressed as $\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_k)$.

[12] The example given in figure 2.4 performs so well precisely because a linear decision boundary can separate the true clusters.

in the specification of $\mathbf{g}$ is difficult to obtain in high dimensions.

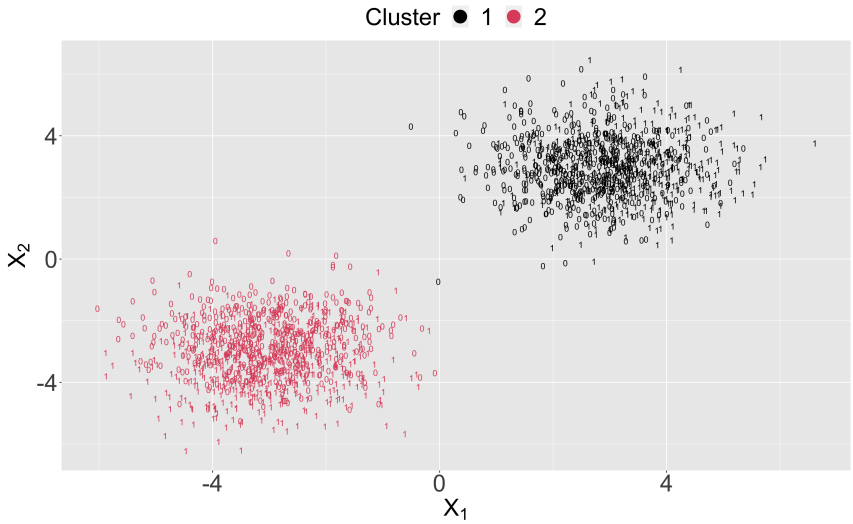


Figure 2.4: MoE clustering assignment output for data generated from two Gaussian's, Gaussian 1: $\mathcal{N}((-3,-3)^T, \mathcal{I}_2)$ and Gaussian 2: $\mathcal{N}((3,3)^T, \mathcal{I}_2)$, each with a differing relationship to the associated response (i.e. $\boldsymbol{\beta}_1 = (3,0,1)^T$ and $\boldsymbol{\beta}_2 = (3,-1,0)^T$). Observation labels are their associated response and observation colours relate to their assigned cluster.

2.2.2.1 Hierarchical Mixture of Experts

If we suspect a non-linear boundary but do not wish to alter the form of $\mathbf{g}$, one alternative is to approximate said boundary through multiple piecewise linear boundaries. This can be achieved through a hierarchical structure of experts being employed within the MoE model [29]. For example, a two-level MoE model would have the form:

$$\begin{aligned}\pi(\mathbf{y} \mid \mathbf{X}, \mathcal{B}^\dagger, \boldsymbol{\Lambda}, &\boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^{|\mathbf{K}|}) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{|\mathbf{K}|} g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}) \sum_{l=1}^{K_k} g_l^k(\mathbf{x}_i \mid \boldsymbol{\Lambda}^k) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l^k) \end{aligned} \tag{2.30}$$

where: $\mathbf{K}$ is now a vector of second-level cluster sizes, $|\mathbf{K}|$ is the number of elements in $\mathbf{K}$, $\mathcal{B}^\dagger = \{\mathcal{B}^1, \ldots, \mathcal{B}^{|\mathbf{K}|}\}$ with $\mathcal{B}^k = (\boldsymbol{\beta}_1^k, \ldots, \boldsymbol{\beta}_{K_k}^k)^T$ being the regression coefficient matrix for the K_k second-level clusters associated with a first-level cluster, $\boldsymbol{\Lambda}$ represents the softmax coefficient matrix for the first level, $\{\boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^{|\mathbf{K}|}\}$ represents the $|\mathbf{K}|$ softmax coefficient matrices for the second-level clusters and $\mathbf{g}, \mathbf{g}^1, \ldots, \mathbf{g}^{|\mathbf{K}|}$

are softmax functions as defined in equation (2.21). A two-level model will contain $\sum_{k=1}^{|\mathbf{K}|} K_k$ experts and therefore $\sum_{k=1}^{|\mathbf{K}|} K_k$ clusters.

Regarding the partitioning of the covariate space one can obtain with a MoE model, introduction of a second level provides further flexibility. At the top level, creation of linear boundaries with $\mathbf{g}$ is as before. However, now each of these $|\mathbf{K}|$ regions (defined by the intersection of these boundaries) can be further partitioned through the intersection of linear boundaries defined through $\mathbf{g}^k$. This partitioning method bears resemblance to the partitioning of the covariate space achieved through basic decision trees [26, 30], albeit with a more complicated modelling framework and less restricted region topology (regions formed by a decision tree are hyper-rectangles).

Introduction of latent cluster allocation vectors $V_1, \ldots, V_n$ also requires reformatting, as

$$V_i = (V_{i11}, \ldots, V_{i1K_1}, \ldots, V_{i|\mathbf{K}|1}, \ldots, V_{i|\mathbf{K}|K_{|\mathbf{K}|}}) \tag{2.31}$$

where $V_{ikl} \in \{0, 1\}$ and $V_{ikl} = 1$ indicates observation i belongs to the l^{th} cluster in cluster k. Letting

$$V_i \sim \text{Mult}(1, \{\mathbf{g}_1(\mathbf{x}_i \mid \boldsymbol{\Lambda})\mathbf{g}^1(\mathbf{x}_i \mid \boldsymbol{\Lambda}^1), \ldots, \mathbf{g}_{|\mathbf{K}|}(\mathbf{x}_i \mid \boldsymbol{\Lambda})\mathbf{g}^{|\mathbf{K}|}(\mathbf{x}_i \mid \boldsymbol{\Lambda}^{|\mathbf{K}|})\}) \tag{2.32}$$

this then gives the following:

$$\begin{aligned}
&\pi(\mathbf{y}, \mathbf{v}_1, \ldots, \mathbf{v}_n \mid \mathbf{X}, \mathcal{B}^\dagger, \boldsymbol{\Lambda}, \boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^{|\mathbf{K}|}) \\
&\quad = \prod_{i=1}^{n} \prod_{k=1}^{|\mathbf{K}|} \prod_{l=1}^{K_k} [g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}) g_l^k(\mathbf{x}_i \mid \boldsymbol{\Lambda}^k) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l^k)]^{v_{ikl}} \qquad (2.33) \\
&\pi(V_1, \ldots, V_n \mid \mathbf{y}, \mathbf{X}, \mathcal{B}^\dagger, \boldsymbol{\Lambda}, \boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^{|\mathbf{K}|}) \\
&\quad = \prod_{i=1}^{n} \prod_{k=1}^{|\mathbf{K}|} \prod_{l=1}^{K_k} \left[\frac{g_k(\mathbf{x}_i \mid \boldsymbol{\Lambda}) g_l^k(\mathbf{x}_i \mid \boldsymbol{\Lambda}^k) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_l^k)}{\sum_{a=1}^{|\mathbf{K}|} g_a(\mathbf{x}_i \mid \boldsymbol{\Lambda}) \sum_{b=1}^{K_a} g_b^a(\mathbf{x}_i \mid \boldsymbol{\Lambda}^a) \pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_b^a)} \right]^{V_{ikl}} \qquad (2.34)
\end{aligned}$$

The EM algorithm then follows these equations in the standard way, with the only difference being that the introduction of a second level requires the maximisation of the parameters $\boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^{|\mathbf{K}|}$, which can be done separately.

To highlight the benefits of a Hierarchical Mixture of Experts (HMoE) model

over the standard MoE model in some situations, consider the following covariate data $\mathbf{X} \in \mathbb{R}^{2000\times 2}$:

$$x_{i1} = \begin{cases} \frac{i-1}{5\times 999} & \text{if } i \in \{1,\dots,1000\} \\ \frac{i-1001}{5\times 999} & \text{if } i \in \{1001,\dots,2000\} \end{cases} \tag{2.35}$$

$$x_{i2} \sim \begin{cases} \mathcal{U}\left(\sin(20x_{i1}) + 0.15, \sin(20x_{i1}) + 1.85\right) & \text{if } i \in \{1,\dots,1000\} \\ \mathcal{U}\left(\sin(20x_{i1}) - 1.85, \sin(20x_{i1}) - 0.15\right) & \text{if } i \in \{1001,\dots,2000\} \end{cases} \tag{2.36}$$

This data produces two curves contained within the hypercube $[0, 0.2] \times [\sin(4) - 1.85, 2.85]$. With this covariate data, we define two clusters, with specification of the regression coefficients and response as follows:

$$\boldsymbol{\beta}_1 = (-1, -8, 1)^T \tag{2.37}$$

$$\boldsymbol{\beta}_2 = (1, 24, 4)^T \tag{2.38}$$

$$y_i \sim \begin{cases} \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_1})^{-1}) & \text{if } i \in \{1,\dots,1000\} \\ \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_2})^{-1}) & \text{if } i \in \{1001,\dots,2000\} \end{cases} \tag{2.39}$$

The two clusters are well separated in covariate space but not linearly separable, and so a standard MoE model will fail to output the true clustering. A HMoE model where $\mathbf{K} = (2, 2)^T$ however allows for the space to be separated into two regions where the two true clusters are linearly separable. This can be seen in figure 2.5. Note that HMoE has produced four clusters, but each of the four clusters only contains observations from a single true cluster, marking an improvement over the standard MoE model.

In summary, MoE models allow for generation of linearly separable regions of the covariate space constructed through a framework which includes the response. Therefore MoE address a lot of key concerns for stakeholders aiming for joint cohort generation and predictive modelling. The restriction of linearly separable regions can result in problematic situations where the true clustering structure is non-linear, however, through a hierarchical tree like structure of experts a non-linear boundary

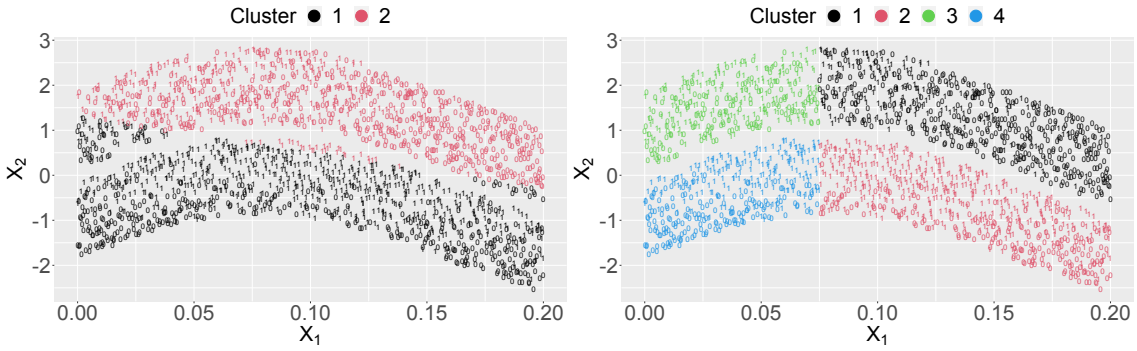


Figure 2.5: MoE (left) and HMoE (right) clustering assignment output for the data detailed in equations (2.35) and (2.36). Observation labels are their associated response and observation colours relate to their assigned cluster.

can be approximated through a piecewise linear boundary. This though produces another issue; the approximation will require the creation of several additional clusters, many of which will be capturing the same relationship between the covariates and the response (as seen in figure 2.5). From an interpretability perspective this becomes problematic, specifically in the scenario of generating cohort-specific intervention plans, as budgetary restrictions typically translate to restrictions on the number of cohorts, which in turn limits the power of a HMoE model. Additionally, increasing the depth of the model will increase computational time along with the common issue of non-automatic selection of the number of clusters (additional computational time will be required to attempt to find the best model).

2.2.3 Summary

From the viewpoint of a stakeholder, the inclusion of a response for covariates should not detract from the goal of producing a model which defines cohorts based on said covariates. Instead, this cohort generation should be given equal priority to the prediction of the response, which leads to ideally a joint procedure. However, from the viewpoint of the statistician, inclusion of a response muddies the waters as to what aspect of clustering observations is most beneficial.

In the unsupervised setting this was clear as we are clustering to produce cohorts which have similar members based on some user-defined measure. In a supervised setting, it is now unclear as to whether the clustering should be used to provide

a more accurate response prediction for new data or to produce cohorts that are clear and can be targeted with unique interventions. This confusion explains the variation in treatment of clustering observations for supervised methods.

A common feature of supervised clustering methods, as seen with FMLRs and MoEs, is to treat the cluster assignment as a latent variable and therefore by definition as unknown, which leads to a direct tackling of the uncertainty that surrounds the clustering procedure. This uncertainty quantification is less apparent in unsupervised methods, where at each iteration of the algorithms for K-means and hierarchical clustering we have certainty in each observation's cluster at that point. Naturally this uncertainty quantification manifests itself as a soft clustering output for many supervised methods. However, expressing uncertainty in cluster-specific model parameters instead of through cluster assignment itself allows for a hard clustering assignment to be fundamental to the clustering algorithm. This form of uncertainty quantification is explored further in chapters 3 and 4.

CHAPTER 3

Bayesian Frameworks & Graphical Representations of Data

With previous methods detailed in chapter 2, this chapter serves as a more direct introduction to chapter 4 by detailing the various tools that the UNCOVER method utilises. Specifically, we introduce and justify the use of a Bayesian setting for cluster modelling. Not only does a Bayesian framework allow for uncertainty quantification of the models parameters, it can be used as a vehicle to introduce expert opinion and to avoid the asymptotic assumptions of frequentist methods. This final point is particularly relevant for the UNCOVER method as dealing with small amounts of data is a necessity.

The second part of this chapter concerns graphical approaches to clustering observations. Mainly operating in unsupervised settings, there exists some well established graphical methods [31] which will be highlighted along with their relevant properties for UNCOVER. In supervised problems the use of graph theory is typically considered only for the parameters of the model, for example in the form of a Directed Acyclic Graph (DAG) [32]. Whilst implicitly DAGs are formed through Bayesian modelling, the graphical aspect in this chapter will be solely focused on creating a graphical representation of the observations, and as such can be viewed temporarily as separate to prediction modelling.

3.1 The Bayesian Paradigm

In all the parametric models detailed thus far, we have used frequentist techniques (such as the expectation–maximisation algorithm) to obtain the parameters for the model. Implicitly the assumption we make when using such methods is that there is a true value for the parameters which can be obtained with a sufficient amount of data. Indeed, without a distributional form for the model parameters, we express a level of certainty in our estimation of the parameters used in the final model.

However, we are always in the finite data setting, with this being exacerbated for clustering methods which partition the training data into smaller observation sets. Specifically we refer to the hard clustering approach here, and the issues surrounding the choice of partition. For example, consider a situation in which the response is binary and a possible partition of the data results in a cluster containing only observations that have the same response. Whether assuming a parametric or non-parametric model for this particular cluster, the result will be typically be degenerate. For a parametric model this would occur when the asymptotic arguments of frequentist modelling break down, giving infinite values to the parameters to ensure a singular response[1]. For non-parametric models, such as decision trees [30], this occurs simply through the optimal model being the assignment of all observations to the single response present. Note that the cluster need not only contain a singular response; if there exists a technique within the frequentist method that completely separates the response then we can arrive at an 'optimal' (in the sense that it has zero error in predicting the response of the clustered data) but unrealistic cluster-level model.

In general there are three common approaches to this particular issue:

1. We allow clusters to share observation information, which allows for varying response types to be present in all cluster-level models.

[1]For example consider a logistic regression model (see equation (2.7)) where the only response for the training data is 1. The optimal estimate for each regression coefficient is $+\infty$ as this ensures that $\hat{\mathbb{P}}(Y = 1 \mid \mathbf{x}) = (1 + e^{-\mathbf{x}^T\hat{\boldsymbol{\beta}}})^{-1} = 1 \; \forall \; \mathbf{x}$.

2. We constrain the cluster-level models to exclude partition selections which cause model degeneracy.

3. We employ a parametric model, but with prior beliefs expressed about the form of the parameters, constraining them in the presence of finite data.

Approach 1 is by definition a soft clustering approach, as using all observations for a cluster-level model would require a weighting of the observations by importance to the cluster to get differing clusters models. This weighting of observations then gives the soft clustering. Approach 2 falls under the umbrella of model mis-specification [33], as we highlight situations in which the model is mis-specified in order to avoid doing so. This then raises the question of detecting when a model will be mis-specified, which is no trivial task. Even in settings where the cluster-level model does not degenerate the particular selection of observations may result in model behaviour that deviates significantly from the user's initial perception of a suitable output. The final approach is the one adopted for the UNCOVER algorithm given in chapter 4; we introduce a Bayesian framework where our prior beliefs on the parameters that govern the model heavily influence our posterior belief on the parameters when confronted with a cluster that contains a small number of observations.

Initially we begin with the frequentist tools for a parametric model, namely the model parameters $\mathbf{\Theta}$ and likelihood $\pi(\mathbf{y}|\mathbf{X}, \mathbf{\Theta})$. In a Bayesian setting, we assume a prior distribution $\pi(\mathbf{\Theta})$ for the parameters $\mathbf{\Theta}$ which can be combined with the likelihood through Bayes theorem to obtain a posterior distribution $\pi(\mathbf{\Theta}|\mathbf{y}, \mathbf{X})$.

$$\pi(\mathbf{\Theta}|\mathbf{y}, \mathbf{X}) = \frac{\pi(\mathbf{y}|\mathbf{X}, \mathbf{\Theta})\pi(\mathbf{\Theta})}{\pi(\mathbf{y}|\mathbf{X})} = \frac{\pi(\mathbf{y}|\mathbf{X}, \mathbf{\Theta})\pi(\mathbf{\Theta})}{\int \pi(\mathbf{y}|\mathbf{X}, \mathbf{\Theta})\pi(\mathbf{\Theta})d\mathbf{\Theta}} \tag{3.1}$$

This formulation of a parametric model is useful in two key areas; the first being from a general standpoint of expressing uncertainty in the model parameters. Indeed, for small clusters with limited data, even when the responses are not linearly separable we may not be certain that the data provided gives a fully accurate representation of the true cluster, and so we can express this uncertainty in a rigorous manner through a Bayesian set-up. Secondly, a Bayesian framework allows for constraints to be placed upon the parameters, which in turn results in realistic

model outputs. Referring back to the singular class response example above, a reasonable prior which gives low probability to extremely large values of the regression coefficients then protects the model from making unrealistic singular predictions.

The use of Bayesian posteriors also naturally allows for prediction of the response of a new observation $\mathbf{x}^\star$ through the definition of the posterior predictive distribution:

$$\begin{aligned}\pi(y^\star \mid \mathbf{x}^\star, \mathbf{y}, \mathbf{X}) &= \int \pi(y^\star \mid \mathbf{x}^\star, \mathbf{y}, \mathbf{X}, \boldsymbol{\Theta})\pi(\boldsymbol{\Theta} \mid \mathbf{x}^\star, \mathbf{y}, \mathbf{X})d\boldsymbol{\Theta} \\ &= \int \pi(y^\star \mid \mathbf{x}^\star, \boldsymbol{\Theta})\pi(\boldsymbol{\Theta} \mid \mathbf{y}, \mathbf{X})d\boldsymbol{\Theta} = \mathbb{E}_{\pi(\boldsymbol{\Theta}|\mathbf{y},\mathbf{X})}\left(\pi(y^\star \mid \mathbf{x}^\star, \boldsymbol{\Theta})\right) \quad (3.2)\end{aligned}$$

Equation (3.2) gives the marginal distribution for a general value of $y^\star$, however, we can also gain specific posterior probabilities. For example, if $Y \in \{0, 1\}$, the posterior probability of $y^\star = 1$ would simply be

$$\mathbb{E}_{\pi(\boldsymbol{\Theta}|\mathbf{y},\mathbf{X})}\left(\pi(y^\star = 1 \mid \mathbf{x}^\star, \boldsymbol{\Theta})\right) \quad (3.3)$$

An important aspect of a Bayesian posterior is the normalisation constant in equation (3.1):

$$Z = \pi(\mathbf{y}|\mathbf{X}) = \int \pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})\pi(\boldsymbol{\Theta})d\boldsymbol{\Theta} \quad (3.4)$$

Z can be viewed as a measure of how well the model explains the data given, and as such is referred to as the 'evidence' of a model, giving the alternate name 'Bayesian evidence'. Due to the integral in equation (3.4) being often intractable[2], calculation of Z typically requires a numerical solution. This can be approached in several ways, some of which are detailed in subsequent sections.

3.1.1 Importance Sampling

In order to provide a method to estimate Z, the fundamentals of importance sampling [34, 35] must be covered, and can typically be viewed as a solution to the

[2]Selection of a conjugate prior would make the integral tractable but this naturally restricts our choice of prior.

following problem with standard Monte Carlo integration.

Equation (3.4) can be interpreted as an expectation with respect to the prior, i.e.

$$Z = \mathbb{E}_{\pi(\boldsymbol{\Theta})}\left(\pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})\right) \tag{3.5}$$

This expectation can then be approximated with a standard Monte Carlo estimate as shown in equation (3.6), given N samples $((\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_N)^T)$ from the prior.

$$Z \approx \frac{1}{N}\sum_{r=1}^{N} \pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta}_r) = \hat{Z} \tag{3.6}$$

One may question the variability of the estimator $\hat{Z}$, which can be expressed as

$$\mathrm{Var}_{\pi(\boldsymbol{\Theta})}(\hat{Z}) = \mathbb{E}_{\pi(\boldsymbol{\Theta})}(\hat{Z}^2) - \left(\mathbb{E}_{\pi(\boldsymbol{\Theta})}(\hat{Z})\right)^2 \tag{3.7}$$

Assuming an i.i.d. sample from the prior (letting $h(\boldsymbol{\Theta}) = \pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})$),

$$\begin{aligned}
\mathbb{E}_{\pi(\boldsymbol{\Theta})}(\hat{Z}) &= \frac{1}{N}\sum_{r=1}^{N} \mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta}_r)) \\
&= \frac{1}{N}\sum_{r=1}^{N} \mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta})) = \mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta})) \qquad (3.8) \\
\mathbb{E}_{\pi(\boldsymbol{\Theta})}(\hat{Z}^2) &= \frac{1}{N^2}\mathbb{E}_{\pi(\boldsymbol{\Theta})}\left(\left(\sum_{r=1}^{N} h(\boldsymbol{\Theta}_r)\right)^2\right) \\
&= \frac{1}{N^2}\left(\sum_{r=1}^{N} \mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta}_r)^2) + \sum_{r\neq s} \mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta}_r))\mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta}_s))\right) \\
&= \frac{1}{N}\mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta})^2) + \frac{N-1}{N}\left(\mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta}))\right)^2 \qquad (3.9)
\end{aligned}$$

Therefore we can express the variance as

$$\begin{aligned}
\mathrm{Var}_{\pi(\boldsymbol{\Theta})}(\hat{Z}) &= \frac{1}{N}\mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta})^2) - \frac{1}{N}\left(\mathbb{E}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta}))\right)^2 \\
&= \frac{1}{N}\mathrm{Var}_{\pi(\boldsymbol{\Theta})}(h(\boldsymbol{\Theta})) \qquad (3.10)
\end{aligned}$$

From equation (3.10) we can deduce that the variability of our estimator depends

on the variability of $h(\boldsymbol{\Theta})$ under the prior. An interpretation of this is that if values with high density under the prior produce extremely varied results when passed through the function h, then the variance of our estimator will be high. Therefore, if we can adapt both the function $h(\boldsymbol{\Theta})$ and the distribution $\pi(\boldsymbol{\Theta})$ such that high density values from the adapted distribution give similar outputs from the adapted function, whilst also still estimating the correct quantity, we will have an estimator with reduced variability. This is the motivation behind importance sampling, and can be expressed through interpreting the Bayesian evidence in the following way[3]

$$\begin{aligned} Z &= \int \pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})\pi(\boldsymbol{\Theta})d\boldsymbol{\Theta} \\ &= \int \pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})\frac{\pi(\boldsymbol{\Theta})}{\tilde{\pi}(\boldsymbol{\Theta})}\tilde{\pi}(\boldsymbol{\Theta})d\boldsymbol{\Theta} \\ &= \mathbb{E}_{\tilde{\pi}(\boldsymbol{\Theta})}\left(\frac{\pi(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})\pi(\boldsymbol{\Theta})}{\tilde{\pi}(\boldsymbol{\Theta})}\right) = \mathbb{E}_{\tilde{\pi}(\boldsymbol{\Theta})}\left(\tilde{h}(\boldsymbol{\Theta})\right) \end{aligned} \tag{3.11}$$

Approximation of the expectation (3.11) now gives an estimator whose variance depends on the variability of $\tilde{h}(\boldsymbol{\Theta})$ under $\tilde{\pi}$, and so selection of $\tilde{\pi}$ (as this also defines $\tilde{h}$) can be utilised to reduce the variance significantly.

Finally, we note that importance sampling is not exclusively a Bayesian concept for determining normalisation constants for posteriors. The scope of importance sampling is much broader than this, as it is perfectly feasible to select any function h along with any distribution. An example of this would be determining the probability of rare events given a distribution. A brute force method would be highly variable due to the rarity of the event, but selection of a truncated distribution which has a much higher density in the target region would produce less variable results. Additionally, the motivation may not even be variance reduction, instead being a method of estimating the expected value of a function using a distribution that is easier to sample from than the true distribution.

[3]Note that this interpretation is only valid if the support of $\tilde{\pi}(\boldsymbol{\Theta})$ contains the support of $\pi(\boldsymbol{\Theta})$.

3.1.2 Sequential Monte Carlo

Importance sampling provides a potential solution to the poor estimation one would achieve with approximating Z using prior samples, however, the question still remains of what distribution to select for $\tilde{\pi}$. Instinctively the desired distribution in terms of variance reduction would be the full posterior, although this returns us to the original issue of obtaining Z. Suitable candidates would then be distributions which are similar to that of the full posterior, but again sampling from these distributions is challenging. Through the technique known as Sequential Monte Carlo (SMC) we can remove the idea of using a single distribution and instead create a sequence of bridging distributions, $\pi_0, \ldots, \pi_T = \pi(\boldsymbol{\Theta}|\mathbf{y}, \mathbf{X})$, to gradually approach the full posterior. With careful construction of these bridging distributions so that each distribution is 'close' to its neighbour in the sequence, we can produce a sequence of samples which eventually arrive at the posterior (whilst also being able to produce a sequence of normalisation constants along the way).

The history behind SMC samplers is extensive and varied, having roots coming from particle filters, sequential importance sampling and Markov Chain Monte Carlo (MCMC). Whilst this history is not covered here, an outstanding introduction into the background and subsequent use of SMC samplers is given by Dai et.al [36]. Indeed, the focus up until this point has been on SMC sampler's capability of producing normalisation constants, however, the ability of SMC samplers to eventually provide samples from the posterior is incredibly useful for inferential purposes, specifically for estimation of the expectation given in equation (3.2).

Amongst the predecessors of SMC samplers, MCMC methods are of particular significance as they remain a popular choice for posterior sampling, and often perform a crucial role as a component of SMC. MCMC can be briefly described as a stochastic process which, starting with an initial value, constructs a chain of samples which will eventually produce samples from the stationary distribution. In Bayesian settings this stationary distribution is the posterior. The general mechanics along with the theoretical guarantees of the process are omitted here, as for the requirements of the methods used in this thesis (specifically the iterated batch importance

sampling scheme) MCMC samplers are only utilised as a component of an SMC sampler, where the theoretical stationary properties are less relevant[4]. Despite this, an important concept that does require definition is that of a Markov Kernel, which is used to transition one sample to the next.

Definition 3.1.1 (Markov Kernel)**.** *Let X and Y be the state space of the current variable and the target variable respectively, and let $M : X \times Y \to [0, 1]$ be a measurable function. If $\int_Y M(x, dy) = 1$ and $\int_{\mathcal{Y}} M(x, dy) = M(x, \mathcal{Y})$ for all measurable subsets $\mathcal{Y} \subset Y$ then M is a Markov kernel.*

For an in-depth description of concepts such as Markov kernels and MCMC a detailed introduction is given by Brooks et.al [37], however a simple intuition of a Markov kernel is the mechanism that allows for the definition of transition probabilities from values in X to measurable subsets of Y. Below we cover the basic form of a generic Bayesian SMC sampler to provide context to Chopin's Iterated Batch Importance Sampling (IBIS) scheme [9], which will be a major component of the UNCOVER method detailed in chapter 4.

Starting initially with samples $(\boldsymbol{\Theta}_1^{\{0\}}, \ldots, \boldsymbol{\Theta}_N^{\{0\}})^T$ from the prior (labelled π_0), in order to gain samples $(\boldsymbol{\Theta}_1^{\{1\}}, \ldots, \boldsymbol{\Theta}_N^{\{1\}})^T$ from the next bridging distribution, π_1, one could apply a Markov kernel $M_1(\boldsymbol{\Theta}_r^{\{0\}}, \boldsymbol{\Theta}_r^{\{1\}})$ that targets π_1 to the individual samples. The samples $(\boldsymbol{\Theta}_1^{\{1\}}, \ldots, \boldsymbol{\Theta}_N^{\{1\}})^T$ obtained from this transition, however, will not be direct samples from π_1, and will have the following proposal distribution

$$\tilde{\pi_1}(\boldsymbol{\Theta}^{\{1\}}) \propto \int M_1(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}}) \pi_0(\boldsymbol{\Theta}^{\{0\}}) d\boldsymbol{\Theta}^{\{0\}} \tag{3.12}$$

As a result the samples will need to be weighted with the ratio of the target distribution over the proposal, i.e. $\pi_1(\boldsymbol{\Theta}^{\{1\}})/\tilde{\pi_1}(\boldsymbol{\Theta}^{\{1\}})$[5].

Unfortunately, the majority of the time the integral $\int M_1(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}}) \pi_0(\boldsymbol{\Theta}^{\{0\}}) d\boldsymbol{\Theta}^{\{0\}}$ is intractable (although as we shall see later there are notable exceptions) and so

[4]Samples are weighted by their neighbouring bridging distributions under SMC, therefore the MCMC sampler is only required to move the samples to an area of higher density with regards to the next bridging distribution.

[5]Note that the weights can be a ratio of un-normalised densities. All that is required is that after calculation the sample weights are transformed such that the sum of all sample weights equals one. This is known as self-normalised importance sampling.

the weights cannot be computed. However, Del Moral et.al [38] offered a different perspective, noting that $M_1(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}})\pi_0(\boldsymbol{\Theta}^{\{0\}})$ can be viewed as the proposal distribution on the joint space $(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}})$. This proposal is then tractable and so by targeting the joint space we can define computable weights. Of course this requires a joint space target, and so we define the target distribution as the distribution proportional to

$$L_0(\boldsymbol{\Theta}^{\{1\}}, \boldsymbol{\Theta}^{\{0\}})\pi_1(\boldsymbol{\Theta}^{\{1\}}) \tag{3.13}$$

where $L_0(\boldsymbol{\Theta}^{\{1\}}, \boldsymbol{\Theta}^{\{0\}})$ is a backward Markov kernel going from $\boldsymbol{\Theta}^{\{1\}}$ to $\boldsymbol{\Theta}^{\{0\}}$. So, with $M_1(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}})\pi_0(\boldsymbol{\Theta}^{\{0\}})$ as the proposal and $L_0(\boldsymbol{\Theta}^{\{1\}}, \boldsymbol{\Theta}^{\{0\}})\pi_1(\boldsymbol{\Theta}^{\{1\}})$ as the target distribution, the un-normalised weights for samples from the proposal can be expressed as

$$\left(\frac{L_0(\boldsymbol{\Theta}_r^{\{1\}}, \boldsymbol{\Theta}_r^{\{0\}})\gamma_1(\boldsymbol{\Theta}_r^{\{1\}})}{M_1(\boldsymbol{\Theta}_r^{\{0\}}, \boldsymbol{\Theta}_r^{\{1\}})\gamma_0(\boldsymbol{\Theta}_r^{\{0\}})}\right) := \tilde{w}^{\{1\}}(\boldsymbol{\Theta}_r^{\{0\}}, \boldsymbol{\Theta}_r^{\{1\}}) \tag{3.14}$$

where γ represents an un-normalised density (i.e. γ_0 and γ_1 are the un-normalised densities of π_0 and π_1 respectively). Crucially, we note that due to the definition of a Markov kernel we have

$$\int L_0(\boldsymbol{\Theta}^{\{1\}}, \boldsymbol{\Theta}^{\{0\}})\pi_1(\boldsymbol{\Theta}^{\{1\}})d\boldsymbol{\Theta}^{\{0\}} = \pi_1(\boldsymbol{\Theta}^{\{1\}}) \tag{3.15}$$

and so π_1 is a marginal distribution of the target distribution for the joint space. Therefore, weights given to the joint samples for the joint target are applicable to the samples $\boldsymbol{\Theta}_r^{\{1\}}$ for $r = 1, \ldots, N$ as weights for the marginal distribution π_1, which allows us to state

$$\left(\frac{\tilde{w}^{\{1\}}(\boldsymbol{\Theta}_r^{\{0\}}, \boldsymbol{\Theta}_r^{\{1\}})}{\sum_{s=1}^{N} \tilde{w}^{\{1\}}(\boldsymbol{\Theta}_s^{\{0\}}, \boldsymbol{\Theta}_s^{\{1\}})}\right) \boldsymbol{\Theta}_r^{\{1\}} \sim \pi_1 \tag{3.16}$$

With equation (3.15) and the equivalent formulation for the marginal distribution of $M_1(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}})\gamma_0(\boldsymbol{\Theta}^{\{0\}})$ with respect to $\boldsymbol{\Theta}^{\{0\}}$, it is clear that the normalisation constants for $M_1(\boldsymbol{\Theta}^{\{0\}}, \boldsymbol{\Theta}^{\{1\}})\gamma_0(\boldsymbol{\Theta}^{\{0\}})$ and $L_0(\boldsymbol{\Theta}^{\{1\}}, \boldsymbol{\Theta}^{\{0\}})\gamma_1(\boldsymbol{\Theta}^{\{1\}})$ are equivalent to the normalisation constants for γ_1 and γ_0 respectively[6]. As a result we can also

[6]This terminology is introduced such that it can be generalised to other bridging distributions, but here we can simplify matters by acknowledging $\gamma_0 = \pi_0$ as π_0 is the prior.

utilise the weights to gain an estimate for the normalisation constant of π_1:

$$\hat{Z}_1 = \frac{1}{N}\sum_{r=1}^{N} \tilde{w}^{\{1\}}(\boldsymbol{\Theta}_r^{\{0\}}, \boldsymbol{\Theta}_r^{\{1\}}) \tag{3.17}$$

In reality what we have estimated is Z_1/Z_0, but given we started with the prior $Z_0 = 1$. In general going from bridging distribution t to $t+1$ we have

$$\begin{aligned} Z_{t+1} &= \int L_t(\boldsymbol{\Theta}^{\{t+1\}}, \boldsymbol{\Theta}^{\{t\}})\gamma_{t+1}(\boldsymbol{\Theta}^{\{t+1\}})d\boldsymbol{\Theta}^{\{t,t+1\}} \\ &= Z_t \int \tilde{w}^{\{t+1\}}(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})M_{t+1}(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})\pi_t(\boldsymbol{\Theta}^{\{t\}})d\boldsymbol{\Theta}^{\{t,t+1\}} \\ \implies \frac{Z_{t+1}}{Z_t} &\approx \frac{1}{N}\sum_{r=1}^{N} \tilde{w}^{\{t+1\}}(\boldsymbol{\Theta}_r^{\{t\}}, \boldsymbol{\Theta}_r^{\{t+1\}}) \end{aligned} \tag{3.18}$$

where $\boldsymbol{\Theta}^{\{t,t+1\}} = (\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})$. Therefore

$$\hat{Z}_t = \prod_{u=0}^{t-1} \widehat{\frac{Z_{u+1}}{Z_u}} \tag{3.19}$$

Through repetition of this weighting procedure, combined with a sequence of forward and backward kernels, we can go from prior samples to full posterior samples by simply applying the forward kernels to the samples at each bridging step and calculating the weights by taking the product of the current weight with all previous weights. Even with the transitioning of samples through the Markov kernels, the weights of some samples can become degenerate. So in order to keep the number of non-degenerate samples high we can resample, with replacement, at each iteration according to the current weights of the sample. This then allows for samples with low weight to be filtered out of the system, ensuring that by the time we reach the full posterior we avoid the scenario where only a few samples can actively contribute to the normalisation constant estimation. Note that there are several available techniques that one can choose for the resampling scheme [39], but for simplicity we continue with a standard multinomial resampling.

Combining all these aspects gives an algorithm for a generic SMC sampler, detailed in algorithm 2.

Algorithm 2: Υ–step Sequential Monte Carlo Sampler

Input : *Un-normalised Bridging Densities* — $\gamma_1, \ldots, \gamma_\Upsilon$,
Forward Kernels — $M_1, \ldots, M_\Upsilon$, *Backwards Kernels* — $L_0, \ldots, L_{\Upsilon-1}$,
Number of Samples — N

Initialisation : Let $t = 0$, $(\boldsymbol{\Theta}_1^{\{t\}}, \ldots, \boldsymbol{\Theta}_N^{\{t\}})^T$ where $\boldsymbol{\Theta}_r^{\{t\}} \sim \pi(\boldsymbol{\Theta})$,
$\mathbf{w}^{\{t\}} = (w_1^{\{t\}}, \ldots, w_N^{\{t\}})^T = \frac{\mathbf{1}}{N}$, $Z = 1$

Step 1 : Let $(\tilde{\boldsymbol{\Theta}}_1^{\{t\}}, \ldots, \tilde{\boldsymbol{\Theta}}_N^{\{t\}})^T = (\boldsymbol{\Theta}_1^{\{t\}}, \ldots, \boldsymbol{\Theta}_N^{\{t\}})^T$

Step 2 : **for** $r = 1, \ldots, N$ **do**
  Sample $\boldsymbol{\Theta}_r^{\{t\}}$ from $\{\tilde{\boldsymbol{\Theta}}_1^{\{t\}}, \ldots, \tilde{\boldsymbol{\Theta}}_N^{\{t\}}\}$, where $\mathbb{P}(\boldsymbol{\Theta}_r^{\{t\}} = \tilde{\boldsymbol{\Theta}}_s^{\{t\}}) = w_s$
end

Step 3 : **for** $r = 1, \ldots, N$ **do**
  Sample $\boldsymbol{\Theta}_r^{\{t+1\}} \sim M_{t+1}(\boldsymbol{\Theta}_r^{\{t\}}, \cdot)$
end

Step 4 : **for** $r = 1, \ldots, N$ **do**
  Let

$$\tilde{w}_r^{\{t+1\}} = \tilde{w}^{\{t+1\}}(\boldsymbol{\Theta}_r^{\{t\}}, \boldsymbol{\Theta}_r^{\{t+1\}}) = \left(\frac{L_t(\boldsymbol{\Theta}_r^{\{t+1\}}, \boldsymbol{\Theta}_r^{\{t\}}) \gamma_{t+1}(\boldsymbol{\Theta}_r^{\{t+1\}})}{M_{t+1}(\boldsymbol{\Theta}_r^{\{t\}}, \boldsymbol{\Theta}_r^{\{t+1\}}) \gamma_t(\boldsymbol{\Theta}_r^{\{t\}})} \right)$$

end

Let

$$\mathbf{w}^{\{t+1\}} = \frac{\tilde{\mathbf{w}}^{\{t+1\}}}{\sum_{r=1}^N \tilde{w}_r^{\{t+1\}}}$$

Step 5 : Update

$$Z = Z \times \frac{1}{N} \sum_{r=1}^N \tilde{w}_r^{\{t+1\}}$$

Step 6 : **if** $t + 1 = \Upsilon$ **then**
  Stop.
else
  Update $t = t + 1$. Go to step 1.
end

Result : *Posterior Samples* — $(\boldsymbol{\Theta}_1^{\{t+1\}}, \ldots, \boldsymbol{\Theta}_N^{\{t+1\}})^T$,
Weight Vector — $\mathbf{w}^{\{t+1\}}$, *Bayesian Evidence* — Z

3.1.2.1 Iterated Batch Importance Sampling

A specific version of the SMC sampler introduced above that will be a key component in later chapters is Chopin's Iterated Batch Importance Sampling (IBIS) scheme [9], so we provide a more detailed exposition of this variant. Here we separate the observation indices $\{1, \ldots, n\}$ into Υ batches, $\mathfrak{B}_1, \ldots, \mathfrak{B}_\Upsilon$, and define the bridging

distributions (noting that observations here are i.i.d.) as

$$\pi_t = \pi\left(\boldsymbol{\Theta} \,\middle|\, \mathbf{y}, \mathbf{X}, \bigcup_{s=1}^{t} \mathfrak{B}_s\right) \propto \left[\prod_{i=1}^{n} [\pi(y_i \mid \boldsymbol{\Theta}, \mathbf{x}_i)]^{\mathbb{1}\left(i \in \bigcup_{s=1}^{t} \mathfrak{B}_s\right)}\right] \pi(\boldsymbol{\Theta}) \tag{3.20}$$

Interestingly, Chopin's method deviates from the generic SMC sampler given in algorithm 2 in a simple but important manner, by changing the order of the steps. The generic SMC sampler follows the order of resample-move-weight, whereas IBIS follows the order of weight-resample-move. The choice of step order does not change the procedure fundamentally, but does allow for further use of the weights as a degenerate sampler indicator, the details of which are clarified later in this section.

For the move step, we require a sampling procedure to transition the old set of particles to a new set more representative of the current bridging distribution. There are many MCMC moves that can achieve this (although for SMC the move need not be associated with MCMC), however, the recommended procedure is the Independent Metropolis–Hastings method [40]. The one-step application of this for a single sample is given in algorithm 3, where q is the proposal distribution.

Algorithm 3: One-step Independent Metropolis–Hastings Sampler

Input : *Current Distribution Samples* — $\boldsymbol{\Theta}^{\{t\}}$, *Proposal Distribution* — q
Step 1 : Sample $\boldsymbol{\Theta}^{\{t+1\}} \sim q$
Step 2 : Let

$$\alpha(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}}) = \min\left\{1, \frac{\pi_{t+1}(\boldsymbol{\Theta}^{\{t+1\}}) q(\boldsymbol{\Theta}^{\{t\}})}{\pi_{t+1}(\boldsymbol{\Theta}^{\{t\}}) q(\boldsymbol{\Theta}^{\{t+1\}})}\right\}$$

Step 3 : Sample $u \sim \mathcal{U}_{[0,1]}$
Step 4 : **if** $u < \alpha(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})$ **then**
  Stop.
else
  Let $\boldsymbol{\Theta}^{\{t+1\}} = \boldsymbol{\Theta}^{\{t\}}$. Stop.
end
Result : *Target Distribution Samples* — $\boldsymbol{\Theta}^{\{t+1\}}$

The proposal distribution was suggested by Chopin to be a multivariate normal

distribution, with parameters

$$\boldsymbol{\mu} = \mathbf{w}^T(\boldsymbol{\Theta}_1^{\{t\}}, \ldots, \boldsymbol{\Theta}_N^{\{t\}})^T \tag{3.21}$$

$$\Sigma = \left((\boldsymbol{\Theta}_1^{\{t\}}, \ldots, \boldsymbol{\Theta}_N^{\{t\}})^T - \boldsymbol{\mu}\right)^T \text{diag}\{\mathbf{w}\} \left((\boldsymbol{\Theta}_1^{\{t\}}, \ldots, \boldsymbol{\Theta}_N^{\{t\}})^T - \boldsymbol{\mu}\right) \tag{3.22}$$

where $\mathbf{w} = (w_1, \ldots, w_N)^T$ refers to the vector of weights associated to the samples $\boldsymbol{\Theta}_1^{\{t\}}, \ldots, \boldsymbol{\Theta}_N^{\{t\}}$ and $\text{diag}\{\mathbf{w}\}_{ij} = w_i \mathbb{1}(i = j)$. The reasoning behind these parameter choices is that the weighted mean and variance is the closest representation we have to the target distribution parameters given the current set of samples. Additionally, although this specification of the parameters gives a weak dependence on the current samples, each proposed sample is not directly obtained from its corresponding current sample and therefore the independent Metropolis–Hastings sampler is still valid. The form of the proposal as a multivariate normal is justified through Bernstein–von Mises theorem.

Theorem 3.1.1 (Bernstein–von Mises)**.** *Let $(Y_1, X_1), \ldots, (Y_n, X_n)$ be i.i.d. variables with likelihood $\prod_{i=1}^n \pi(Y \mid X, \Theta)$. Let; $l = \log(\pi(Y \mid X, \Theta))$, Θ_0 be the true parameter and $I(\Theta) = -\mathbb{E}_{(Y,X)}\left(\frac{\partial^2 l}{\partial \Theta^2}\right)$ be the fisher information matrix. Under certain regularity conditions, we have*

$$\sqrt{n}(\Theta - \Theta_0) \xrightarrow{\mathcal{D}} \mathcal{N}(0, I^{-1}(\Theta_0))$$

Theorem 3.1.1 only shows a singular result from the umbrella term of *Bernstein von Mises theorem*, and the regularity conditions for convergence can vary significantly. Indeed, a wide variety of proofs exist for various conditions [41–43], but for the purposes of this thesis we will assume that the conditions are met for the remainder of the problems unless stated otherwise. Indeed, for Bayesian logistic regression models these conditions can be easily met through selection of a suitable prior, as the log likelihood is twice differentiable.

From theorem 3.1.1, given our prior specification is reasonable, we can assume that the distribution of the partial posterior (i.e. a bridging distribution for an IBIS sampler) after a certain number of steps is approximately normal, as for IBIS

progression through the bridging distributions results in the addition of observations which increases n.

The transition kernel $M_{t+1}(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})$ for this procedure is given as

$$\alpha(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})q(\boldsymbol{\Theta}^{\{t+1\}}) + \left(\int (1 - \alpha(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}}))dq(\boldsymbol{\Theta}^{\{t+1\}})\right) \delta_{\boldsymbol{\Theta}^{\{t\}}}(\boldsymbol{\Theta}^{\{t+1\}}) \tag{3.23}$$

where α is as defined in algorithm 3 and $\delta_{\boldsymbol{\Theta}^{\{t\}}}(\boldsymbol{\Theta}^{\{t+1\}}) = \delta(\boldsymbol{\Theta}^{\{t+1\}} - \boldsymbol{\Theta}^{\{t\}})$ is the Dirac delta function describing a point mass distribution at $\boldsymbol{\Theta}^{\{t\}}$. Through a suitable choice of backward kernels, i.e.

$$L_t(\boldsymbol{\Theta}^{\{t+1\}}, \boldsymbol{\Theta}^{\{t\}}) = \frac{\pi_{t+1}(\boldsymbol{\Theta}^{\{t\}})M_{t+1}(\boldsymbol{\Theta}^{\{t\}}, \boldsymbol{\Theta}^{\{t+1\}})}{\pi_{t+1}(\boldsymbol{\Theta}^{\{t+1\}})} \tag{3.24}$$

the weight equation given in algorithm 2 simplifies to

$$\tilde{w}_r^{\{t+1\}} = \frac{\gamma_{t+1}(\boldsymbol{\Theta}_r^{\{t\}})}{\gamma_t(\boldsymbol{\Theta}_r^{\{t\}})} = \prod_{i=1}^{n} \left[\pi(y_i \mid \boldsymbol{\Theta}_r^{\{t\}}, \mathbf{x}_i)\right]^{\mathbb{1}(i \in \mathfrak{B}_{t+1})} \tag{3.25}$$

Equation (3.25) represents the weights used in the weight step of IBIS. Initially we will have not completed a move step, however, at this stage we already have direct samples from the previous distribution (the prior) and so the form of the weights is unaltered.

One may wonder what the purpose of changing the step order is. Whilst Chopin's paper was published years before a generalised framework was introduced by Del Moral et.al [38], the order Chopin selected alludes to the fact that resample-move steps need not happen at every iteration. Indeed, for bridging distributions sufficiently close, a large amount of samples are unlikely to be degenerate and therefore resampling and moving at every iteration could become unnecessarily computationally expensive. Therefore, if we can detect when the sample set is in need of rejuvenation then we only need resample-move when this occurs. Detection can be achieved by first noting that a sample becomes degenerate when its corresponding weight becomes degenerate, and then noting that degeneracy of the weights can be assessed through their variability. Kong et.al [44] introduced a measure for weight variability, known as the Effective Sample Size (ESS), which for weight vector $\mathbf{w}$ is

given as:

$$\frac{\left(\sum_{r=1}^{N} w_r\right)^2}{\mathbf{w}^T\mathbf{w}} \tag{3.26}$$

The ESS has a minimum value of 1 (which can be achieved when one sample has weight 1 and the other samples have weight 0) and a maximum value of N (which can be achieved when all samples have weight $\frac{1}{N}$). Higher values of the ESS are obtained through greater uniformity of the weights, with the rationale being that uniformity represents either all samples being in areas of high density under the next distribution or all samples being in areas of low density under the next distribution. Given the bridging distributions are selected to be similar to their neighbours it should be highly likely that uniformity implies the former. On the other hand, highly variable weights implies that some samples are in areas of much higher density than other samples, and so a rejuvenation is required to remove these under-performing samples.

So, through selection of a threshold ξ, we need only implement a resample-move step if the ESS falls below ξ. If we do not resample and move, however, we must note that the sample set obviously remains the same and so in essence this translates to remaining at the current distribution (when only considering the samples and not their associated weights). Therefore we have skipped over a bridging distribution and moved to the next distribution in the sequence. This might call into question our assumption of a uniform weighting being desirable; however, if all bridging distributions are similar to their neighbours it still remains much more likely that samples degenerate at different rates as opposed to all samples becoming degenerate at the same time. Chopin noticed a separate issue with the ESS, namely the handling of identical samples. If samples are identical (which can occur through resampling and rejected moves), the ESS can be artificially increased, potentially leading to degeneracy going undetected. A simple example to highlight is the following: one sample will always have an ESS of 1, but a sample set of one sample repeated N times will always have an ESS of N. So to combat this, for the calculation of the ESS we pool identical samples weights together.

All of these techniques combined leads to the IBIS algorithm (algorithm 4).

Algorithm 4: Υ-batch Iterated Batch Importance Sampler

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *ESS Threshold* — ξ, *Batches* —$\mathfrak{B}_1, \dots \mathfrak{B}_\Upsilon$, *Number of Samples* — N

Initialisation : Let $t = 0, \tilde{t} = 0$, $(\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T$ where $\boldsymbol{\Theta}_r^{\{t\}} \sim \pi(\boldsymbol{\Theta})$, $\mathbf{w}^{\{t\}} = (w_1^{\{t\}}, \dots, w_N^{\{t\}})^T = \frac{\mathbf{1}}{N}$, $Z = 1$

Step 1 : **for** $r = 1, \dots, N$ **do**

Let

$$\tilde{w}_r^{t+1} = \prod_{i=1}^{n} \left[\pi(y_i \mid \boldsymbol{\Theta}_r^{\{t\}}, \mathbf{x}_i) \right]^{\mathbb{1}\left(i \in \bigcup_{s=\tilde{t}+1}^{t+1} \mathfrak{B}_s\right)}$$

end

Let $\mathbf{w}^{\{t+1\}} = \frac{\tilde{\mathbf{w}}^{\{t+1\}}}{\sum_{r=1}^{N} \tilde{w}_r^{\{t+1\}}}$

Step 2 : Pool together identical samples: Let $\boldsymbol{\delta}^\dagger$ be the index vector of the unique elements of $(\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T$. Let $\boldsymbol{\delta}^{\dagger\dagger} \in \mathbb{N}^{|\boldsymbol{\delta}^\dagger|}$ be such that $\delta_i^{\dagger\dagger} = |\{r \in \{1, \dots, N\} : \boldsymbol{\Theta}_r = \boldsymbol{\Theta}_{\delta_i^\dagger}\}|$.

Step 3 : **if**

$$\left(\sum_{r \in \boldsymbol{\delta}^\dagger} w_r^{\{t+1\}} \delta_r^{\dagger\dagger} \right)^2 \Bigg/ \sum_{r \in \boldsymbol{\delta}^\dagger} \left(w_r^{\{t+1\}} \delta_r^{\dagger\dagger} \right)^2 < \xi$$

then

Let $\tilde{t} = t + 1$. Let $(\tilde{\boldsymbol{\Theta}}_1^{\{t\}}, \dots, \tilde{\boldsymbol{\Theta}}_N^{\{t\}})^T = (\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T$. Let $q \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} = (\mathbf{w}^{\{t+1\}})^T (\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T$,

$$\Sigma = \left((\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T - \boldsymbol{\mu} \right)^T \operatorname{diag}\{\mathbf{w}^{\{t+1\}}\} \left((\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T - \boldsymbol{\mu} \right)$$

For $r = 1, \dots, N$ sample $\boldsymbol{\Theta}_r^{\{t\}}$ from $\{\tilde{\boldsymbol{\Theta}}_1^{\{t\}}, \dots, \tilde{\boldsymbol{\Theta}}_N^{\{t\}}\}$, where $\mathbb{P}(\boldsymbol{\Theta}_r^{\{t\}} = \tilde{\boldsymbol{\Theta}}_s^{\{t\}}) = w_s^{\{t+1\}}$, and then sample $\boldsymbol{\Theta}_r^{\{t+1\}}$ using algorithm 3.

Update $Z = Z \times \frac{1}{N} \sum_{r=1}^{N} \tilde{w}_r^{\{t+1\}}$

else

$(\boldsymbol{\Theta}_1^{\{t+1\}}, \dots, \boldsymbol{\Theta}_N^{\{t+1\}})^T = (\boldsymbol{\Theta}_1^{\{t\}}, \dots, \boldsymbol{\Theta}_N^{\{t\}})^T$

end

Step 4 : **if** $t + 1 = \Upsilon$ **then**

if $\tilde{t} = \Upsilon + 1$ **then**

Stop.

else

Update $Z = Z \times \frac{1}{N} \sum_{r=1}^{N} \tilde{w}_r^{\{t+1\}}$. Stop.

end

else

Update $t = t + 1$. Go to step 1.

end

Result : *Posterior Samples* — $(\boldsymbol{\Theta}_1^{\{t+1\}}, \dots, \boldsymbol{\Theta}_N^{\{t+1\}})^T$, *Weight Vector* — $\mathbf{w}^{\{t+1\}}$, *Bayesian Evidence* — Z

3.2 Selection of K

Clustering methods seen previously in chapter 2, both supervised and unsupervised, have all had the common issue of a priori specification of the number of clusters, K. The process for selecting K (which is not automatic for the majority of models) can be viewed in two ways: either we aim to select the model which optimises a certain criterion or we apply further Bayesian treatment by treating K as an unknown parameter of the model. The latter will be explored in section 3.2.3 and the former can be viewed from either a frequentist or Bayesian standpoint, both of which are detailed below.

3.2.1 Frequentist Model Selection — Information Criteria

Frequentist methods for selecting K have been mentioned previously, with examples being the silhouette method and the gap statistic for K-means clustering. These methods are specific to unsupervised clustering, however. For supervised clustering a method for selection of K must incorporate the modelling of the response given the covariates. Specifically here we focus on model comparison between parametric models, as this allows for the utilisation of information criteria, which gives a principled methodology for cluster selection (and indeed model selection in general).

A natural trade-off to consider when selecting the number of clusters for a parametric model is that of performance versus complexity. Indeed, this trade-off can even be seen in non-parametric models such as decision trees through pruning of the tree. In a frequentist parametric setting, however, we can view performance as the likelihood value given the parameters and complexity as the number of parameters (which is governed by the number of clusters). Akaike [45] quantifies this trade-off through the Akaike Information Criterion (AIC)

$$\text{AIC} = 2\iota - 2\log(\hat{L}) \tag{3.27}$$

where ι is the total number of estimated model parameters and $\hat{L}$ is the maximum value of the likelihood.

Justification for the criterion in equation (3.27) is rooted in the concept of evaluating the Kullback–Leibler (KL) divergence between the suggested model and the true distribution which generated the data. Indeed, it can be shown [46] that minimising the KL divergence is equivalent to minimising the Kullback discrepancy. The expectation of this discrepancy (with respect to the true distribution) when evaluated with the Maximum Likelihood Estimator (MLE) is asymptotically equivalent to the expected value of the AIC with respect to the true distribution. The key phrase here is asymptotically equivalent, meaning for finite data there are ignored terms (resulting from the Taylor approximations made in the derivation of this result) which could potentially be highly influential when comparing models. Hurvich and Tsai [47] tackle this issue for small datasets with a more accurate approximation.

An example for a specific model, say Finite Mixtures of Logistic Regression (FMLR) models, would be

$$\mathrm{AIC}_{\mathrm{FMLR}} = 2(K(p+1)+K-1) - 2\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\hat{\tau}_k\pi(y_i|\mathbf{x}_i,\hat{\boldsymbol{\beta}}_k)\right) \tag{3.28}$$

where p being the number of covariate variables and $(\hat{\boldsymbol{\tau}},\hat{\boldsymbol{\mathcal{B}}})$ being the maximum likelihood estimates obtained through Iteratively Reweighted Least Squares (IRLS).

Models that produce a large likelihood value with a low number of parameters are favourable, and so we aim to minimise the AIC. An alternative quantification of the trade-off is the Bayesian Information Criterion (BIC) given by Schwarz [48]

$$\mathrm{BIC} = \log(n)\iota - 2\log(\hat{L}) \tag{3.29}$$

Here we can see the two criteria are similar in structure, with the difference coming in the model complexity penalty. The change from 2 to $\log(n)$ results in BIC more heavily penalising complexity in settings where $n \gg \iota$.

For sufficiently large values of n it can be shown that a transformation of the BIC value for a model is a suitable approximation to the Bayesian Evidence. The approximation is given below:

$$Z \approx e^{-\frac{\mathrm{BIC}}{2}} \tag{3.30}$$

A rough guide for the derivation of this approximation would be to produce a second-order Taylor approximation of the log likelihood, and then use this approximation to produce a further Laplace approximation of the integral itself. A complete derivation is given by Konishi and Kitagawa [49]. Naturally, these approximations become exact asymptotically and therefore we arrive at the suitable approximation for large n, provided of course the prior behaves in a suitable way (the MLE being a possible sample from the prior, for example). For small n, however, these approximations will not be suitable and the results of Z and $e^{-\frac{\text{BIC}}{2}}$ can differ wildly. This failure of approximation can also occur if the MLE is not finite, which is a possibility for completely separable data.

For both criteria IRLS [25] is required to determine the maximum likelihood. In order to give the procedure for IRLS, we must first relate the method to typical maximum likelihood estimation. Starting with the log likelihood ($l = \log(L)$), we aim to derive estimates of the model parameters $\boldsymbol{\Theta}$[7] which give

$$\frac{\partial l}{\partial \boldsymbol{\Theta}^T} = 0 \tag{3.31}$$

Typically these set of equations cannot be solved analytically and so starting from an initial point $\boldsymbol{\Theta}_0$ we can produce a Taylor expansion to get the following approximation

$$\frac{\partial l}{\partial \boldsymbol{\Theta}^T} \approx \frac{\partial l}{\partial \boldsymbol{\Theta}^T}(\boldsymbol{\Theta}_0) + (\boldsymbol{\Theta} - \boldsymbol{\Theta}_0)\frac{\partial^2 l}{\partial \boldsymbol{\Theta}^T \boldsymbol{\Theta}}(\boldsymbol{\Theta}_0) \tag{3.32}$$

With this approximation we can then determine the next iteration's value as

$$\boldsymbol{\Theta}_1 = \boldsymbol{\Theta}_0 + \left[-\frac{\partial^2 l}{\partial \boldsymbol{\Theta}^T \boldsymbol{\Theta}}(\boldsymbol{\Theta}_0)\right]^{-1} \frac{\partial l}{\partial \boldsymbol{\Theta}^T}(\boldsymbol{\Theta}_0) \tag{3.33}$$

and can repeat this process until convergence, as shown in algorithm 5.

Algorithm 5 highlights the potential challenges with using information criteria, as convergence of this algorithm in general is not guaranteed, which has been witnessed when regression coefficients tend to infinite values.

[7] $\boldsymbol{\Theta}$ here represents a combined vector of all model parameters, so for example this would take the form of $(\boldsymbol{\tau}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)^T$ for a FMLR model.

Algorithm 5: Iteratively Reweighted Least Squares

Input : *Covariate Matrix* — $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, *Coefficient Vector* — $\boldsymbol{\Theta}$, *Response Vector* — $\mathbf{y} = (y_1, \ldots, y_n)^T$, *Convergence Threshold* — $\eta > 0$

Step 1 : Let

$$\tilde{\boldsymbol{\Theta}} = \boldsymbol{\Theta} + \left[-\frac{\partial^2 l}{\partial \boldsymbol{\Theta}^T \boldsymbol{\Theta}}(\boldsymbol{\Theta}) \right]^{-1} \frac{\partial l}{\partial \boldsymbol{\Theta}^T}(\boldsymbol{\Theta})$$

Step 2 : **if** $\|\tilde{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_2 < \eta$ **then**
  Let $\boldsymbol{\Theta} = \tilde{\boldsymbol{\Theta}}$. Stop.
else
  Let $\boldsymbol{\Theta} = \tilde{\boldsymbol{\Theta}}$. Go to step 1.
end

Result : $\boldsymbol{\Theta}$

3.2.2 Bayesian Model Selection

Examples of model selection in Bayesian settings have been seen previously, namely the Bayesian evidence Z. As stated in section 3.1, Z gives a measure of how well the model is explaining the data. Logically it follows that if model A produces a higher Bayesian evidence than model B, model A should be selected. Whilst the use of Z for model selection has a much wider scope than purely the selection of the number of clusters, models with differing values for K can be compared using this method as they (by definition) present differing model structures. This naturally is true regardless of whether the cluster-level models are identical due to the different number of model parameters. For example consider MoE models where $K = 2$ and $K = 3$ respectively. The form of the individual expert models may be identical but the overall models will have different structure (if there are p covariates then the $K = 2$ model will have $2 \times ((p+1) + (p+1))$ parameters and the $K = 3$ model will have $3 \times ((p+1) + (p+1))$ parameters[8]).

One of the criticisms leveraged at the use of the Bayesian evidence as a measure of model fit is that assessment of the model is too heavily dependent on the prior [50]. Whilst it can be argued from a frequentist standpoint that the prior acting as an abstract penalty term is undesirable, for a large number of observations this becomes

[8]In general, if a one cluster model has $a \in \mathbb{R}$ parameters, a K cluster model must have $K \times a$ parameters.

a less contentious topic as (under certain conditions) Bernstein von Mises theorem shows that the posterior asymptotically is not impacted by the prior.

An alternative to basic Bayesian evidence comparison is to consider the ratio of Bayesian evidences, also known as the Bayes factor [51,52]. This, however, refers to a specific form of model comparison where we have pre-selected a model to be the null hypothesis and the other model acts as an alternative (this selection decides the numerator and denominator for the ratio). In reality it may not be possible to make this distinction for two competing models. Nevertheless, the Bayes factor will produce a value on which we can evaluate our preference for the null hypothesis in response to the alternative. This evaluation is typically carried out in accordance to the Jeffreys scale [53], which provides an interpretation to values which fall within certain ranges.

Computation of Z, as we have seen, is not a trivial task due to the possible intractability of the integral [54]. Methods such as Sequential Monte Carlo (SMC) do provide a practical method of estimation, but the storage of large numbers of samples as well as their propagation through various bridging distributions could result in a computationally expensive method for model comparison.

3.2.3 Bayesian Treatment of K

Given the model parameters are considered unknown and therefore subject to Bayesian analysis, it would appear natural to also consider the number of clusters K as unknown. This treatment of K, however, is not as straightforward as the standard model parameters, as the existence of model parameters intrinsically depends on K itself. There are various different methods to combat this issue, with the two most popular general methods being Reversible Jump Markov Chain Monte Carlo (RJMCMC) [55] and Dirichlet Process Priors [56], both detailed in appendix A.4.

Despite the popularity of these methods, quantifying uncertainty in K (in the setting of joint cohort detection and predictive modelling) results in more complex models being utilised to further depart from the fundamental interpretability aspect of the thesis. Indeed, specifying a prior belief in the number of clusters by definition results in a probabilistic output for the number of clusters, and therefore the cluster

assignment process. This soft clustering output is (as we have argued previously) undesirable if clear separable cohort detection is required, and so does not justify the increased complexity of the model.

3.2.4 Summary

Often in settings where data is partitioned into much smaller subsets, the asymptotic guarantees of parameter estimation in a frequentist framework serve little relevance. Therefore, adopting the Bayesian paradigm allows us to work with small finite data and makes the creation of smaller clusters more viable.

A Bayesian model also avoids potential frequentist issues with model selection when presented with finite data, as the prior can behave as a necessary model constraint (when using the Bayesian evidence as a model selection tool). The estimation of Bayesian evidence can be difficult, but through techniques such as SMC it is possible to provide an accurate estimate.

Finally, we note that the shift from frequentist to Bayesian frameworks should not alter the key requirements of the stakeholder. We can categorise this as internal and external uncertainty quantification. Internal uncertainty quantification expresses uncertainty in the parameters of the models which drive prediction, but does not explicitly express uncertainty in the cohort generation methodology. This is seen as internal as the stakeholder does not witness the uncertainty quantification, as both frequentist and Bayesian models still produce a predictive score, regardless of the handling of the model parameters. On the other hand, examples of external uncertainty quantification are the treatment of the number of clusters as unknown or a soft clustering assignment. This is external as the uncertainty quantification *is* witnessed by the stakeholder as it appears in the final output (which is not desirable from an interpretability perspective). Referring back to a previous example of predicting risk scores for patients whilst simultaneously creating patient cohorts, giving patients risk scores offers assistance for preventative measures to be put in place, but cohort detection allows for separate treatment plans to be devised. If the number of treatment plans is probabilistic this instantly creates an issue, as does patients being assigned to treatment plans probabilistically (ideally patients should

be assigned to one treatment plan and one treatment plan only). This highlights the fine line one must consider when approaching uncertainty quantification for practical applications.

3.3 Graphical Representation of Data

When considering the visualisation of data for stakeholders, it is important to allow the clustering to be interpretable in low dimensions. However, constructing a clustering method which operates in a high-dimensional state but that can be visualised in an interpretable manner in low dimensions is not a trivial task. Construction of such a method can be used in conjuncture with stakeholder's prior opinions on the relevance of certain covariates to cohort generation — a technique which forms a crucial part of the research conducted for this thesis and as such will be covered in detail in chapter 4. An existing alternative to this is to project the data in lower dimensions and then perform the clustering method. An example of this would be Principal Component Analysis [57], but crucially dimension reduction techniques by default require a loss of information when projecting down into a lower space.

Another alternative, which will be the focus of this section, is to provide alternative visual aids to clarify the structure of the data in higher dimensions. This approach has been seen previously in dendograms for hierarchical clustering (see figure A.2). Instead of dendograms, we shall instead represent the data as a graph, where observations form the vertices and edges between vertices will be weighted by the Euclidean distance between observations. As a basic visualisation tool we can plot a complete graph (see section 3.3.1) of the data, colour by cluster and then use edge width to represent the distance between observations (edge width will be given as the reciprocal of the Euclidean distance, meaning observations closer to each other will have a larger edge width). This gives a visual understanding as to how separated observations are within a cluster. An example of this can be seen in figure 3.1 for the iris dataset [58].

It is well known property of the iris dataset that the setosa species is well separated in covariate space, whereas the versicolor and virginica species are much closer

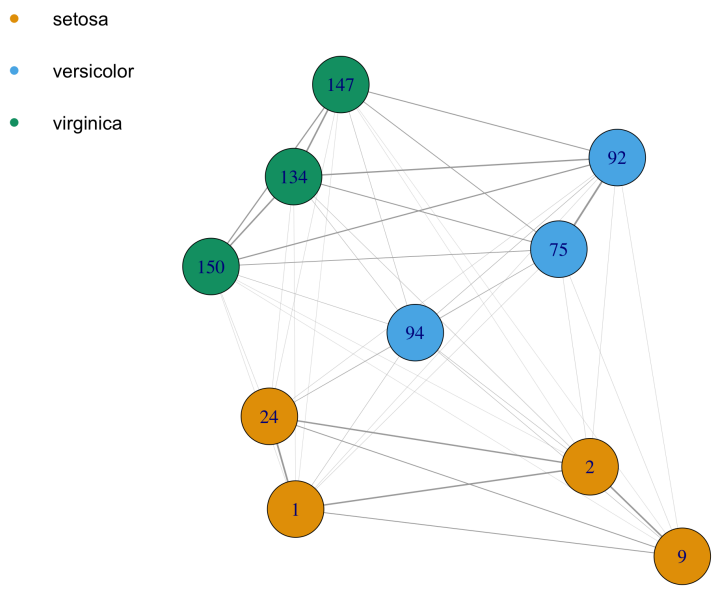


Figure 3.1: Complete graph for a sample of ten observations from the iris dataset. Colours correspond to the species of iris and labels represent observation indices in the dataset.

together; this property can be expressed through the edge thickness in the graph.

In the remainder of this chapter we will highlight the key concepts and terminology of graph theory that will be used in subsequent chapters. In addition to this we shall also provide an in-depth description of the concept of a Minimum Spanning Tree (MST). MSTs provide a crucial function in understanding the underlying structure of the covariate data and also provide much more informative plots than the introductory plot given in figure 3.1. Finally, we will cover the current clustering techniques that utilise MSTs.

3.3.1 Basic Graph Terminology

We begin with the formal definition of a graph:

Definition 3.3.1 (Graph)**.** *A Graph* $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$ *is an ordered pair consisting of a set of vertices* $\mathfrak{V}$ *and a set of edges* $\mathfrak{E}$*.*

In the context of data representation, we can assume the set of vertices $\mathfrak{V} = \{1, \ldots, n\}$ represent the index set for our observations or alternatively the row index

of the matrix $\mathbf{X}$. Elements of the edge set $\mathfrak{E}$ take the form $\{i, j\}$ for the edge connecting observation i to observation j. In this context i and j are known as endpoints of the edge. The type of graph which is the focus for data representation is a weighted graph, which insists that each edge $\{i, j\}$ is assigned a weight e_{ij}. Typically we assume that these weights are generated through the Euclidean distance metric

$$e_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2 \tag{3.34}$$

Due to this definition of edge weights we initially represent our data as a complete graph.

Definition 3.3.2 (Complete Graph)**.** *A Graph $\mathcal{G}$ is said to be complete if for every pair of vertices i and j there exists an edge $\{i, j\}$.*

Other important concepts within graph theory are that of walks and paths:

Definition 3.3.3 (Walk)**.** *A walk $\mathcal{W}$ is a sequence of edges that results in a sequence of vertices; letting $\mathcal{W}_i = \{a, b\}$ and $\mathcal{W}_{i+1} = \{c, d\}$, $\mathcal{W}$ cannot be a walk if $b \neq c$.*

Definition 3.3.4 (Path)**.** *A path $\mathcal{P}$ is a walk such that the resulting sequence of vertices is unique.*

The concept of a path can lead to certain properties of a graph being defined, such as the diameter of a graph and a connected graph:

Definition 3.3.5 (Diameter)**.** *Let the weight of a path be the summation of the weights of all the edges in the path. Let the shortest path between vertices i and j be the path from i to j with the lowest weight. Considering the set of all possible shortest paths in the graph $\mathcal{G}$, the diameter path of $\mathcal{G}$ is the path in this set with the largest weight. The diameter of $\mathcal{G}$ is the weight of the diameter path.*

Definition 3.3.6 (Connected Graph)**.** *A graph $\mathcal{G}$ is said to be connected if there exists a path between every pair of vertices.*

By definition a complete graph is a connected graph, as the edge between a pair of vertices is a path, and so creation of a complete graph from the data also ensures the entire dataset is connected. The notion of a graph being connected has parallels

to the concept of connected regions within covariate space. Connected regions have been seen before, for example with K-means, and offer confidence to the user that two sets of observations in the same cohort cannot have wildly differing attributes. As such another important definition (whose relevance with respect to clustering will be apparent in subsequent chapters) is that of subgraphs and components:

Definition 3.3.7 (Subgraph and Edge-induced Subgraph)**.** *A graph $\mathcal{G}^\star = (\mathfrak{V}^\star, \mathfrak{E}^\star)$ is known as a subgraph of $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$ if; $\mathfrak{V}^\star \subseteq \mathfrak{V}$, $\mathfrak{E}^\star \subseteq \mathfrak{E}$ and for every $\{i, j\} \in \mathfrak{E}^\star$, $i \in \mathfrak{V}^\star$ and $j \in \mathfrak{V}^\star$. An edge-induced subgraph is a subgraph defined by the edge set $\mathfrak{E}^\star$, with $\mathfrak{V}^\star$ consisting of only the edge endpoints in $\mathfrak{E}^\star$.*

Definition 3.3.8 (Component)**.** *$\mathcal{G}^\star$ is a component of graph $\mathcal{G}$ if $\mathcal{G}^\star$ is a connected subgraph of $\mathcal{G}$ and there does not exist any other subgraph of $\mathcal{G}$ that contains $\mathcal{G}^\star$ and is still connected.*

As final definitions we introduce the idea of cycles and cuts, concepts crucial to the construction of minimum spanning trees:

Definition 3.3.9 (Cycle)**.** *A cycle is a walk in which; the resulting vertex sequence begins and ends at the same vertex and the removal of the start or end point of the sequence results in a path.*

Definition 3.3.10 (Cut and Cut-set)**.** *A cut is a partition of the vertices of a graph $\mathcal{G}$ into two sets. The set of edges which have endpoints in different vertex sets is known as the cut-set.*

An example of a cut of the vertex set $\mathfrak{V} = \{1, \ldots, 10\}$ into sets $\mathfrak{V}_1 = \{1, \ldots, 5\}$ and $\mathfrak{V}_2 = \{6, \ldots, 10\}$ from a complete graph is given in figure 3.2. Removal of all of the edges in a cut set results in the splitting of one component of the graph into two. This fact is important to remember for the UNCOVER algorithm in chapter 4. This section covers a small set of definitions crucial to the research presented in this thesis, however, for a more general introduction to graphical concepts and theory West et.al [59] is highly recommended.

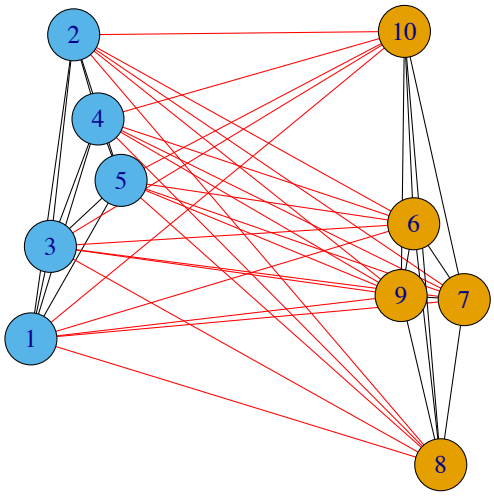


Figure 3.2: Complete graph for ten vertices. The vertex set is partitioned into two sets, represented by vertex colour. The edges highlighted in red are edges belonging to the cut-set.

3.3.2 Minimum Spanning Trees

Minimum Spanning Trees (MSTs) offer an insight into the structure of covariate data without enforcing any pre-determined properties (for example K-means clustering enforcing linear separability). Once the data has been represented graphically an MST simply offers a minimalist structure which captures the distance properties in the data[9]. Below is the formal definition of spanning trees and minimum spanning trees:

Definition 3.3.11 (Spanning Tree)**.** *If* $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$ *is a connected graph, then* $\mathfrak{T} \subseteq \mathfrak{E}$ *is a Spanning Tree of* $\mathcal{G}$ *if the edge-induced subgraph* $\mathcal{G}_{\mathfrak{T}} = (\mathfrak{V}_{\mathfrak{T}}, \mathfrak{T})$ *is such that;* $\mathfrak{V} = \mathfrak{V}_{\mathfrak{T}}$, $|\mathfrak{T}| = |\mathfrak{V}| - 1$ *and* $\mathcal{G}_{\mathfrak{T}}$ *is connected.*

Definition 3.3.12 (Minimum Spanning Tree)**.** *Let* $\mathfrak{T}^{\star}$ *be the set of all possible*

[9]One may argue that the distance metric is pre-defined by the user and as such we are enforcing some constraints on the data. Whilst this is valid, the distance metric can be defined in a manner that best suits the individual problem, with the Euclidean distance being a common dissimilarity measure for numerical data.

spanning trees of $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$. *Spanning tree* $\mathfrak{T}$ *is a minimum spanning tree if*

$$\mathfrak{T} = \arg\min_{\tilde{\mathfrak{T}} \in \mathfrak{T}^\star} \left\{ \sum_{\{i,j\} \in \tilde{\mathfrak{T}}} e_{ij} \right\}$$

Due to the properties of an MST (that for a connected graph of n observations it contains $n-1$ edges and connects n vertices) it follows that an MST contains no cycles. Therefore the connectivity within the graph is minimal, a fact which can be exploited for clustering purposes. This will be explored briefly below and in more detail in chapter 4.

Figure 3.3 gives a visual example of an MST edge-induced subgraph. The structure of the data is highlighted here as we can clearly see how the numerical covariates have an impact on species categorisation, as the MST shows how each of the different species can be separated.

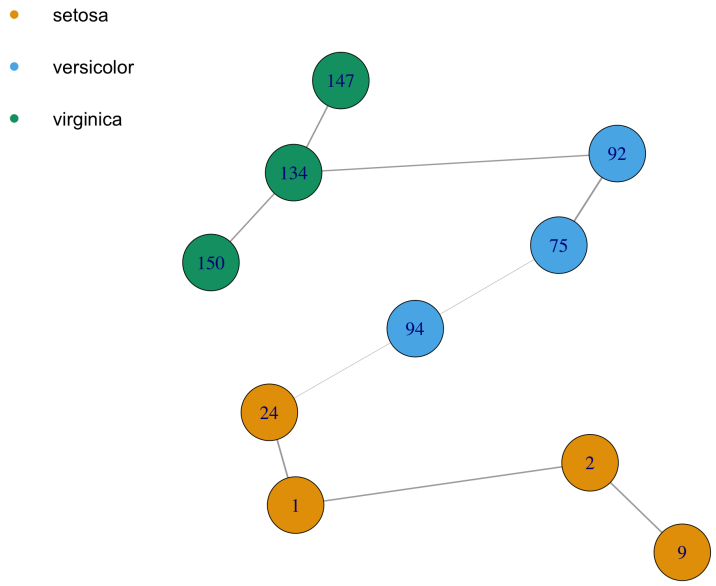


Figure 3.3: MST edge-induced subgraph of the graph given in figure 3.1. Colours correspond to the species of iris and labels represent observation indices in the dataset.

Construction of an MST is not a unique process, with multiple methods introduced each utilising different properties of MSTs [60]. In this thesis we will focus

on Prim's algorithm [61]. Alongside Kruskal's algorithm [62], Prim's algorithm remains a popular method for constructing MSTs due to the algorithm's interpretable methodology. It must be noted, however, that there are more complex computationally efficient algorithms available, and indeed a vast amount of literature has been dedicated to the production of these algorithms [63–65]. Even within the methodologies of Prim and Kruskal, different computational implementations have large effects on the efficiency of the algorithms with regard to run-time. Specifically for Prim's, differences include whether the implementation is achieved through matrix manipulation or priority queuing [66]. The rest of this section will not focus on the computation of MSTs, however, this is discussed in section 8.1.3.

Prim's algorithm utilises the following lemma:

Lemma 3.3.1 (Minimum Spanning Tree Cut Property)**.** *Let $\mathcal{G}$ be a connected graph and let $\mathfrak{T}$ be any MST of $\mathcal{G}$. For any cut of $\mathcal{G}$, if the cut set ($\mathfrak{C}$) of this cut contains an edge $\{i, j\}$ such that every other edge in $\mathfrak{C}$ has weight strictly greater than e_{ij}, then $\{i, j\} \in \mathfrak{T}$.*

The proof of this lemma is straightforward by considering the contrary that $\{i, j\} \notin \mathfrak{T}$. Because of the connectivity property of an MST, every cut set in $\mathcal{G}$ must contain an edge belonging to $\mathfrak{T}$, therefore if $\{i, j\} \notin \mathfrak{T}$ then there must be at least one other edge in $\mathfrak{C}$ belonging to $\mathfrak{T}$. Addition of $\{i, j\}$ to $\mathfrak{T}$ would then create a cycle containing both $\{i, j\}$ and another edge in $\mathfrak{C}$, say edge $\{a, b\}$. Removal of $\{a, b\}$ would then produce a spanning tree (due to the addition of $\{i, j\}$), however, the weight of this tree would be less than the tree not containing $\{i, j\}$, therefore exclusion of $\{i, j\}$ cannot produce a spanning tree of minimal weight.

As a result, starting with a cut which separates a singular vertex from the other vertices we can determine an MST edge. Moving endpoints to the other side of the partition then creates a sequence of cuts which will accumulate in a full MST being formed. This is the basis of Prim's algorithm, and is given in full in algorithm 6.

It is important to note that algorithm 6 was initialised using a random vertex. In the majority of scenarios, specifically where there exists a unique MST, the choice of the initial vertex has no effect on the outputted MST. However, in the setting where there does not exist a unique MST, the choice of initial vertex will influence

Algorithm 6: Prim's Algorithm

Input : *Graph* — $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$
Initialisation : Let $\mathfrak{T} = \emptyset$, $\tilde{\mathfrak{V}} = \emptyset$.
Step 1 : Select vertex i at random from $\mathfrak{V}$ and add this vertex to $\tilde{\mathfrak{V}}$.
Step 2 : **while** $\tilde{\mathfrak{V}} \neq \mathfrak{V}$ **do**
  Let $\mathfrak{C}$ be the cut set between $\tilde{\mathfrak{V}}$ and $\mathfrak{V} \backslash \tilde{\mathfrak{V}}$. Select edge $\{a, b\} \in \mathfrak{C}$ such that

$$e_{ab} = \min_{c \in \tilde{\mathfrak{V}}, d \in \mathfrak{V} \backslash \tilde{\mathfrak{V}}} \{e_{cd}\}$$

  Add $\{a, b\}$ to $\mathfrak{T}$ and add b to $\tilde{\mathfrak{V}}$.
end
Result : *Minimum Spanning Tree* — $\mathfrak{T}$

the output. In extreme situations different initialisations could produce MSTs that capture different aspects of the data's structure. As such, relating back to the graphical representation of data, one should be cautious of the distance or dissimilarity metric used in order to ensure these extreme situations do not occur.

Moving away briefly from the setting of connected graphs, consider a scenario now in which the graph is unconnected and formed through a union of components of the graph. Here we no longer have the necessary requirements to produce an MST, however, we can produce a Minimum Spanning Forest (MSF).

Definition 3.3.13 (Minimum Spanning Forest)**.** *Assume graph* $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$ *can be partitioned into* $K > 1$ *components, i.e.* $\mathcal{G} = \bigcup_{k=1}^{K} \mathcal{G}_k$*. Let* $\mathcal{G}_k$ *have its own associated minimum spanning tree* $\mathfrak{T}_k$*, for* $k = 1, \ldots, K$*. Then* $\mathfrak{T}$ *is a minimum spanning forest if* $\mathfrak{T} = \bigcup_{k=1}^{K} \mathfrak{T}_k$*.*

The use of MSTs and MSFs for unsupervised cluster analysis is well founded [67–69], and in fact a particular use of MSTs for clustering has been presented previously in the form of Agglomerative Single Linkage Hierarchical Clustering (ASLHC) [70]. Indeed, assume that for ASLHC the target is $K = 1$, then consider the process of combining clusters as adding an edge between the two observations that produced the minimum distance (which resulted in the clusters combination) to $\mathfrak{T}$. Due to the definition of single linkage, all edges added satisfy the cut property and therefore must all be MST edges, so if $K = 1$ the process will create an MST. As a result ASLHC can be achieved through creation of an MST and then the subsequent

removal of the $K-1$ edges with the largest distance.

ASLHC is by no means the only way in which one could perform unsupervised clustering using MSTs — there is a vast array of algorithms to produce clusters from the MST structure, depending on the criteria for edge removal [71]. Interestingly, the use of MSTs does not need to be restricted to unsupervised learning. This is a core principle of the UNCOVER method given in chapter 4, but the idea of supervised spanning tree clustering has been documented before by Luo et.al [72]. The methodology in this paper regards spatial modelling, specifically a spatial regression model in which the spatial data is partitioned for each regression coefficient using a Bayesian treatment of the spanning trees combined with a RJMCMC algorithm. This differs somewhat to the problem outlined in this thesis (although the UNCOVER method highlighted in chapter 4 is capable of clustering based on spatial data), however, this method does highlight one of the multiple ways graph theory can be utilised for supervised clustering.

CHAPTER 4

UNCOVER: Utilising Normalisation Constant Optimisation Via Edge Removal

The creation of a hard clustering output for cohort detection is an important aspect for many practical applications of predictive modelling. Expressing the covariate data graphically allows a hard clustering output to be achieved through the removal of edges to create components of the graph. Interpreting these components as cohorts then gives the hard clustering desired. Furthermore, utilising minimum spanning trees, we are able to capture the structure of the data in covariate space without the need for distributional assumptions.

Whilst a graphical representation of the covariate data provides a framework for interpretable clustering via edge removal, the other key outline for this thesis, predictive power of the resulting model, remains unaddressed. Incorporating the response when creating the components of the graph would achieve this, and so this chapter will be focused on detailing a novel procedure to accomplish just that. By constructing a Bayesian product model which is induced by the current form of the graph, we can obtain a principled model selection technique through comparison of the Bayesian evidence of the various models suggested, resulting in a model with components/clusters which best explains the data. In more detail, we aim to opti-

mise the selection of an edge to remove *given* the current state of the graph. As such, the method described in this chapter is greedy, but avoids the Bayesian treatment of K whilst still allowing the number of clusters to vary. Giving the Bayesian Evidence it's alternate name of Normalisation Constant explains the title for this method — Utilising Normalisation Constant Optimisation Via Edge Removal (UNCOVER).

This chapter will extensively detail the UNCOVER procedure, highlighting; the initialisation of the graph and definition of the model structure, the process of estimating the Bayesian evidence of the current model and potential models, the removal edges and finally the reintroduction of edges to meet pre-specified criteria.

4.1 Initialisation

4.1.1 Sub-selection of Covariates

Starting initially with the covariate matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, we wish to construct a graph from this data which has a minimum spanning tree as an edge set. However, given that the construction of said graph primarily is to aid interpretability, one must take careful consideration into which covariates are selected. Indeed, the minimum spanning tree generated from the data guides the possible cohorts one can obtain from UNCOVER, and as a result expert opinion can be introduced here to select the attributes which are believed to form the most informative cohorts.

Referring back to the SPARRA example in section 1.2, the cohorts generated for SPARRA v3 were selected by medical professionals to mainly encompass age, with type of interaction with the healthcare system coming in as a secondary factor. Clearly here age is an interpretable attribute in which to cluster patients as it was devised by the people utilising the model. On the other hand, an attribute such as SIMD (Scottish Index of Multiple Deprivation — this categorises how deprived a patient's home location is) may not be favourable as spatially driven intervention plans could have ethical issues as well as a lack of consistency for general practitioners covering several areas. Therefore, selecting a subset of the covariates for construction of the graph is beneficial when UNCOVER is utilised in practical settings.

In addition to this, from a statistical point of view, selection of a subset of the

covariates could potentially provide a much clearer structure in which to cluster observations. An example of this can be seen in figure 4.1, where we have three clusters, and we state that each cluster has a unique relationship between the covariates and a binary response. In this setting, building a minimum spanning tree

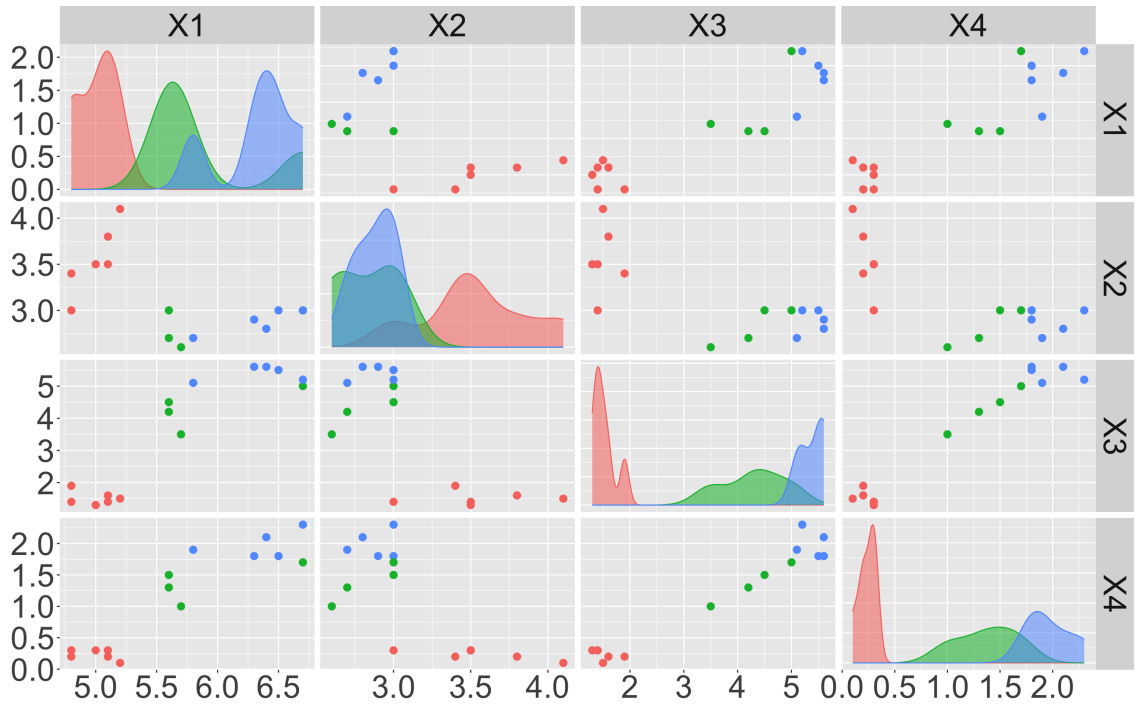
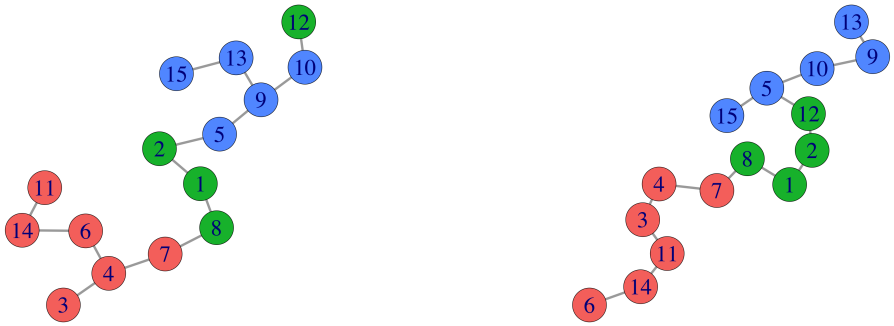


Figure 4.1: Pairs plot for the 4 numerical attributes of 15 observations. Colours correspond to true cluster.

based purely on covariates X3 and X4 would likely result in a better model of the response due to the clustering structure being clearer for this subset of the covariates. This is highlighted in figure 4.2. Of course here a model has not been specified, however, any model utilising a Minimum Spanning Tree (MST) structure would not be able to achieve a completely accurate partition of the data when using all of the covariates.

In general, without expert guidance on the subset of the covariates that would produce cohorts of most use to the stakeholder, one could employ variable selection techniques to determine the most appropriate subset. Examples of such techniques are forward selection and backward elimination [30] (see section 6.2 for an example of forward selection being utilised for the derivation of a covariate subset, and appendix B.1 for further discussion on covariate subset selection). Typically the op-



(a) MST using all covariates **(b)** MST using just covariates X3 and X4

Figure 4.2: Minimum Spanning Trees (MSTs) for the dataset showcased in figure 4.1, constructed using different subsets of the covariates. Vertex labels represent observation index and colour represents cluster.

timality criterion for this procedure would be the Bayesian evidence of the resulting UNCOVER model.

Note that sub-selection of the covariates can be framed as an ad-hoc dimension reduction technique, where the aim of the reduced dataset is to capture the true clustering structure of the data. This is in contrast to other popular dimension reduction techniques such as principal component analysis, which reduces the dimension of the data with the aim to capture variability. As mentioned in chapter 3, graphical plotting in general allows for visualisation of the properties of a high dimensional dataset, but sub-selection of the covariates also would allow for direct visualisation of the data and the cohorts. Indeed, because the cohorts formed by UNCOVER are heavily influenced by the MST, the concept of cohorts having similar attributes will be captured by the plotting of solely the MST attributes. An example of this can be seen in figure 4.1, as if only X3 and X4 are selected for the MST construction, then only plotting X3 against X4 is necessary to capture the potential cohorts that could be formed.

Let $\mathfrak{P} \subseteq \{1, \ldots, p\}$ be a subset of the covariate indices of $\mathbf{X}$. Letting $\mathbf{X}_{\cdot\mathfrak{P}}$ be the submatrix of $\mathbf{X}$ induced by $\mathfrak{P}$, this then allows for the definition of the complete

graph $\mathcal{G} = (\mathfrak{V}, \mathfrak{E})$ with respect to $\mathbf{X}$ and $\mathfrak{P}$:

$$\mathfrak{V} = \{1, \ldots, n\} \tag{4.1}$$

$$\mathfrak{E} = \{\{i, j\}; i \in \mathfrak{V}, j \in \mathfrak{V}, i \neq j\} \tag{4.2}$$

$$e_{ij} = \|\mathbf{x}_{i,\mathfrak{P}} - \mathbf{x}_{j,\mathfrak{P}}\|_2 \tag{4.3}$$

Using Prim's algorithm [61] (algorithm 6) we can obtain the MST $\mathfrak{T}$, which then induces the subgraph $\mathcal{G}_{\mathfrak{T}} = (\mathfrak{V}, \mathfrak{T})$.

As stated before, obtaining an MST offers structure in the covariate space that could aid with interpretability without the need to assume a distributional form for the covariates. However, there is a secondary computational benefit to this method as well. Consider the unconstrained case when initially the task is to partition the data into two sets. This is equivalent to placing n objects into 2 non-empty sets[1], of which the number of unique ways to achieve this is a Stirling number of the second kind [73], i.e.

$$S(n, 2) = \frac{1}{2}\sum_{i=0}^{2}(-1)^i\binom{2}{i}(2-i)^n = 2^{n-1} - 1 \tag{4.4}$$

Examining all possible partitions through brute force is an $O(2^n)$ operation. In contrast the UNCOVER setting need only examine $n-1$ edges of the graph $\mathcal{G}_{\mathfrak{T}}$ (as $|\,\mathfrak{T}\,| = n-1$) and as such the process of considering the next possible model through removal of an edge is a greatly reduced $O(n)$ operation.

As final points on MST construction, we note that the Euclidean distance metric used in equation 4.3 is replaceable with other distance metrics if required. An example may be a metric to incorporate mixed type variables. However, the selection of metrics for mixed variables is a contentious topic within the framework of UNCOVER, as attempts to place categorical and numerical attributes on the same scale becomes challenging if one has to introduce further data for prediction. To expand on this, we take a popular choice of metric for mixed data, Gower's distance [74], as an example. For a numerical variable ϱ and $n \times p$ covariate matrix $\mathbf{X}$, Gower's

[1]This has been seen previously with divisive hierarchical clustering.

distance between observations i and j for variable ϱ is given as:

$$\frac{|x_{i\varrho} - x_{j\varrho}|}{\max_{a\in\{1,\dots,n\}}\{x_{a\varrho}\} - \min_{a\in\{1,\dots,n\}}\{x_{a\varrho}\}} \tag{4.5}$$

The use of equation (4.5), specifically the denominator, allows numerical variable distances to be on the scale $[0, 1]$ which is crucial for their integration with categorical variables (whose distances are also scaled to be within the region $[0, 1]$). However, UNCOVER is a predictive model, and as such the assignment of new observations to clusters is vital for prediction. If a new observation contains a numerical value which lies outside of its respective range, then there will exist a distance which does not belong to the region $[0, 1]$, breaking the concept of placing numerical and categorical variables on the same scale. One may be tempted then to update the range; however, this act could change the form of the MST used to initialise UNCOVER which would then clearly invalidate the outputted clusters.

There are specific circumstances where Gower's distance would be appropriate, however. If there are known boundaries for numerical attributes (for example an observation's age can reasonably be constrained to the interval $[0, 200]$), then replacing the range of the attribute in equation (4.5) with the length of the interval would resolve the issue. Furthermore, for new observations which are deemed extreme outliers based on the training data, one may be cautious of producing a prediction regardless, as uncertainty in the model's output will be high whichever cluster is assigned due to extrapolation. An application of UNCOVER with mixed data types is given in section 6.4.

In addition to metric specification, scaling of the data before implementation of UNCOVER is highly recommended. Scaling occurs through subtraction of the variable mean and division over the variable standard deviation of each column in the dataset, and is necessary when constructing an MST to ensure that certain variables do not have a dominating effect when calculating distances. An example of this would be an MST construction based on the variables sugar (measured in grams) and flour (measured in milligrams). Simply based on the units of measurement, observations are likely to have a much larger discrepancy in flour than in sugar,

resulting in an unwanted bias towards flour when constructing the MST.

4.1.2 Bayesian Product Logistic Regression Models

For UNCOVER the base model of the relationship between the covariates and the response is the logistic regression model. This makes an assumption of linearity between the covariates and the regression coefficients which may not be appropriate for certain types of datasets, however, we note that UNCOVER as a modelling framework is not intrinsically tied to its chosen base model. Indeed, any Bayesian model which encapsulates the relationship between the response and the covariates is suitable for use in UNCOVER. However, the logistic regression model is selected here due to the interpretability the model has to stakeholders as well as the ease at which one can apply a Bayesian treatment to this model.

Initially, given the graph $\mathcal{G}_{\mathfrak{T}}$, we have a one cluster model and so the Bayesian set-up of the posterior for the model regression coefficients ($\boldsymbol{\beta}$) can be expressed in the standard way through equation (3.1) and the following likelihood equation:

$$\pi(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^{n} (1 + e^{-\mathbf{x}_i^T \boldsymbol{\beta}})^{-y_i} (1 + e^{\mathbf{x}_i^T \boldsymbol{\beta}})^{-(1-y_i)} \tag{4.6}$$

In order to compare models of differing numbers of clusters we must generalise this set-up to models which have K clusters. This is accomplished through a product model, where we express the clustering through the partitioning on the index set $\mathfrak{V}$ into K sub-sets[2]:

$$\mathfrak{V} = \{\mathfrak{V}_1, \ldots, \mathfrak{V}_K\} \tag{4.7}$$

[2]For consistency if $K = 1$ we let $\mathfrak{V} = \{\mathfrak{V}_1\}$.

This general model can be expressed as:

$$\pi(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K \mid \mathbf{y}, \mathbf{X}, \mathfrak{V}) = \frac{\pi(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K, \mathfrak{V})\pi(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)}{\pi(\mathbf{y} \mid \mathbf{X}, \mathfrak{V})} \tag{4.8}$$

$$\pi(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K, \mathfrak{V}) = \prod_{k=1}^{K}\prod_{i=1}^{n}\left[(1+e^{-\mathbf{x}_i^T\boldsymbol{\beta}_k})^{-y_i}(1+e^{\mathbf{x}_i^T\boldsymbol{\beta}_k})^{-(1-y_i)}\right]^{\mathbb{1}(i\in\mathfrak{V}_k)} \tag{4.9}$$

$$\pi(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K) = \prod_{i=1}^{K}\pi(\boldsymbol{\beta}_k) \tag{4.10}$$

$$\begin{aligned}\pi(\mathbf{y} \mid \mathbf{X}, \mathfrak{V}) &= \int \pi(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K, \mathfrak{V})\pi(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)d\boldsymbol{\beta}_1 \ldots d\boldsymbol{\beta}_K \\ &= Z\end{aligned} \tag{4.11}$$

Dissecting equations (4.8 – 4.11), we have described a hard cluster product model. The hard clustering is evident through the indicator function in equation (4.9), which operates in a similar manner to the latent cluster assignment matrix used for finite mixtures of logistic regression and mixture of experts models, but here the cluster assignment is not latent (see section 4.3). As a result of this, we no longer need variables such as $\boldsymbol{\tau}$ or softmax functions such as $\mathbf{g}$ as we have an explicit hard clustering, therefore we are not assigning proportions of all models to single observations. We have also made an assumption of independent and identically distributed priors for each of the K clusters in equation (4.10). The notion that clusters in this model are completely separate gives a justification toward this assumption, as we would not expect interaction between the coefficients of separate models. The i.i.d. priors means that Z, given in equation (4.11), reduces to:

$$\begin{aligned}Z &= \int \prod_{k=1}^{K}\prod_{i=1}^{n}\left[(1+e^{-\mathbf{x}_i^T\boldsymbol{\beta}_k})^{-y_i}(1+e^{\mathbf{x}_i^T\boldsymbol{\beta}_k})^{-(1-y_i)}\right]^{\mathbb{1}(i\in\mathfrak{V}_k)}\pi(\boldsymbol{\beta}_k)d\boldsymbol{\beta}_k \\ &= \prod_{k=1}^{K}\int \prod_{i=1}^{n}\left[(1+e^{-\mathbf{x}_i^T\boldsymbol{\beta}_k})^{-y_i}(1+e^{\mathbf{x}_i^T\boldsymbol{\beta}_k})^{-(1-y_i)}\right]^{\mathbb{1}(i\in\mathfrak{V}_k)}\pi(\boldsymbol{\beta}_k)d\boldsymbol{\beta}_k \\ &= \prod_{k=1}^{K} Z_k\end{aligned} \tag{4.12}$$

where the interchanging of product with integral is possible through a combination of cluster independence and the Fubini–Tonelli theorem [75]. This separation of the

Bayesian evidence has major computational benefits in regard to estimation, which will be discussed in detail in chapter 5. This allows for a final representation of the posterior given in equation (4.8) as a product of K sub-posteriors:

$$\pi(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K \mid \mathbf{y}, \mathbf{X}, \mathfrak{V}) = \prod_{k=1}^{K} \pi(\boldsymbol{\beta}_k \mid \mathbf{y}, \mathbf{X}, \mathfrak{V}_k) \tag{4.13}$$

4.2 Assessing Cluster Quality

Reverting back to our initialisation of the system as a one cluster model, i.e. $\pi(\boldsymbol{\beta}_1 \mid \mathbf{y}, \mathbf{X}, \mathfrak{V}_1 = \{1, \ldots, n\})$, the natural choice for assessing the quality of this model (given the Bayesian framework under which UNCOVER is defined) is the Bayesian evidence Z. As mentioned in section 3.1 the estimation of Z can be achieved through Sequential Monte Carlo (SMC) [36] techniques, specifically Iterated Batch Importance Sampling (IBIS) [9].

Although we allow the user to specify the Effective Sample Size (ESS) threshold ξ within UNCOVER[3], the selection of batches, $\mathfrak{B}_1, \ldots, \mathfrak{B}_\Upsilon$, will be restricted. Namely the restriction is that $\Upsilon = n$ and as such $|\mathfrak{B}_s| = 1$ for $s = 1, \ldots, \Upsilon$. The justification for this is that batches were introduced by Chopin as an efficient way to reduce computational time, with the caveat being that this would produce poor samples if the batches were large enough to cause the bridging distributions to no longer be close to their neighbours in the sequence. In algorithmic terms dissimilar neighbouring distributions results in all weights becoming degenerate at the same iteration, meaning degeneracy of the samples can no longer be detected by the ESS and therefore will not be resampled and moved to create a more suitable set of samples. Even if detected, degeneracy of all samples simultaneously would provide no guarantee of a successful move step as the proposal distribution relies on degeneracy information from the weights to produce a distribution close to the target.

Chopin's original algorithm did not reference the generation of Bayesian evidences, however, it follows that poor samples produce poor estimates of Z. Given

[3]Recommendations for ξ along with the number of SMC samples N can be found in appendix B.1.

Z is crucial for the selection of models within the UNCOVER algorithm it is essential that we can produce reliable estimates of Z. Therefore, creating a sequence of distributions differing to their neighbours only by inclusion (or exclusion) of a single observation reduces the opportunity for said distributions to be vastly dissimilar.

Similarity of neighbouring distributions is paramount to an effective SMC sampler, and as such we also insist the batches have the following form:

$$\mathfrak{B}_s = \sigma(s) \quad \text{for } s = 1, \ldots, n \tag{4.14}$$

where σ is a permutation of the set $\{1, \ldots, n\}$. This is necessary to guard against the possibility of a hidden order existing with the observation indices. A hidden order could result in 'pivot' observations — observations which differ significantly to all previous observations added with regards to their relationship with the response, and as a result the inclusion of such a pivot gives a distribution where the majority of the current set of samples have low density. Permuting the order in which observations are added then acts as an additional layer of protection against distribution dissimilarity.

Note that although we restrict the batch size to one the inclusion of an ESS condition statement allows in essence for an adaptive batch size selector within the algorithm. Indeed, as when the ESS falls below a certain threshold we resample and move, but the move is from samples generated from the last move step (when not considering their associated weights). If we had selected a batch of observations containing the observations added to the posterior between the last move step and the current iteration, then the weights for the samples would have been identical and therefore the same outcome would have been achieved. So, whilst the batch size for UNCOVER is restrictive, computational efficiency can still be leveraged through the specification of the ESS threshold ξ.

For latter iterations of UNCOVER, when the graph comprises of several components, cluster quality can still be assessed by the Bayesian evidence through first estimating the K sub-model's Bayesian evidences and then taking the product of

these values, i.e.

$$\hat{Z} = \prod_{k=1}^{K} \hat{Z}_k \tag{4.15}$$

This follows on directly from equation (4.12).

As a final note, one must take into consideration the uncertainty in estimation of the Bayesian evidence when using Sequential Monte Carlo. As we shall see in upcoming sections, selecting an edge to remove to split a cohort requires the comparison of several Bayesian evidences. Therefore, poor estimates of the Bayesian evidence could lead to a sub-optimal edge being removed, which would have a knock-on effect for the rest of the algorithm. Whilst this is clearly undesirable, for genuine clustering structure the optimal edge to remove should still be apparent as other edges, even with slight variability in estimation, are unlikely to produce an estimate for Z that exceeds the estimate for Z from the optimal edge removal. Additional measures can be taken as well, such as increasing the number of samples in the IBIS scheme. This will increase the accuracy of estimation for Z, albeit at the cost of increased computational time.

4.3 Component Generation

UNCOVER operates as a greedy algorithm due to computational efficiency, and as such can only make decisions regarding singular edges given the current state of the graph. As one would expect, these decisions are driven by the model that produces the largest Bayesian evidence. Predominately this will be through the removal of edges, as removal of an edge by definition splits a component of the graph into two components, therefore increasing the number of clusters in our model by one for each edge removed. This can be re-phrased as splitting a cluster, and given this context the idea of merging clusters can also be introduced. The merging of clusters (or components) must not impact on the structure of the covariates determined by the Minimum Spanning Tree (MST), however. Therefore we only allow clusters to be merged through the reintroduction of edges previously removed, respecting the original MST structure.

4.3.1 Edge Removal

Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$ denote the current state of the graph. Initially for the one-cluster model, due to the properties of an MST, removal of any edge from $\mathfrak{T}$ creates a Minimum Spanning Forest (MSF), with the subgraph consisting of all observation vertices and the MSF containing exactly two components. Labelling these components then determines the partition of $\{1, \ldots, n\}$ into two sets, $\mathfrak{V}_1$ and $\mathfrak{V}_2$. Creating separate Bayesian logistic regression models for each vertex set then taking the product of the resulting Bayesian evidences gives a comparative tool to compare the original one-cluster model with the two-component model. Letting $Z^{\{i,j\}^-}$ be the Bayesian evidence of the model created by removing edge $\{i, j\}$ from the current graph, we have

$$Z^{\{i,j\}^-} = Z_1^{\{i,j\}^-} \times Z_2^{\{i,j\}^-} \tag{4.16}$$

where $Z_l^{\{i,j\}^-}$ for $l = 1, 2$ represents the Bayesian evidence of the sub-model created through removal of edge $\{i, j\}$. If $Z^{\{i,j\}^-} > Z$ (where Z refers to the Bayesian evidence of the one-cluster model) then there is evidence that removal of edge $\{i, j\}$ results in a better model than the initial one-cluster model. An example of this process is given in figure 4.3.

Repeating this process for each edge in $\mathfrak{T}$ then gives the following:

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{T}} \{Z^{\{i,j\}^-}\} \tag{4.17}$$

and therefore if $Z^{\epsilon^-} > Z$ we remove ϵ from the graph to update $\mathfrak{T} = \mathfrak{T} \backslash \{\epsilon\}$ which then (in combination with all observation vertices) gives the updated subgraph $\mathcal{G}_{\mathfrak{T}}$.

For all subsequent edge removals, we will have a MSF with K components, and given the separation property of an UNCOVER model's Bayesian evidence we can view this component in isolation when considering an edge removal. In more detail, say for the model generated by current graph $\mathcal{G}_{\mathfrak{T}}$ the Bayesian evidence is $Z = \prod_{k=1}^{K} Z_k$. Removal of edge $\{i, j\}$ only affects the value of Z_k (where $i, j \in \mathfrak{V}_k$) due to the hard clustering meaning models are fitted to disjoint partitions of the

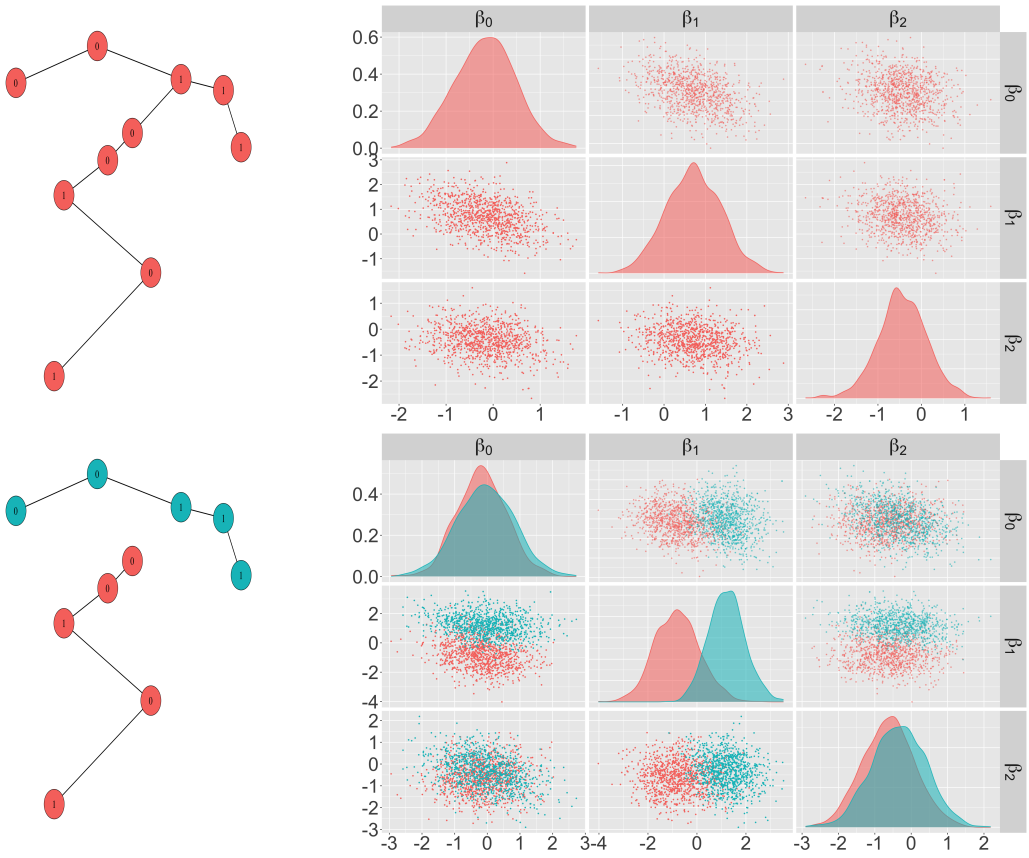


Figure 4.3: Two-dimensional data consisting of ten observations and their associated responses — shown as vertices and their corresponding labels for graph plots. The posterior samples (obtained using an iterated batch importance sampler with a standard normal prior) for each model are plotted to the right of their associated graph. Top: One-cluster model. Bottom: Two-cluster model.

data, and so we revert to the initial setting by considering component-k as a single graph, i.e.

$$Z^{\{i,j\}^-} = Z_{k1}^{\{i,j\}^-} \times Z_{k2}^{\{i,j\}^-} \times \prod_{l \neq k} Z_l \tag{4.18}$$

where here Z_k is replaced by the Bayesian evidence of the two sub-models created through the removal of edge $\{i, j\}$ ($Z_{k1}^{\{i,j\}^-}$ and $Z_{k2}^{\{i,j\}^-}$). Removal of edge $\{i, j\}$ from the current edge set $\mathfrak{T}$ then gives a new graph with $K+1$ components. This process is described in algorithm 7, where $\mathfrak{R}$ is the set of all previously removed edges. As a final remark we draw attention to the fact that it remains a possibility that no edge removal improves upon our current state. Therefore we have a natural stopping criterion for the UNCOVER algorithm.

Algorithm 7: Edge Removal

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *ESS Threshold* — ξ, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$

Step 1 : **for** $k = 1, \ldots, K$ **do**

Let $\mathfrak{T}_k = \{\{i,j\} \in \mathfrak{T} : i,j \in \mathfrak{V}_k\}$ and $\mathcal{G}_{\mathfrak{T}_k} = (\mathfrak{V}_k, \mathfrak{T}_k)$ be a subgraph of $\mathcal{G}_{\mathfrak{T}}$. **for** $\{i,j\} \in \mathfrak{T}_k$ **do**

Let $\tilde{\mathcal{G}} = (\mathfrak{V}_k = \tilde{\mathfrak{V}}_{k1} \cup \tilde{\mathfrak{V}}_{k2}, \mathfrak{T}_k \backslash \{i,j\})$ be a subgraph of $\mathcal{G}_{\mathfrak{T}_k}$. **for** $l = 1, 2$ **do**

Let $\mathfrak{B}_1, \ldots, \mathfrak{B}_{|\tilde{\mathfrak{V}}_{kl}|}$ be such that

$$\mathfrak{B}_s = \sigma(s) \quad \text{for } s \in \tilde{\mathfrak{V}}_{kl}$$

Estimate $Z_{kl}^{\{i,j\}^-}$ through algorithm 4.

end

Let

$$Z^{\{i,j\}^-} = Z_{k1}^{\{i,j\}^-} \times Z_{k2}^{\{i,j\}^-} \times \prod_{l \neq k} Z_l$$

end

end

Step 2 : Let

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{T}} \{Z^{\{i,j\}^-}\}$$

if $Z^{\epsilon^-} > Z$ **then**

Update $\mathfrak{T} = \mathfrak{T} \backslash \epsilon$ and update $Z = Z^{\epsilon^-}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K+1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \cup \epsilon$.

end

Result : $\mathcal{G}_{\mathfrak{T}}, Z, \mathfrak{R}$

4.3.2 Edge Reintroduction

The reintroduction of edges may initially seem a redundant task, as for certain edges reintroduction is always detrimental to the overall model. Indeed, at any iteration if we consider the edge just removed, reintroduction would trivially be detrimental as removal of said edge improved the previous system. This is also true for the second to last edge removed, if no edges have been reintroduced between these two removals. This is shown in Lemma 4.3.1.

Lemma 4.3.1 (Reintroduction of Recent Edges)**.** *Let* $\mathcal{G} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$ *be a minimum spanning forest which defines the posterior* $\pi(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K \mid \mathbf{y}, \mathbf{X}, \mathfrak{V})$ *with Bayesian evidence* Z*. Let* $\epsilon^\dagger = \arg\max_{\{i,j\} \in \mathfrak{T}} \{Z^{\{i,j\}^-}\}$*,* $Z^\dagger = Z^{\epsilon^{\dagger -}} > Z$ *and* $\mathcal{G}^\dagger =$

$(\bigcup_{k=1}^{K+1} \mathfrak{V}_k^\dagger, \mathfrak{T}^\dagger = \mathfrak{T} \backslash \{\epsilon^\dagger\})$. *Let* $\epsilon^{\dagger\dagger} = \arg\max_{\{i,j\} \in \mathfrak{T}^\dagger} \{(Z^\dagger)^{\{i,j\}^-}\}$, $Z^{\dagger\dagger} = (Z^\dagger)^{\epsilon^{\dagger\dagger -}} > Z^\dagger$ *and* $\mathcal{G}^{\dagger\dagger} = (\bigcup_{k=1}^{K+2} \mathfrak{V}_k^{\dagger\dagger}, \mathfrak{T}^{\dagger\dagger} = \mathfrak{T}^\dagger \backslash \{\epsilon^{\dagger\dagger}\})$. *Reintroduction of the edges* $\epsilon^\dagger$ *or* $\epsilon^{\dagger\dagger}$ *to* $\mathcal{G}^{\dagger\dagger}$ *will not result in a Bayesian evidence greater than* $Z^{\dagger\dagger}$.

Proof. Addition of $\epsilon^{\dagger\dagger}$ trivially gives the graph $\mathcal{G}^\dagger$ and therefore $Z^\dagger < Z^{\dagger\dagger}$. Reintroduction of $\epsilon^\dagger$ would give the graph $\mathcal{G}^\ddagger = (\bigcup_{k=1}^{K+1} \mathfrak{V}_k^\ddagger, \mathfrak{T}^\ddagger = \mathfrak{T}^{\dagger\dagger} \cup \epsilon^\dagger)$ with corresponding Bayesian evidence $Z^\ddagger$. Note that $\mathfrak{T}^{\dagger\dagger} \cup \epsilon^\dagger = \mathfrak{T} \backslash \{\epsilon^{\dagger\dagger}\}$ and so reintroduction of $\epsilon^{\dagger\dagger}$ is equivalent to removing $\epsilon^{\dagger\dagger}$ from $\mathcal{G}$. Now assume the contrary, that $Z^\ddagger > Z^{\dagger\dagger}$. As $Z^{\dagger\dagger} > Z^\dagger \implies Z^\ddagger > Z^\dagger$. However, as $\epsilon^\dagger = \arg\max_{\{i,j\} \in \mathfrak{T}} \{Z^{\{i,j\}^-}\}$ we must have $Z^\ddagger \leq Z^\dagger$, and so this is a contradiction. □

This is not true in general, however, due to the greedy nature of the UNCOVER algorithm, and so consideration of previously removed edges being reintroduced is necessary. For UNCOVER, this consideration is made immediately after an edge has been removed. Note that from Lemma 4.3.1 theoretically this is unnecessary for the first two edge removals of the algorithm.

Again letting the set of removed edges be $\mathfrak{R}$ and the current model's Bayesian evidence be Z, we define for the reintroduction of $\{i,j\} \in \mathfrak{R}$ which combines components k and l:

$$Z^{\{i,j\}^+} = Z_{kl}^{\{i,j\}^+} \times \prod_{a \in \{1,\ldots,K\} \backslash \{k,l\}} Z_a \tag{4.19}$$

where $Z_{kl}^{\{i,j\}^+}$ is the Bayesian evidence for the sub-posterior

$$\pi(\boldsymbol{\beta}_{kl} \mid \mathbf{y}, \mathbf{X}, \mathfrak{V}_k, \mathfrak{V}_l) = \prod_{i=1}^{n} [\pi(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}_{kl})]^{\mathbb{1}(i \in \mathfrak{V}_k \cup \mathfrak{V}_l)} \tag{4.20}$$

and $\boldsymbol{\beta}_{kl}$ is a new set of regression coefficients for the merged component. Taking

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{R}} \{Z^{\{i,j\}^+}\} \tag{4.21}$$

we reintroduce edge ϵ if $Z^{\epsilon^+} > Z$. The process of edge reintroduction is given formally in algorithms 8 and 9.

Algorithm 8: Edge Reintroduction Bayesian Evidence Generator

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *ESS Threshold* — ξ, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$

Step 1 : **for** $\{i, j\} \in \mathfrak{R}$ **do**

Let $k = \{a : i \in \mathfrak{V}_a\}$, $l = \{a : j \in \mathfrak{V}_a\}$. Let $\tilde{\mathcal{G}} = (\tilde{\mathfrak{V}} \cup \bigcup_{b \neq k,l} \mathfrak{V}_b, \mathfrak{T} \cup \{i, j\})$ where $\tilde{\mathfrak{V}} = \mathfrak{V}_k \cup \mathfrak{V}_l$. Let $\mathfrak{B}_1, \ldots, \mathfrak{B}_{|\tilde{\mathfrak{V}}|}$ be such that

$$\mathfrak{B}_s = \sigma(s) \quad \text{for } s \in \tilde{\mathfrak{V}}$$

Estimate $Z_{kl}^{\{i,j\}^+}$ through algorithm 4. Let

$$Z^{\{i,j\}^+} = Z_{kl}^{\{i,j\}^+} \times \prod_{a \neq k,l} Z_a$$

end

Result : *Bayesian Evidence* — $Z^{\{i,j\}^+}$ for $\{i, j\} \in \mathfrak{R}$

Algorithm 9: Edge Reintroduction

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *ESS Threshold* — ξ, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$

Step 1 : Obtain $Z^{\{i,j\}^+}$ for $\{i, j\} \in \mathfrak{R}$ through algorithm 8.

Step 2 : Let

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{R}} \{Z^{\{i,j\}^+}\}$$

if $Z^{\epsilon^+} > Z$ **then**

Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon$ and update $Z = Z^{\epsilon^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon$. Go to Step 1.

end

Result : $\mathcal{G}_{\mathfrak{T}}, Z, \mathfrak{R}$

4.3.3 Combination of Edge Actions

The reintroduction of edges is a corrective process which checks the system once a change has been made, in order to combat the greedy nature of the algorithm. Typically changes to the system (or graph) are made through edge removal, but edge reintroductions are also changing the graph. It is by this reasoning that if an edge is reintroduced and removed from $\mathfrak{R}$, we then must reconsider all edges still in $\mathfrak{R}$ again before considering further edge removals. Note this is again a greedy process, but necessary to ease the computational burden that arises with an exhaustive search.

Therefore, the process is as follows; the first edge actions one can make (if beneficial to do so) are to remove three edges greedily, then edges in $\mathfrak{R}$ are reintroduced greedily until it is no longer beneficial to do so. After this we return to focus on edge removals, with any edge removal triggering a greedy reintroduction of edges until it is no longer beneficial to do so. Note that this is distinct from typical pruning stages found in decision trees or mixture of expert models [30, 76] (which shall be discussed later in this chapter), as we make the corrections during the process of constructing the model. We advocate for this on the basis that the computational burden, which can be minimal for small $\mathfrak{R}$, does not outweigh the benefits one receives from reintroducing an edge which improves the system. Additionally, given the greedy nature of UNCOVER, it is likely that edges removed in a previous iteration wouldn't be removed for the current state of the graph, resulting in the computational expense of assessing removed edges for reintroduction rarely going without reward over the course of the algorithm.

4.4 Deforestation

As previously stated in section 4.3.1, through the removal of edges a natural stopping criterion occurs when there exists no singular edge removal that increases the Bayesian evidence. This follows even when edge reintroductions are included, as if no edge is removed then the graph remains unchanged, therefore all possible reintroductions have already been considered and deemed not beneficial to the model (in terms of increasing the Bayesian evidence). As a result, there is no requirement to provide additional stopping criteria for the algorithm[4].

The resulting model will be a product of at most n sub-models (though achieving this maximum is clearly an undesirable setting as it suggests severe overfitting and is seldom seen within the UNCOVER framework). Whilst use of this model is perfectly acceptable one may have additional criteria, from either the stakeholder or statistician, which requires further alteration of the model. This section covers some of the criteria that could be suggested and how to alter the outputted UNCOVER

[4]Although computationally it may be desirable to add further stopping rules.

model to achieve this criteria. An assumption is made here that any criteria set is met by the one-cluster model in order to guarantee that an output that satisfies the criteria exists.

UNCOVER employs two methods to change the current model, edge removal and edge reintroduction. In section 4.3.2 we highlighted the use of edge reintroduction as a corrective process, and so edge reintroductions are the most natural actions to take to attempt to meet the criteria set, especially due to the fact reintroduction of all edges to the graph is guaranteed to give an acceptable model. It is possible to include edge removals in this second corrective stage of UNCOVER but we omit this action. The justification of this is two-fold; the first being from a computational perspective the set of excluded edges $\mathfrak{R}$ is typically much smaller than the set of included edges and therefore faster to evaluate, and the second being that it is possible for edge removals to be a direct contrast with meeting the criteria. An example of this would be if the criterion was a maximum number of clusters allowed — here removing an edge will only increase the number of clusters and therefore not immediately help in meeting the criterion.

Only allowing edge reintroductions from a graphical viewpoint is equivalent to reducing the number of component-specific Minimum Spanning Trees (MSTs) in our minimum spanning forest, hence the second stage of UNCOVER being labelled the deforestation stage. In a similar vein, from this point forward we shall now refer to the main construction stage implemented before deforestation as the 'planting' stage.

As a final point, one may question why this criteria is not respected in the initial phase of model building, i.e. what is the purpose in creating two separate stages of UNCOVER? The reasoning for this is that the criteria can be too restrictive to properly explore the clustering structure of the data, resulting in premature termination of the algorithm. This phenomenon is similar to that of pruning decision trees or mixture of experts models [30, 76], whereby addition of a pruning stage to remove 'leaf nodes' ensures we do not underfit the data in the construction stage. A simple example of this would be to consider a two-dimensional dataset generated from three Gaussians, with each Gaussian having a different relationship between

the response and the covariates. The first few iterations of UNCOVER are shown in figure 4.4.

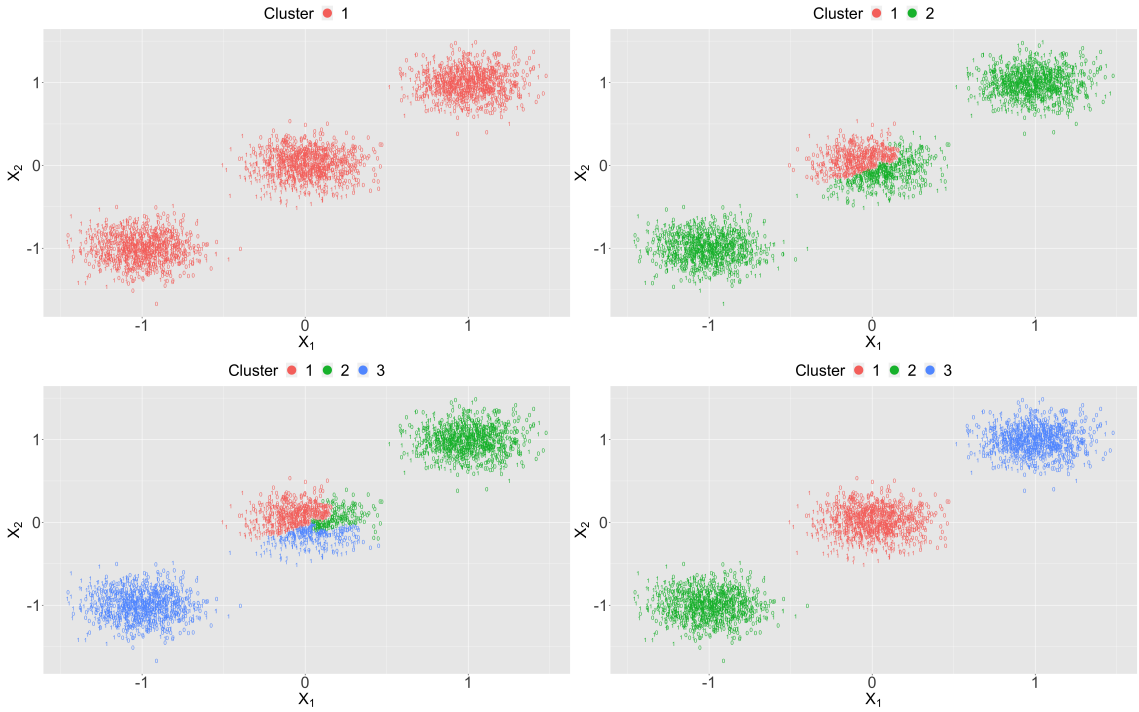


Figure 4.4: Two-dimensional simulated data consisting of three Gaussian centered at $(-1,-1)^T$, $(0,0)^T$ and $(1,1)^T$ with true regression coefficients of $(-3,-3,0)^T$, $(0,-9,-9)^T$ and $(3,-3,0)^T$ respectively. The four panels represent different iterations of an UNCOVER model with a deforestation criterion of at most three clusters in the final output. Top left is the initialisation, top right is after one edge removal, bottom left is after two edge removals and bottom right is the output after completing the planting stage followed by deforestation (specifically that a maximum of three clusters are allowed in the final output).

Assume that the criterion specified here is that there can only be at most three clusters in the outputted model, a reasonable criterion given the true clustering structure of the data. Initially UNCOVER can only remove a single edge, however. This is not guaranteed to correspond to separation of one cluster from the other two (as this is a greedy process). With this seemingly incorrect edge removed we would proceed to remove another edge in order to reveal additional structure. At this iteration we cannot remove any further edges as this would break our criterion and so we would output a sub-optimal model (bottom-left panel of figure 4.4). On the other-hand, allowing the algorithm to continue and meet the criterion in the deforestation stage would lead to the true model (bottom-right panel of figure 4.4)

being generated. This gives an indication as to the dangers of under-fitting one can encounter when not employing a two-stage algorithm.

4.4.1 Basic Criteria

The criteria which are believed to have the most practical use to stakeholders is the specification of a maximum number of cohorts or a minimum number of observations per cohort. Specification of a maximum number of cohorts could have an appeal to a stakeholder with budgetary restrictions — if intervention or action plans were to be executed for each cohort based upon their predicted response, then it may be desirable to restrict the number of cohorts if cost of intervention was high. Similarly, a stakeholder may require at least a certain number of patients (in a medical context such as SPARRA for example) to benefit from a cohort-specific intervention plan being constructed, as development of such a plan could be costly.

From a statistical point of view, employing these basic criteria offers an ad-hoc method of attempting to ensure that the clusters formed have a sufficient amount of data to realistically capture a signal between the covariates and the response. This is in no way guaranteed by these criteria, however. For example, let the criterion be a maximum of two clusters in the output — it is possible to output clusters of sizes $n-1$ and 1, where the one-observation cluster will clearly be unable to capture any true signal. Specification of a minimum cluster size also offers no guarantees, as one cluster could be of reasonable size but contain only one response type. Nevertheless, the simplicity and ease of implementation for these methods make them viable options.

Reintroduction of edges automatically reduces the number of clusters and therefore every edge reintroduced helps the model meet the maximum number of clusters criterion. Therefore, we simply add edges which result in the highest increase (or smallest decrease) of the Bayesian evidence back to the graph, until the criterion is met.

Minimum cluster size is less straightforward, as reintroduction of the most beneficial (or least detrimental) edge may not be the optimal process. Letting K' be the number of clusters which do not meet the criterion, we define $\mathfrak{R}'$ as the subset of $\mathfrak{R}$

whose edge reintroduction would decrease K'. With this we can employ the method of selecting the edge in $\mathfrak{R}$ that increases the Bayesian evidence Z by the largest amount, and if no edge reintroduction increases Z then we select the least detrimental edge from $\mathfrak{R}'$. One may wonder why we do not always select an edge from $\mathfrak{R}'$, however, note that every edge reintroduction is a step closer to an acceptable one cluster model. As a consequence, any edge reintroduction made which benefits the system also indirectly assists in meeting the criteria. The formal processes for both deforestation criteria are given in algorithms 10 and 11.

Algorithm 10: Number of Clusters Deforestation Criterion

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *ESS Threshold* — ξ, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$, *Maximum Number of Clusters Allowed* — κ

Step 1 : Obtain $Z^{\{i,j\}^+}$ for $\{i,j\} \in \mathfrak{R}$ through algorithm 8.

Step 2 : Let

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{R}} \{Z^{\{i,j\}^+}\}$$

if $K > \kappa$ **then**

Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon$ and update $Z = Z^{\epsilon^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon$. Go to Step 1.

end

Result : $\mathcal{G}_{\mathfrak{T}}, Z, \mathfrak{R}$

Note that due to the greedy nature of the algorithm, the deforestation stage may result in a model that has a lower Bayesian evidence than a model encountered during the planting stage of the algorithm that happened to meet the criterion. This obviously is not desirable and therefore for practical implementation of this type of criterion we must produce a saved state of the best model so far, each time the criterion is met during the planting stage of UNCOVER. Then the output will simply be either the deforestation output model or the saved 'best' model depending on which model has the highest Bayesian evidence.

4.4.2 Maximal Regret

When constructing our model in the planting stage of UNCOVER, consideration of how much each change of the graph improves the Bayesian evidence could be taken

Algorithm 11: Size of Clusters Deforestation Criterion

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *ESS Threshold* — ξ, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$, *Minimum Size Allowed for each Cluster* — $\bar{\kappa}$

Step 1 : Obtain $Z^{\{i,j\}^+}$ for $\{i,j\} \in \mathfrak{R}$ through algorithm 8.

Step 2 : Let $\mathfrak{R}' = \{\{i,j\} \in \mathfrak{R} : |\mathfrak{V}_k| < \bar{\kappa} \cup |\mathfrak{V}_l| < \bar{\kappa} \text{ where } i \in \mathfrak{V}_k, j \in \mathfrak{V}_l\}$. Let

$$\epsilon = \arg\max_{\{i,j\} \in \mathfrak{R}} \{Z^{\{i,j\}^+}\} \qquad \epsilon' = \arg\max_{\{i,j\} \in \mathfrak{R}'} \{Z^{\{i,j\}^+}\}$$

if $Z^{\epsilon^+} > Z$ **then**
  Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon$ and update $Z = Z^{\epsilon^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon$. Go to Step 1.
else
  if $|\mathfrak{V}_k| \geq \bar{\kappa}\ \forall\ k = 1, \ldots, K$ **then**
    Stop.
  else
    Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon'$ and update $Z = Z^{(\epsilon')^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon'$. Go to Step 1.
  end
end

Result : $\mathcal{G}_{\mathfrak{T}}, Z, \mathfrak{R}$

as opposed to the binary choice of whether the change is beneficial or detrimental. One might be cautious to split a cluster into two (which could have cost implications to the stakeholder) if the benefit of the split is minimal. In addition to this, due to the Bayesian evidence being reliant on the prior, if one has reservations on the prior specification then a more conservative approach to splitting clusters may be warranted. Letting $\bar{\nu} > 1$ be a minimum improvement factor, if an action (either edge removal or edge reintroduction) produced a model with Bayesian evidence $\tilde{Z}$, then instead of accepting the action if $\tilde{Z} > Z$ we would only accept the action if $\tilde{Z} > \bar{\nu} Z$ (where Z is the Bayesian evidence of the current model).

Given previous reservations of employing criteria during the planting stage of UNCOVER, we instead introduce the reverse of this concept in the deforestation stage. Indeed, instead of making changes which result in a minimum improvement being made, we instead specify a maximum we are willing to regret by reintroducing an edge. Letting $\nu > 1$ be the maximum regret factor, for edges $\{i,j\} \in \mathfrak{R}$ we deem

Algorithm 12: Maximal Regret Deforestation Criterion

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *ESS Threshold* — ξ, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$, *Maximum Regret Factor* — ν

Step 1 : Obtain $Z^{\{i,j\}^+}$ for $\{i,j\} \in \mathfrak{R}$ through algorithm 8.

Step 2 : Let

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{R}} \{Z^{\{i,j\}^+}\}$$

if $\nu Z^{\epsilon^+} > Z$ **then**

Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon$ and update $Z = Z^{\epsilon^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon$. Go to Step 1.

end

Result : $\mathcal{G}_{\mathfrak{T}}, Z, \mathfrak{R}$

it acceptable to reintroduce edge $\{i,j\}$ if

$$\nu Z^{\{i,j\}^+} > Z \tag{4.22}$$

Therefore, the process of edge reintroduction at the deforestation stage can be viewed as a more lenient version of algorithm 9. The algorithm for maximal regret is given as algorithm 12. In practice this criterion is more difficult to implement then the basic criteria discussed previously, as it requires knowledge of how the scale of Bayesian evidence increase transfers to increase in model quality, although interpretations such as Jeffrey's scale for Bayes factors [53] could aid in this.

4.4.3 Validation Data

The use of validation data plays a pivotal role in the tuning of hyperparameters for many statistical learning models, with examples being random forests [77], smoothing splines [30] and K-nearest neighbours [78]. Indeed, the introduction of additional data at the deforestation stage to ensure the model constructed in the planting stage has not overfit to the training data is beneficial for the creation of a generalisable output model.

There is an additional challenge in the use of validation data for UNCOVER compared to a standard training-validation set split, due to the initial construction

of the MST. One could construct the MST with all of the data and then restrict the edge removal process to only remove edges that leave at least one training observation in all clusters. This, however, would not mimic the prediction of new observations and crucially would result in the model building process having some dependence on the validation data. Therefore, the construction of the MST shall be solely based on the training data. We then add the validation data as if they were independent new observations, and therefore assign them to the cluster of their nearest training data neighbour. This process is detailed in algorithm 13, where $\mathfrak{v} \subset \{1, \ldots, n\}$ is the index set of the training data, $\mathfrak{v}^{\complement} = \{1, \ldots, n\} \backslash \mathfrak{v}$ is the index set of the validation data and $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}} = (\bigcup_{k=1}^{K} \mathfrak{V}^{\mathfrak{v}}_k, \mathfrak{T}^{\mathfrak{v}})$ is the graph obtained from applying the planting stage of UNCOVER on the training data.

Algorithm 13: Validation Data Addition

Input : *Covariate Matrix* — $\mathbf{X}$, *Training Data Index Set* — $\mathfrak{v}$,
Training Data MSF Graph — $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}} = (\bigcup_{k=1}^{K} \mathfrak{V}^{\mathfrak{v}}_k, \mathfrak{T}^{\mathfrak{v}})$, *Variable Subset* — $\mathfrak{P}$
Initialisation : Let $\mathfrak{T} = \mathfrak{T}^{\mathfrak{v}}$, $\mathfrak{v}^{\complement} = \{1, \ldots, n\} \backslash \mathfrak{v}$.
Step 2 : **for** $i \in \mathfrak{v}^{\complement}$ **do**
  Let
$$j = \arg\min_{a \in \mathfrak{v}} \{\|\mathbf{x}_{i,\mathfrak{P}} - \mathbf{x}_{a,\mathfrak{P}}\|_2\}$$
  Update $\mathfrak{T} = \mathfrak{T} \cup \{i, j\}$.
end
Step 3 : Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k = \{1, \ldots, n\}, \mathfrak{T})$.
Result : *Complete Data Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k = \{1, \ldots, n\}, \mathfrak{T})$

The Bayesian model associated with $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$ will have Bayesian evidence $Z^{\mathfrak{v}}$. Given the set of component vertex sets $\mathfrak{V} = \{\mathfrak{V}_1, \ldots, \mathfrak{V}_K\}$ from the graph $\mathcal{G}_{\mathfrak{T}}$ (obtained from algorithm 13), the measure of the model's performance will then be based upon the posterior predictive distribution

$$\pi(\mathbf{y}_{\mathfrak{v}^{\complement}} \mid \mathbf{X}_{\mathfrak{v}^{\complement}\cdot}, \mathbf{X}_{\mathfrak{v}\cdot}, \mathbf{y}_{\mathfrak{v}}, \mathfrak{V}) \coloneqq \varpi = \int \pi(\mathbf{y}_{\mathfrak{v}^{\complement}} \mid \mathcal{B}, \mathbf{X}_{\mathfrak{v}^{\complement}\cdot}, \mathfrak{V}) \pi(\mathcal{B} \mid \mathbf{X}_{\mathfrak{v}\cdot}, \mathbf{y}_{\mathfrak{v}}, \mathfrak{V}) d\mathcal{B} \tag{4.23}$$

where $\mathcal{B} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K\}$. Note that $\mathfrak{V}^{\mathfrak{v}}_k \subseteq \mathfrak{V}_k$ for $k = 1, \ldots, K$, therefore

$$\pi(\mathcal{B} \mid \mathbf{X}_{\mathfrak{v}\cdot}, \mathbf{y}_{\mathfrak{v}}, \mathfrak{V}) = \pi(\mathcal{B} \mid \mathbf{X}_{\mathfrak{v}\cdot}, \mathbf{y}_{\mathfrak{v}}, \mathfrak{V}^{\mathfrak{v}}) \tag{4.24}$$

This then allows for, in combination with the property of observations being i.i.d.,

the following reformulation of $\pi(\mathbf{y}_{\mathfrak{v}\mathfrak{C}} \mid \mathbf{X}_{\mathfrak{v}\mathfrak{C}\cdot}, \mathbf{X}_{\mathfrak{v}\cdot}, \mathbf{y}_{\mathfrak{v}}, \mathfrak{V})$ given in equation (4.23):

$$\begin{aligned}
\varpi &= \int \pi(\mathbf{y}_{\mathfrak{v}\mathfrak{C}} \mid \mathcal{B}, \mathbf{X}_{\mathfrak{v}\mathfrak{C}\cdot}, \mathfrak{V})\pi(\mathcal{B} \mid \mathbf{X}_{\mathfrak{v}\cdot}, \mathbf{y}_{\mathfrak{v}}, \mathfrak{V})d\mathcal{B} \\
&= \frac{1}{Z^{\mathfrak{v}}} \int \pi(\mathbf{y}_{\mathfrak{v}\mathfrak{C}} \mid \mathcal{B}, \mathbf{X}_{\mathfrak{v}\mathfrak{C}\cdot}, \mathfrak{V})\pi(\mathbf{y}_{\mathfrak{v}} \mid \mathcal{B}, \mathbf{X}_{\mathfrak{v}\cdot}, \mathfrak{V})\pi(\mathcal{B})d\mathcal{B} \\
&= \frac{1}{Z^{\mathfrak{v}}} \int \pi(\mathbf{y} \mid \mathcal{B}, \mathbf{X}, \mathfrak{V})\pi(\mathcal{B})d\mathcal{B} \\
&= \frac{Z}{Z^{\mathfrak{v}}}
\end{aligned} \tag{4.25}$$

where Z is the Bayesian evidence of the full posterior generated from $\mathcal{G}_{\mathfrak{T}}$. Of course in a Bayesian setting this reformulation is not necessary for the estimation of ϖ as we can simply use samples from the posterior of the training data model. As we will see in chapter 5, however, in order to improve computational speed we may not always have access to posterior samples, and therefore this form of ϖ is required. Additionally, this form allows for an algorithm structured similarly to previous deforestation algorithms, namely that of working solely with Bayesian evidences.

We can now reintroduce edges which increase ϖ instead of the Bayesian evidence, as an increase in ϖ translates to an increase in the model's ability to predict the correct response of the validation data. The formal algorithm is given as algorithm 14.

Note that without restriction, a model which overfits in the planting stage (likely resulting in small clusters) is likely to produce clusters which have no validation data attached in the deforestation stage. This clearly is harmful as due to the independence of clusters we gain no insight into the generalisability of the small cluster sub-model. To remedy this, we insist that any edge removed in the planting stage of UNCOVER must not result in a cluster without any validation data attached. Formally, we apply algorithm 13 to the initial one-cluster graph to obtain $\mathcal{G}_{\mathfrak{T}}$. Then we only consider the removal of edge $\{i, j\}$ from the training data graph $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$ if the resulting graph $(\bigcup_{k=1}^{K+1} \mathfrak{V}_k, \mathfrak{T} \backslash (\mathfrak{R} \cup \{i, j\}))$ has the following property:

$$\mathfrak{V}_k \nsubseteq \mathfrak{v} \qquad \text{for } k = 1, \ldots, K+1 \tag{4.26}$$

Algorithm 14: Validation Data Deforestation Criterion

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *Variable Subset* — $\mathfrak{P}$, *Training Data Index Set* — $\mathfrak{v}$, *Training Data MSF Graph* — $\mathcal{G}_{\mathfrak{T}^{\mathfrak{v}}}^{\mathfrak{v}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k^{\mathfrak{v}}, \mathfrak{T}^{\mathfrak{v}})$, *ESS Threshold* — ξ, *Training Data Model's Bayesian Evidence* — $Z^{\mathfrak{v}}$, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$

Step 1 : Obtain $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k = \{1, \ldots, n\}, \mathfrak{T})$ from algorithm 13. From the resulting $\mathfrak{V}_1, \ldots, \mathfrak{V}_K$ estimate $Z = \prod_{k=1}^{K} Z_k$ using K applications of algorithm 4.

Step 2 : Let

$$\varpi = \frac{Z}{Z^{\mathfrak{v}}}$$

Step 3 : Obtain $(Z^{\mathfrak{v}})^{\{i,j\}^+}$ and $Z^{\{i,j\}^+}$ for each $\{i, j\} \in \mathfrak{R}$ through algorithm 8. Let

$$\varpi^{\{i,j\}^+} = \frac{Z^{\{i,j\}^+}}{(Z^{\mathfrak{v}})^{\{i,j\}^+}}$$

Step 4 : Let

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{R}} \{\varpi^{\{i,j\}^+}\}$$

if $\varpi^{\epsilon^+} > \varpi$ **then**

Update $\mathfrak{T}^{\mathfrak{v}} = \mathfrak{T}^{\mathfrak{v}} \cup \epsilon$, $\mathfrak{T} = \mathfrak{T} \cup \epsilon$, $Z^{\mathfrak{v}} = (Z^{\mathfrak{v}})^{\epsilon^+}$ and $Z = Z^{\epsilon^+}$. Let $\mathcal{G}_{\mathfrak{T}^{\mathfrak{v}}}^{\mathfrak{v}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k^{\mathfrak{v}}, \mathfrak{T}^{\mathfrak{v}})$ and $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraphs. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon$. Go to Step 1.

end

Result : $\mathcal{G}_{\mathfrak{T}^{\mathfrak{v}}}^{\mathfrak{v}}, Z^{\mathfrak{v}}, \mathfrak{R}$, *Complete Data Graph* — $\mathcal{G}_{\mathfrak{T}}$, *Complete Data Model's Bayesian Evidence* — Z

This method of deforestation offers an almost automatic selection process, with the only choice being that of $\mathfrak{v}$. Typically the selection of training data is random, with a split parameter $o \in [0, 1]$ determining the proportion of data used to train the model. Selection of o is not an easy choice, however. Higher values of o gives more assurance that the structure of the covariate data is captured by the training data but could lead to poor generalisability. Alternatively, low values of o could give better generalisability and less restrictions on the choice of edges in the planting stage, but then lead to a model unable to capture the true clustering structure.

As a final note, it is important to highlight the work done previously on generalisability of Bayesian evidences (or marginal likelihoods) by Lofti et.al [79]. Here they discuss the potential pitfalls of using solely the Bayesian evidence as a quality measure and introduce the Conditional Marginal Likelihood (CML) as an alterna-

tive. The CML is equivalent to our validation data set-up in the one-cluster model. Whilst not specifically focused on clustering problems and the challenges that come with providing validation data for K sub-models, the recognition of validation data as a viable method of model selection with Bayesian evidences gives a strong justification for use of this method in practice.

4.4.4 Response Diversity

In a frequentist framework, as discussed previously in chapter 3, a misleading phenomenon about the quality of a cluster's sub-model occurs when we partition the data such that one cluster contains only one response type — the sub-model typically degenerates, and we obtain an output of this one response type regardless of the input. This fits the training data perfectly but is likely not to generalise well, therefore UNCOVER is designed to alleviate this issue through the use of Bayesian priors to restrict the model from making one class predictions with certainty.

However, whilst we ensure probabilistic predictions with UNCOVER, it is still possible to produce a one response type cluster. Given these clusters are undesirable and not particularly informative, a reasonable criterion to set (to avoid such clusters being present in the final output) is that the number of unique responses for observations in a cluster is greater than one.

Insistence on only one observation response being in the minority class may not be a strong enough criterion to ensure generalisability of the sub-models, and so we introduce a minimum minority factor $\upsilon \in \{1, \ldots, n^\dagger\}$ such that:

$$\upsilon \leq \sum_{i \in \mathfrak{V}_k} y_i \leq |\mathfrak{V}_k| - \upsilon \quad \text{for } k = 1, \ldots, K \tag{4.27}$$

where $\mathfrak{V}_k$ are the vertex sets of the K clusters in the model and $n^\dagger$ is the number of observations in the total dataset which have an associated response in the minority class (i.e. $n^\dagger \leq \frac{n}{2}$). One must make a considered choice for υ, as υ indirectly restricts the number of clusters in the final model. To see this, note that if $\upsilon = \frac{n^\dagger}{K}$ then a model with $K + 1$ clusters automatically breaks the criterion as there will exist a cluster with less than $\frac{n^\dagger}{K}$ observations with response in the overall minority class.

The deforestation method for a diverse response in each cluster follows a similar process to that of a minimum size for each cluster. Indeed, given this similarity it is also worth noting that as with the size of clusters criterion it is possible we achieve a better model at some iteration of the planting stage of UNCOVER than at the end of the deforestation stage, and so a saved optimal criterion fitting model at the planting stage is also required here. The diverse response criterion could even be said to fall within the umbrella term of basic deforestation criteria, though we make the distinction here as the basic criteria mentioned in subsection 4.4.1 are of interest primarily to the stakeholder, whereas a diverse response has appeal mainly from a modelling perspective. The formal algorithm is given in algorithm 15.

Algorithm 15: Diverse Response Deforestation Criterion

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *ESS Threshold* — ξ, *Graph* — $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K} \mathfrak{V}_k, \mathfrak{T})$, *Bayesian Evidence* — Z, *Number of Samples* — N, *Removed Edge Set* — $\mathfrak{R}$, *Minimum Count of Minority Class Responses Allowed for each Cluster* — υ

Step 1 : Obtain $Z^{\{i,j\}^+}$ for $\{i,j\} \in \mathfrak{R}$ through algorithm 8.

Step 2 : Let $\mathfrak{R}' = \{\{i,j\} \in \mathfrak{R} : (\upsilon > \sum_{a \in \mathfrak{V}_k} y_a \cup \upsilon > |\mathfrak{V}_k| - \sum_{a \in \mathfrak{V}_k} y_a) \cup (\upsilon > \sum_{b \in \mathfrak{V}_l} y_b \cup \upsilon > |\mathfrak{V}_l| - \sum_{b \in \mathfrak{V}_l} y_b)$ where $i \in \mathfrak{V}_k, j \in \mathfrak{V}_l\}$. Let

$$\epsilon = \arg \max_{\{i,j\} \in \mathfrak{R}} \{Z^{\{i,j\}^+}\} \qquad \epsilon' = \arg \max_{\{i,j\} \in \mathfrak{R}'} \{Z^{\{i,j\}^+}\}$$

if $Z^{\epsilon^+} > Z$ **then**
  Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon$ and update $Z = Z^{\epsilon^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon$. Go to Step 1.
else
  if $\upsilon \leq \sum_{a \in \mathfrak{V}_k} y_a \leq |\mathfrak{V}_k| - \upsilon \ \forall \ k = 1, \ldots, K$ **then**
    Stop.
  else
    Update $\mathfrak{T} = \mathfrak{T} \cup \epsilon'$ and update $Z = Z^{(\epsilon')^+}$. Let $\mathcal{G}_{\mathfrak{T}} = (\bigcup_{k=1}^{K-1} \mathfrak{V}_k, \mathfrak{T})$ be the updated subgraph. Update $\mathfrak{R} = \mathfrak{R} \backslash \epsilon'$. Go to Step 1.
  end
end

Result : $\mathcal{G}_{\mathfrak{T}}, Z, \mathfrak{R}$

4.4.5 Summary

Deforestation criteria can be used for either practical or theoretical purposes. If real-world restrictions such as budgets influences the form of the desired model for the stakeholder, this can be met through basic criteria such as a maximum number of clusters. If generalisability of the model to new data is of concern, then other deforestation criteria such as the use of validation data may be more appropriate. There is of course overlap between these two goals of practicality and model quality, with different criteria often resulting in the same output. There is also no restriction on the number of criteria used in a single UNCOVER algorithm, for example it is perfectly reasonable to require a maximum number of clusters, all of which have at least a certain number of observations with response in the minority class. In conclusion, deforestation is intended as a flexible framework for model selection within the UNCOVER context in order to meet a pre-specified criterion or criteria.

4.5 The UNCOVER Algorithm

Algorithms 16, 17 and 18 provide the general UNCOVER method, namely initialisation, the planting stage and the deforestation stage. Note here that the algorithm employs a stopping criterion, $\varkappa$, at the planting stage in the form of number of clusters. This is in contrast to previous explanations of the dangers of stopping the algorithm in the planting stage, however, in a practical setting one must balance exploration against compute time. Therefore, whilst it is advised to set $\varkappa \in \{1, \ldots, n\}$ as large as possible, for large data problems a lower value of $\varkappa$ may be more suitable (see appendix B.1 for more details).

Note that the deforestation variable input can be one of $(\kappa, \text{ϟ}, \nu, o, \upsilon)$ depending on the deforestation criterion. Additionally, for the basic or diverse response deforestation criteria, the graph $\mathcal{G}^{\mathfrak{v}}_{\tilde{\mathfrak{V}}^{\mathfrak{v}}}$ meeting the specific criteria is as previously discussed in 4.4, i.e.

- Number of Clusters — $\mathcal{G}^{\mathfrak{v}}_{\tilde{\mathfrak{V}}^{\mathfrak{v}}}$ has $K \leq \kappa$ components.
- Size of Clusters — The partition of the vertex set of $\mathcal{G}^{\mathfrak{v}}_{\tilde{\mathfrak{V}}^{\mathfrak{v}}}$, $\mathfrak{V}^{\mathfrak{v}} = \bigcup_{k=1}^{K} \mathfrak{V}^{\mathfrak{v}}_1, \ldots, \mathfrak{V}^{\mathfrak{v}}_K$,

Algorithm 16: UNCOVER — Initialisation

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *Deforestation Variable (Validation Criterion Only)* — o, *Variable Subset* — $\mathfrak{P}$

Step 1 : **if** *Validation Data Deforestation Criterion* **then**
| Randomly select $o \times n$ observation indices to form $\mathfrak{v}$.
else
| Let $\mathfrak{v} = \{1, \ldots, n\}$.
end

Step 2 : Obtain the complete graph from the Euclidean distance matrix of $\mathbf{X}_{\mathfrak{v},\mathfrak{P}}$.

Step 3 : Obtain the minimum spanning tree edge-induced subgraph $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$ from algorithm 6.

Step 4 : **if** *Validation Data Deforestation Criterion* **then**
| Obtain $\mathcal{G}_{\mathfrak{T}} = (\mathfrak{V} = \{1, \ldots, n\}, \mathfrak{T})$ from algorithm 13.
else
| Let $\mathcal{G}_{\mathfrak{T}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$
end

Result : *Training Data Minimum Spanning Tree Graph* — $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, *Complete Data Minimum Spanning Tree Graph* — $\mathcal{G}_{\mathfrak{T}}$, *Training Data Index Set* — $\mathfrak{v}$

into the K components of the graph is such that $|\mathfrak{V}^{\mathfrak{v}}_k| \geq \mathfrak{K}$ for $k = 1, \ldots, K$.

- Diverse Response — The partition of the vertex set of $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $\mathfrak{V}^{\mathfrak{v}} = \bigcup_{k=1}^{K} \mathfrak{V}^{\mathfrak{v}}_1, \ldots, \mathfrak{V}^{\mathfrak{v}}_K$, into the K components of the graph is such that $\upsilon \leq \sum_{i \in \mathfrak{V}^{\mathfrak{v}}_k} y_i \leq |\mathfrak{V}^{\mathfrak{v}}_k| - \upsilon$ for $k = 1, \ldots, K$.

Additionally, one may notice that the algorithms detailing UNCOVER have a final output of just the training data graph, and not posterior samples. This is due to the iterated batch importance sampling algorithm (4) being replaceable with other Bayesian evidence approximation algorithms (see section 5.4) which may not require posterior samples. Note, the graph is the key component here as it defines the clusters, which defines the model, which then ultimately defines the posterior. Samples from this posterior (which may be required for prediction) can then be obtained through K implementations of algorithm 4, which gives K sets of weighted samples which can combined to achieve B and W, where; B is an $N \times K(p+1)$ matrix such that $\beta_{r,(k-1)(p+1)+i}$ is the r^{th} regression coefficient sample from the k^{th} cluster for variable $i-1$ and W is an $N \times K(p+1)$ associated weight matrix.

Algorithm 17: UNCOVER — Planting Stage

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$, *Deforestation Variable*, *Training Data MST Graph* — $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, *Complete Data MST Graph* — $\mathcal{G}_{\mathfrak{T}}$, *ESS Threshold* — ξ, *Number of Samples* — N, *Stopping Criterion* — $\varkappa$

Initialisation : Let $\mathfrak{R} = \emptyset$.

Step 1 : Obtain $Z^{\mathfrak{v}}$ from algorithm 4. Let $\mathcal{G}^{\text{best}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $Z^{\text{best}} = Z^{\mathfrak{v}}$.

Step 2 : Let $\tilde{Z} = Z^{\mathfrak{v}}$. **if** *Validation Data Response Criterion* **then**
  Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 7, with the alteration that edges should not be considered for removal if their removal in $\mathcal{G}_{\mathfrak{T}}$ leaves components containing no validation observations.
else
  Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 7. **if** *Basic or Diverse Response Deforestation Criteria* **then**
    If $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$ meets the specific criterion then let $\mathcal{G}^{\text{best}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $Z^{\text{best}} = Z^{\mathfrak{v}}$.
  end
end

Step 3 : **if** $Z^{\mathfrak{v}} = \tilde{Z}$ **then**
  Stop.
else
  Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 9. **if** *Basic or Diverse Response Deforestation Criteria* **then**
    If $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$ meets the specific criterion then let $\mathcal{G}^{\text{best}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, $Z^{\text{best}} = Z^{\mathfrak{v}}$.
  end
end

Step 4 : **if** $K = \varkappa$ **then**
  Stop.
else
  Go to step 2.
end

Result : $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$, *Training Data Model's Bayesian Evidence* — $Z^{\mathfrak{v}}$, *Removed Edge Set* — $\mathfrak{R}$, *Best Criteria Meeting Graph* —$\mathcal{G}^{\text{best}}$, *Best Criteria Meeting Model's Bayesian Evidence* — Z^{best}

4.6 Simulated Example

In this section we aim to highlight the capabilities of the UNCOVER method compared to established methods on a challenging simulated example. This dataset is comprised of spiral data, with a clustering set-up comprised such that knowledge of the response is required alongside the structure of the data in covariate space to fully understand the true clusters present.

In order to compare the methods, we require metrics of both cluster assignment

Algorithm 18: UNCOVER — Deforestation Stage

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$,
Deforestation Variable, *Training Data MSF Graph* — $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$,
Training Data Model's Bayesian Evidence — $Z^{\mathfrak{v}}$, *Removed Edge Set* — $\mathfrak{R}$,
Training Data Index Set — $\mathfrak{v}$, *ESS Threshold* — ξ, *Variable Subset* — $\mathfrak{P}$,
Number of Samples — N
$\star$ Basic or Diverse Criteria Only: *Best Criteria Meeting Graph* —$\mathcal{G}^{\text{best}}$,
Best Criteria Meeting Model's Bayesian Evidence — Z^{best}

Step 1 : **if** *Basic or Diverse Response Deforestation Criteria* **then**
  if *Number of Clusters Deforestation Criterion* **then**
    Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}, Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 10.
  end
  if *Size of Clusters Deforestation Criterion* **then**
    Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}, Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 11.
  end
  if *Diverse Response Deforestation Criterion* **then**
    Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}, Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 15.
  end
  if $Z^{\mathfrak{v}} \geq Z^{best}$ **then**
    Let $\mathcal{G}^{\text{out}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$.
  else
    Let $\mathcal{G}^{\text{out}} = \mathcal{G}^{\text{best}}$.
  end
end
Step 2 : **if** *Maximal Regret Deforestation Criterion* **then**
  Update $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}, Z^{\mathfrak{v}}$ and $\mathfrak{R}$ through algorithm 12. Let $\mathcal{G}^{\text{out}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$.
end
Step 3 : **if** *Validation Data Deforestation Criterion* **then**
  Update; $\mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}, Z^{\mathfrak{v}}$ and $\mathfrak{R}$ and obtain; $\mathcal{G}_{\mathfrak{T}}$ and Z through algorithm 14. Let $\mathcal{G}^{\text{out}} = \mathcal{G}^{\mathfrak{v}}_{\mathfrak{T}^{\mathfrak{v}}}$.
end
Result : *Final MSF Graph* — $\mathcal{G}^{\text{out}}$

and predictive power. Both the metrics we use in this analysis will be based on elements of the confusion matrix for binary outputs, given in table 4.1. For prediction, the output is a binary response of success (i.e $Y = 1$) or failure (i.e. $Y = 0$) and so a true positive for training observation i occurs when the model predicts a success (i.e. $\hat{y}_i = 1$) and the actual response for this observation is a success (i.e. $y_i = 1$). This logic naturally extends to explain the meaning behind the other elements of the confusion matrix. In terms of cluster assignment, we consider pairwise similarity of observations as our output. For example, with training observations i and j, if i and j are predicted to have the same cluster assignment this is a positive and otherwise

is a negative. If the actual cluster assignment places the two observations in the same cluster and the predicted cluster assignment does as well then this is a true positive result. So one could view the confusion matrix for prediction as the sum of n individual matrices for each of the training observations, and for cluster assignment as the sum of $\binom{n}{2}$ individuals matrices for each of the pairwise comparisons of the training observations. Note that a pairwise comparison confusion matrix is applied here instead of a simpler metric such as the number of individual observations matching their actual cluster, as the arbitrary labelling of clusters results potentially misleading results. As a simple example of where individual metrics fail, consider a clustering which has partitioned the observations into the correct clusters. Depending on the label chosen, this clustering will be measured as either completely correct or completely incorrect.

	Predicted Output	
Actual Output	True Positive (TP)	False Negative (FN)
	False Positive (FP)	True Negative (TN)

Table 4.1: Confusion Matrix.

In terms of predictive power, a standard utilisation of the confusion matrix is the AUC — Area Under the Receiver Operating Characteristic (ROC) Curve [80]. The ROC curve is achieved by plotting the True Positive Rate (TPR)

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.28}$$

against the False Positive Rate (FPR)

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{4.29}$$

at different thresholds between 0 and 1 (where for threshold a an observation's predicted response is classified as a success 1 or failure 0 depending on whether the associated predicted probability of success is above or below a). Note that prediction of success for UNCOVER is achieved through the posterior predictive distribution which gives an estimate of the probability of success. The AUC is simply the area under this curve. Based on the assumption that it is desirable to achieve a high TPR

and low FPR regardless of the threshold we set for binary predictions, the AUC will achieve maximum value of 1 if this is the case. If the opposite is true then the AUC will achieve the minimum of 0[5]. For assignment of success probabilities completely at random, we would expect TPR = FPR regardless of the threshold and as such the AUC = 0.5.

For cluster assignment, we utilise the elements of the confusion matrix through the Fowlkes–Mallows index (FMI) [81]:

$$\sqrt{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \times \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}} \tag{4.30}$$

The FMI is a combination of two measures; the first being given we assign two observations to the same cluster how often are the observations actually in the same cluster $\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\right)$ and the second being given two observations are actually in the same cluster how often do we assign them to the same cluster $\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\right)$. Naturally for a suitable clustering both measures will be high. The FMI is also largely governed by the number of true positives as opposed to the number of true negatives, which is justifiable from a clustering perspective as a pair of observations both being correctly assigned to the same cluster contains much more information than both observations correctly being assigned to different clusters.

An output which performs well for both measures will then represent an ideal model for stakeholders. Established methods typically excel in one of these measures, but not both.

Note that for unsupervised methods such as K-means or hierarchical clustering which produce a hard clustering output we apply sequential predictive modelling to obtain a separate model for each of the clusters. The choice of model is flexible and so we opt to build Bayesian logistic regression models here, with the same justification as given for their use in UNCOVER. The default prior will be a standard normal, and new observations are assigned to the cluster of their nearest training data observation.

[5]Though this case may suggest an encoding issue as a switch of the classes results in the perfect model.

4.6.1 Spirals

The covariate data $\mathbf{X} \in \mathbb{R}^{4000\times 2}$ will be simulated as follows:

$$x_{i1} \sim \begin{cases} \frac{-4(i-1)\cos\left(\frac{4\pi(i-1)}{1999}\right)}{3\times 1999} + \mathcal{U}(-0.05, 0.05) & \text{if } i \in \{1, \ldots, 2000\} \\ \frac{4(i-2001)\cos\left(\frac{4\pi(i-2001)}{1999}\right)}{3\times 1999} + \mathcal{U}(-0.05, 0.05) & \text{if } i \in \{2001, \ldots, 4000\} \end{cases} \tag{4.31}$$

$$x_{i2} \sim \begin{cases} \frac{-4(i-1)\sin\left(\frac{4\pi(i-1)}{1999}\right)}{3\times 1999} + \mathcal{U}(-0.05, 0.05) & \text{if } i \in \{1, \ldots, 2000\} \\ \frac{4(i-2001)\sin\left(\frac{4\pi(i-2001)}{1999}\right)}{3\times 1999} + \mathcal{U}(-0.05, 0.05) & \text{if } i \in \{2001, \ldots, 4000\} \end{cases} \tag{4.32}$$

This data produces two spirals contained within the hypercube $[0, 2] \times [0, 2]$. With this covariate data, we define four clusters, with specification of the regression coefficients and response as follows:

$$\boldsymbol{\beta}_1 = (0, -8, 3)^T \tag{4.33}$$

$$\boldsymbol{\beta}_2 = (0, 5, -9)^T \tag{4.34}$$

$$\boldsymbol{\beta}_3 = (0, 10, 5)^T \tag{4.35}$$

$$\boldsymbol{\beta}_4 = (0, -14, 9)^T \tag{4.36}$$

$$y_i \sim \begin{cases} \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_1})^{-1}) & \text{if } i \in \{1, \ldots, 1000\} \\ \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_2})^{-1}) & \text{if } i \in \{1001, \ldots, 2000\} \\ \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_3})^{-1}) & \text{if } i \in \{2001, \ldots, 3000\} \\ \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_4})^{-1}) & \text{if } i \in \{3001, \ldots, 4000\} \end{cases} \tag{4.37}$$

The resulting complete dataset can be visualised in figure 4.5. Note that from this dataset an 80 : 20 split of the data is taken for each cluster (which each cluster containing 1000 observations) to obtain training and test datasets.

The non-linear structure of the data presents challenges for methods such as Mixture of Experts (MoEs) and K-means, as the clusters are not linearly separable. Additionally, as some clusters are not separated in the covariate space, reliance on solely the covariates for cluster generation will lead to unsatisfactory results. This can be showcased by applying the methods of K-means, hierarchical clustering,

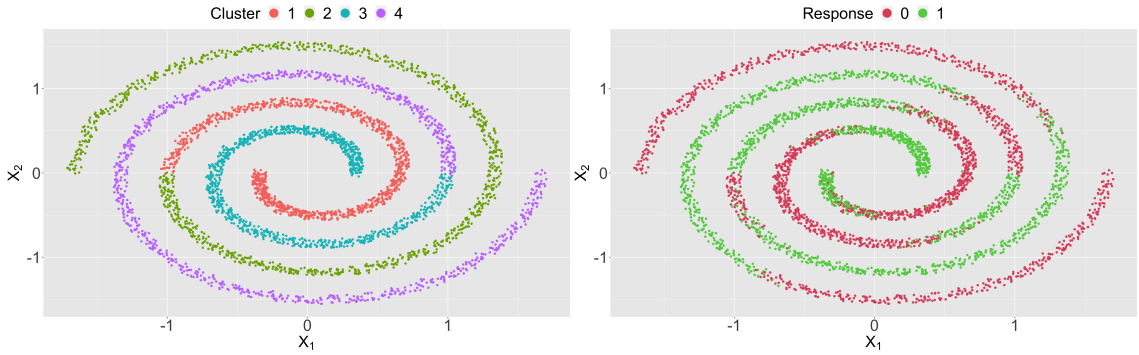


Figure 4.5: Spiral Dataset. The left plot shows the covariate data with their associated true clusters (shown through the colouring) and the right plot shows the covariate data with their associated responses (shown through the colouring).

Finite Mixtures of Logistic Regression (FMLR) models and MoEs to the dataset. Note that as the true number of clusters is known here, we pre-specify for all methods that the outputted number of clusters $K = 4$. The results[6] are shown in table 4.2.

Method	**Train**		**Test**	
	FMI	**AUC**	**FMI**	**AUC**
K-means	0.2516276	0.6735	0.2546817	0.6839
HC–SL	0.6772377	0.7649	0.6751498	0.7672
HC–CL	0.2625201	0.6937	0.2622268	0.6903
HC–AL	0.2606972	0.6796	0.256761	0.6677
FMLR	NA	0.7011	NA	0.6842
MoE	0.298163	0.8586	0.3027186	0.8705
HMoE	0.2727915	0.7312	0.2695054	0.7238

Table 4.2: Performance metrics for established methods on the spirals dataset.

Clearly from a cohort detection perspective HC–SL has the best performance, although even this methods clusters do not resemble the true clustering, as is evident from the predictive performance of HC–SL. On the other hand, MoE had (unsurprisingly) the best predictive performance, but at the cost of not being able to recognise the non-linear cohort structure in the data, resulting in poor FMI values.

In order to examine UNCOVER's performance as a whole on this dataset we must first provide an analysis on the applicability of each of the deforestation criteria for this type of problem. For consistency we shall keep all other parameters fixed for

[6]The abbreviations that have not been previously specified are as follows: Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), Hierarchical Clustering — Average Linkage (HC–AL) and Hierarchical Mixture of Experts (HMoE).

each run of UNCOVER, that being the number of samples $N = 1000$, effective sample size threshold $\xi = \frac{N}{2}$ and the stopping criterion $\varkappa = 10$. We begin with the number of clusters deforestation criterion, whose results for various 'maximum number of clusters' thresholds can be seen in figure 4.6. Here we can see that any

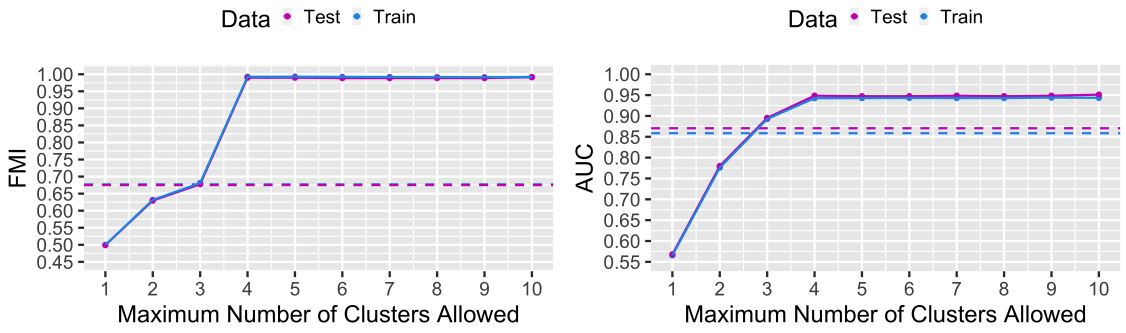


Figure 4.6: UNCOVER performance metrics on the spiral dataset when the 'number of clusters' deforestation criterion is specified . The metrics FMI (left) and AUC (right) are shown for an increasing maximum number of clusters being allowed in the final output. Dashed lines in each plot shows the maximum value obtained by previous methods for both the training data (blue) and the test data (purple). For FMI the dashed lines represent single linkage hierarchical clustering and for AUC the dashed lines represent one level mixture of experts.

specification of the maximum number of clusters allowed (κ) above 2 outperforms the previous methods both in terms of cohort detection and in predictive modelling. However, it must be noted that all 10 runs of UNCOVER attained their maximum number of allowed clusters in the output. Indeed, for $\kappa > 4$ it was deemed beneficial to remove a small number[7] of observations which belong to a large cluster but whose responses appears contradictory to the regression signal present in the large cluster. Whilst a fair comparison of UNCOVER with this deforestation criterion would be to consider the scenario when $\kappa = 4$, which does not suffer this small cluster issue[8], in an unknown cluster setting the potential creation of small clusters due to an overfitting of the data is undesirable.

A natural solution to this is the next deforestation criterion considered, setting a minimum cluster size. The results of this criterion for various minimum sizes are

[7]This is evident through observing the FMI values for the training data, which do not greatly fall for large values of κ as only small numbers of observations are affected.

[8]The smallest cluster size for $\kappa = 4$ is 791.

seen in figure 4.7. Here we can see a similar story to that of the maximum number

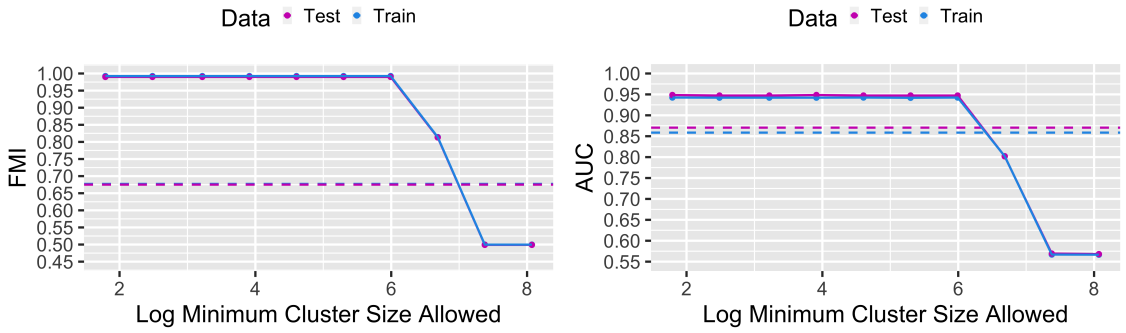


Figure 4.7: UNCOVER performance metrics on the spiral dataset when the 'size of clusters' deforestation criterion is specified. The metrics FMI (left) and AUC (right) are shown for an increasing minimum cluster size threshold. Dashed lines in each plot shows the maximum value obtained by previous methods for both the training data (blue) and the test data (purple). For FMI the dashed lines represent single linkage hierarchical clustering and for AUC the dashed lines represent one level mixture of experts. The natural logarithm of the minimum cluster sizes are shown, with the actual sizes used being $6, 12, 25, 50, 100, 200, 400, 800, 1600$ and 3200.

of clusters criterion, mainly that with a reasonable specification we can out-perform all non-UNCOVER methods tested previously. Interestingly, unlike the maximum number of clusters criterion, the number of clusters does not increase as the cluster size restriction decreases, such that even at the generous threshold of 6 observations UNCOVER returns four clusters resembling the true cluster structure. This perhaps suggests that in unknown cluster settings the minimum cluster size criterion should be preferred, as it mitigates UNCOVER's propensity to create small observation clusters to further enhance the Bayesian evidence of the larger clusters. This does not necessarily invalidate the number of clusters criterion, however. In practical settings in may be more natural for stakeholders to place restrictions on the number of cohorts as opposed to the size of cohorts in the training data. We also note that for a cluster size of 800 (the logarithm of which is 6.68) we see an unusual drop in predictive performance despite the model still being able to theoretically retrieve the four true clusters. This is due to the overlap of observations that occur between clusters 1 and 2 and between clusters 3 and 4, making it impossible to obtain a perfect true cluster separation through UNCOVER. The issue of overlapping is discussed in section 6.1.

Instead of controlling the size of the outputted clusters to mitigate the overfitting issue, one could control the diversity in the response. This naturally leads to the response diversity deforestation criterion, in which we specify for all clusters a minimum number of observations whose associated response is in that cluster's minority class — υ. The simplest case is when $\upsilon = 1$, ensuring that every cluster has at least one observation with a response of 1 and at least one observation with a response of 0 but for larger values of υ we can gain a greater confidence that the outputted clusters are capturing a signal within the data instead of defining an area of the space where all training observations have a similar response. The results of this criterion on the spiral dataset for differing υ is given in figure 4.8. Here we can see

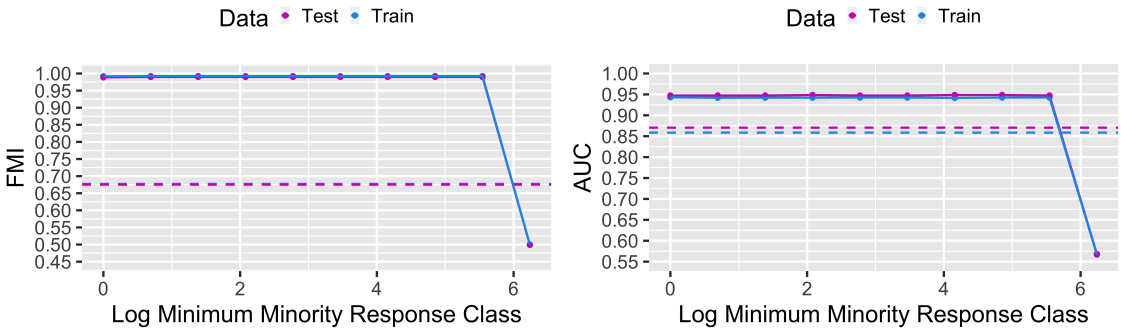


Figure 4.8: UNCOVER performance metrics on the spiral dataset when the response diversity deforestation criterion is specified. The metrics FMI (left) and AUC (right) are shown for an increasing minimum minority response class threshold. Dashed lines in each plot shows the maximum value obtained by previous methods for both the training data (blue) and the test data (purple). For FMI the dashed lines represent single linkage hierarchical clustering and for AUC the dashed lines represent one level mixture of experts. The natural logarithm of the minimum minority response class thresholds are shown, with the actual values used being 2^j for $j = 0, \ldots, 9$.

that as expected we can outperform the non-UNCOVER methods previously tested with relative ease, and in fact for $\upsilon > 1$ we remove the overfitting issue. Whilst this criterion therefore appears relatively robust and simple to specify, one must note that is possible to encounter settings where small overfit clusters appear with $\upsilon \geq 2$, and in general specification of the number of minority response classes which give an acceptable regression signal may be difficult and require careful consideration of the balance of responses in the overall data. Finally for this criterion, we note that extreme specifications of υ typically result in a collapse to a one-cluster model, as

is the case here when $\upsilon = 2^9$.

The next deforestation criterion we consider is maximal regret. As previously stated in section 4.4.2, specification of the maximal regret factor is difficult to obtain without knowledge of how the Bayesian evidence behaves for this specific problem. In light of this, we specify a generous range of thresholds — 3^j for $j = 1, \ldots, 10$, which for $j = 10$ allows combinations of clusters even when the resulting Bayesian evidence is almost 60000 times worse than the current Bayesian evidence. The results of evaluating these thresholds are given in figure 4.9. Even at the extremes

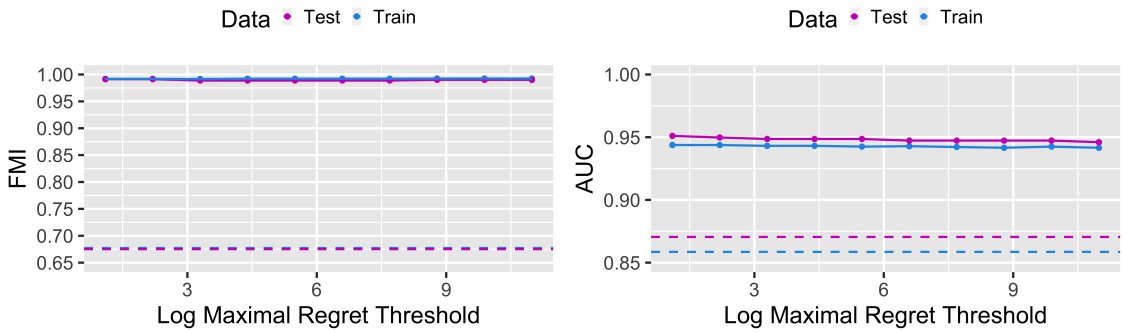


Figure 4.9: UNCOVER performance metrics on the spiral dataset when the maximal regret deforestation criterion is specified. The metrics FMI (left) and AUC (right) are shown for an increasing maximal regret threshold. Dashed lines in each plot shows the maximum value obtained by previous methods for both the training data (blue) and the test data (purple). For FMI the dashed lines represent single linkage hierarchical clustering and for AUC the dashed lines represent one level mixture of experts. The natural logarithm of the maximal regret thresholds are shown, with the actual values used being 3^j for $j = 1, \ldots, 10$.

of our given range of thresholds, the resulting output performs well in both cohort detection and predictive modelling. For the maximal regret parameter $\nu = 3$ we revert back to the overfitting issue where the threshold is not restrictive enough to prevent small clusters being formed to accommodate large clusters. However, from $\nu = 3^8$ onwards we consistently produce four large clusters which resemble the true clustering. Interestingly, even for seemingly large values of maximal regret, combining any of the four large clusters is seen as too detrimental an action to take. Indeed, the maximal regret criterion is powerful due to the fact that overfitting (which UNCOVER is sometimes prone to) only results in small gains in the Bayesian evidence, as opposed to the much larger gains UNCOVER makes when creating

clusters that generalise well. The result of which is a seemingly robust method when one has enough knowledge of the problem to specify a threshold.

The final criterion we consider is the validation data criterion, the seemingly most autonomous of the deforestation criteria. Whilst a train:validation split is all that is required, the size of the training data must still be specified through the proportion parameter $o \in [0, 1]$. Therefore, we assess the effect o has on the output from UNCOVER with this criterion. Note that we must also assess the variability of outputs for certain thresholds as UNCOVER is governed by the covariate structure found in the training data, which clearly is dependent not only on the size of the training data but also the individual observations within the training data. Assessment of the variability of output is determined through several runs at particular thresholds. The results are shown in figure 4.10. The first noteworthy element of

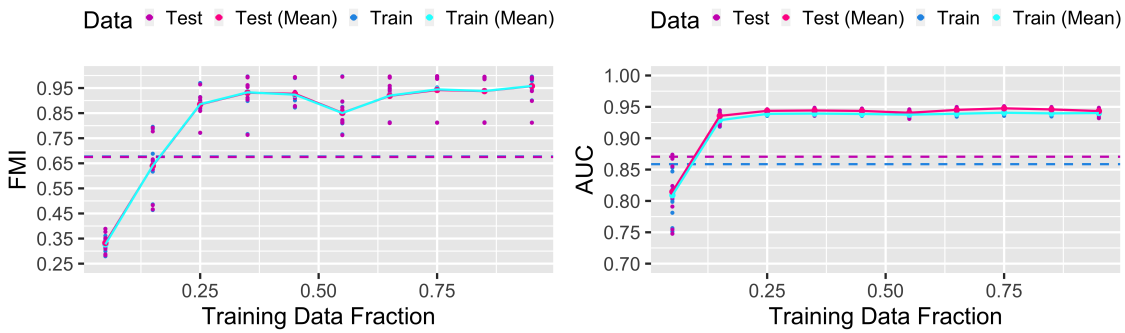


Figure 4.10: UNCOVER performance metrics on the spiral dataset when the validation data deforestation criterion is specified. The metrics FMI (left) and AUC (right) are shown for an increasing fraction of the data assigned as training data, each with multiple runs (10). Individual run results for this method's training data and test data are given as blue and purple points, and their mean results are given as cyan and pink respectively. Dashed lines in each plot shows the maximum value obtained by previous methods for both the training data (blue) and the test data (purple). For previous methods, the training data consisted of both the training data and validation data for this method. For FMI the dashed lines represent single linkage hierarchical clustering and for AUC the dashed lines represent one level mixture of experts.

figure 4.10 is that when only a small amount of observations are assigned as training data, the results are poor. In terms of FMI values, this is due to the small amount of training observations not being able to capture the spiral structure in covariate space. Therefore, the resulting minimum spanning tree will not contain the nec-

essary edges to create the four true clusters through edge removal. Regarding the AUC values, whilst it is possible to capture the true clustering signal through the production of additional clusters[9], the small amount of observations results in weak signals between the response and the covariates and so the strong true cluster signals are not able to be uncovered in the planting stage, leading to a poor predictive performance.

For higher training data fractions these problems are seemingly mitigated, as there is a sufficient amount of data to capture the covariate structure as well as the true clustering signals. However, figure 4.10 does not reveal whether this particular criterion is robust to overfitting. Figure 4.11 addresses this question through visualisation of the number of clusters and smallest cluster sizes of the various individual runs. Here we see that whilst it appears that as the training fraction increases we

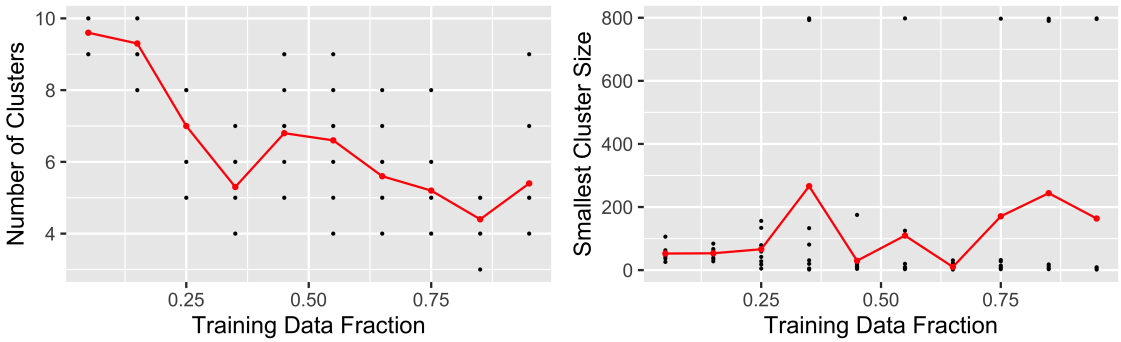


Figure 4.11: UNCOVER cluster information on the spiral dataset when the validation data deforestation criterion is specified. The number of clusters (left) and smallest cluster size (right) of the various runs are shown for an increasing fraction of the data assigned as training data, each with multiple runs (10). Individual run results are given as black points, and their mean results are given as red points. Note that when determining the smallest cluster size, both training and validation data are considered.

output fewer clusters, the variability in the runs for larger training data fractions is high. This can also be deduced from examination of the smallest cluster sizes, as it appears that although larger training set fractions allow for the possibility of large cluster sizes that one might expect (i.e. a smallest cluster size near 800 and a high

[9]If a true cluster is disconnected by the minimum spanning tree graph, with sufficient data UNCOVER will create multiple clusters for the one true cluster, each with the same regression coefficients.

FMI value indicate the correct clusters were selected in this setting), these fractions also allow the possibility of small clusters which overfit the data. This is likely the result of clusters only requiring the assignment of a single validation observation. Naturally, smaller clusters are much more likely to only be represented by a single validation observation, which therefore can lead to the potentially misleading conclusion that the small cluster generalises well due to its prediction of a single observation's response. Note that this scenario is more common for large training fractions as a result of the scarcity of validation observations. Indeed, for small sized training data (in comparison to the validation data) there is an abundance of data to test generalisability and so this small cluster issue is softened[10]. The caveat to this analysis, however, is that for large amounts of data one should have more confidence in capturing the structure in the covariates and the cluster signals even with a small training data fraction. As a result, we have more flexibility to lower the fraction to increase the possibility that all clusters uncovered have a sufficient amount of validation data attached to thoroughly test the generalisability of the clusters.

As a final point, one may wonder why it is not insisted upon that the overall training fraction is upheld for each individual cluster or why we do not insist more than one validation observation is attached to any cluster formed. The reasoning for the former point is that it is not trivial to enforce, as this insistence would clearly have a large impact on which (if any) training data graph edges would be suitable to remove, leading in some cases to one-cluster models despite a clear clustering structure. The latter point also can lead to edge eligibility policy that is too restrictive whilst simultaneously introducing another parameter which is difficult to specify for a given problem.

In conclusion for the deforestation criteria, it has been shown that in this example, for the majority of specifications, UNCOVER outperforms established methods — even in the scenario where the true number of clusters for these established methods are known. In addition to this, each criterion has been evidenced as a viable

[10]For training data fraction $o = 0.05$, the smallest cluster size out of all runs was 35.

choice to address the overfitting issue, with the selection of the criteria dependent of the specific needs of the stakeholder. Basic criteria is a clear choice for a stakeholder requiring interpretable cohorts which can be actioned upon. If the aim is generalisability of the model in the presence of test data, criteria such as the validation criterion, maximal regret or response diversity are appropriate choices, with each criterion coming with their own requirements. Validation data requires a large number of observations, maximal regret requires knowledge of the behaviour of the Bayesian evidence and response diversity requires knowledge of the balance of the responses[11].

What is common in all versions of UNCOVER, however, is the reliance on the Bayesian evidence, with its subsequent reliance on the choice of prior. Given the Bayesian evidence's sensitivity to the prior [50], one may question how UNCOVER performs for different priors. To examine this behaviour, we select a deforestation criterion which gave near perfect results, that being the minimum size of any cluster being greater or equal than 400 observations. We then shall select nine multivariate normal priors representing all combinations of three choices of mean and three choices of variance. The means are $(0, 0, 0)^T$, $(0, -2, 2)^T$ and $(0, 10, 5)^T$ — the standard option, a mean close to all true cluster's regression coefficients and a mean representing a single cluster's true coefficients, namely $\boldsymbol{\beta}_3$. The variances take the structure of the identity matrix $\mathcal{I}_3$ multiplied by a constant, either 1, 16 or 64. The increasing spread gives a vaguer prior, but one more likely to produce samples close to all true regression coefficients.

Figure 4.12 shows the results. Regarding the log Bayesian evidence, we can see that when the variance becomes more diffuse the final output does not differ significantly. This is due to the fact that the prior is less informative, and given the deforestation criterion forces each cluster produced to consist of a large number of observations, this allows the likelihood to have a dominating effect[12]. In contrast, when the prior is more concentrated the position of the prior mean has greater im-

[11] General recommendations for all deforestation criteria are given in appendix B.1.

[12] This dominating effect need not occur due to diffuse priors however. An increase in the number of training observations would also cause less reliance on the prior.

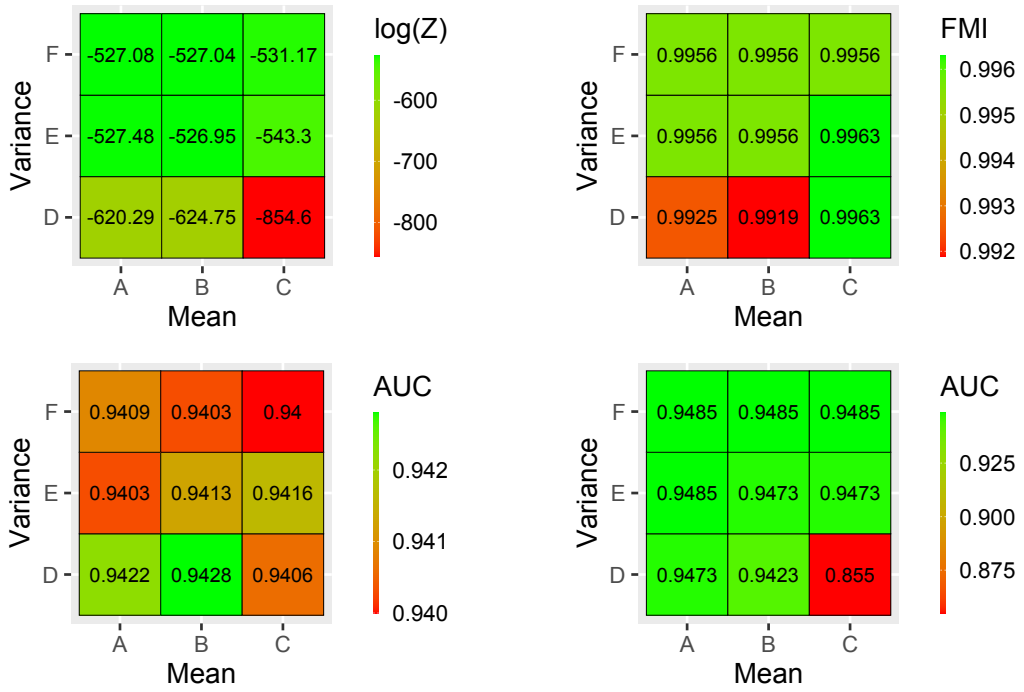


Figure 4.12: Heatmaps of various metrics of the UNCOVER algorithm output at differing prior specifications. These are; the natural logarithm of the Bayesian evidence (top left), the FMI values (top right) and the AUC values (bottom). The factors A, B and C refer to the means $(0,0,0)^T$, $(0,-2,2)^T$ and $(0,10,5)^T$ respectively. The factors D, E and F refer to the variances $\mathcal{I}_3$, $16\mathcal{I}_3$ and $64\mathcal{I}_3$ respectively. All outputs bar the bottom right heatmap (which refers to the test data) refer to the training data.

pact, as seen for the prior $\mathcal{N}((0,10,5)^T, \mathcal{I}_3)$. This prior's location and spread makes it difficult to produce samples close to the true coefficients of some clusters (even with the number of training observations being greater or equal to 400), resulting in the poor Bayesian evidence.

An interesting advantage of ill-placed priors, however, is their ability to restrict overfitting. In more detail, consider a common scenario with UNCOVER, that being a large cluster benefiting from removing a small number of observations that in reality are part of the large cluster. A well-placed prior ensures that this small number of observations are explained relatively well as their own cluster, and so they are removed. An ill placed prior on the other hand explains these observations extremely poorly (as the number is small and so the prior has a dominating effect) and so UNCOVER decides not to remove them. In essence poorly chosen priors ensure that only clusters with large amounts of observations are formed as for these

clusters the prior has an insignificant effect. The caveat to this is of course that an ill-chosen prior may miss key clustering structure and restricts the user if there is genuine prior knowledge on the potential region of cluster's regression coefficients.

The effects of ill-chosen priors for overfitting cannot be fully seen for this example due to the deforestation criterion but potentially can be partially seen in the FMI values. Indeed, whilst the FMI values for all priors are relatively similar, the greatest FMI values are reserved for the worst placed prior. This is a consequence of the edge connecting the two spirals being positioned such that a few observations belonging to one cluster are seen as more beneficial to another cluster with a well specified prior. For the poorly specified prior more relevance is placed on each cluster having a large number of observations and so the removal of those observations from one cluster is ultimately not seen as preferential.

Regarding the AUC, the bottom left image in figure 4.12 shows that for the training data the AUC values are extremely similar but in general the prior centered at $(0, -2, 2)^T$ performs better, as expected. The test data AUC is slightly more informative, however, showcasing that for a concentrated ill centered prior, predictive performance can be lower as UNCOVER struggles to produce posteriors near the true coefficients.

As mentioned above, due to the dataset size and the choice of deforestation criterion (which ensures each cluster contains at least 400 observations), the likelihood has a dominating effect over the prior. For smaller sized datasets the prior is much less likely to be dominated by the likelihood resulting in the choice of prior potentially having a larger impact.

To showcase this, we construct two new datasets — one containing 2000 observations and one containing 400 observations. Both datasets are generated in the same manner as the original spirals dataset, i.e. through equations (4.31 — 4.37), with the key difference being the number of observations in each spiral and the number of observations in each cluster. Each spiral will contain $\frac{n}{2}$ observations and each true cluster will contain $\frac{n}{4}$ observations, with n being either 2000 or 400 depending on the dataset. The minimum cluster size $\aleph$ will also require adjustment due to the new dataset size. So, for the 2000-observation dataset $\aleph = 200$ and for the

400-observation dataset $\daleth = 40$. With this set-up, we shall assess the effect of the prior using the same nine priors as used with the full dataset.

For the 2000-observation dataset, the results of the nine priors are shown in figure 4.13. The results are similar to the results found with the original dataset,

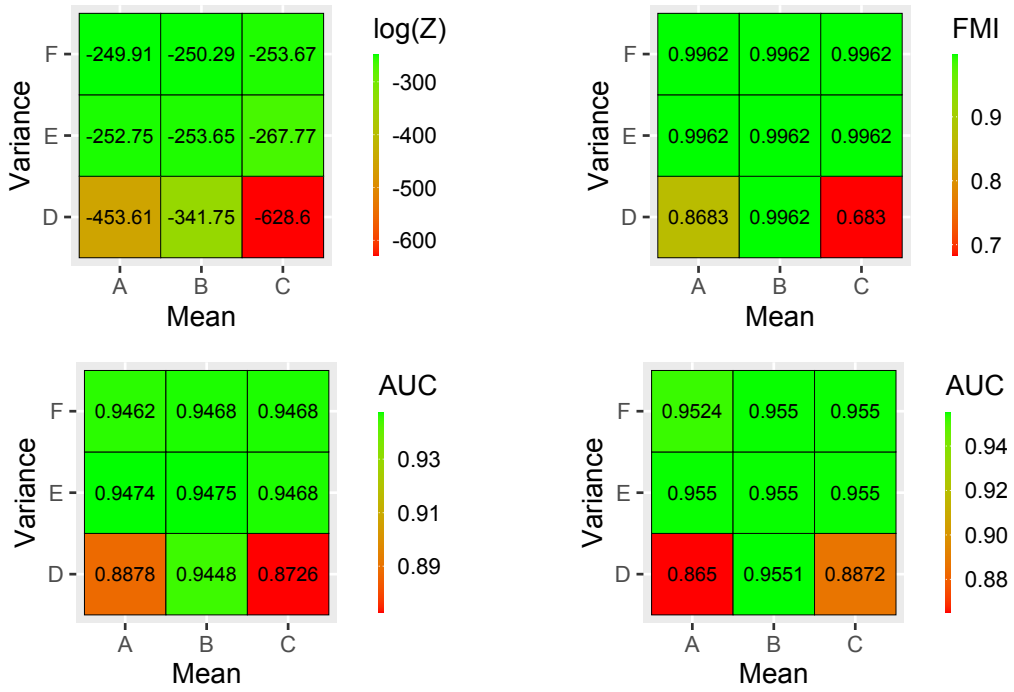


Figure 4.13: Heatmaps of various metrics of the UNCOVER algorithm output at differing prior specifications, for the 2000-observation spiral dataset. These are; the natural logarithm of the Bayesian evidence (top left), the FMI values (top right) and the AUC values (bottom). The factors A, B and C refer to the means $(0,0,0)^T$, $(0,-2,2)^T$ and $(0,10,5)^T$ respectively. The factors D, E and F refer to the variances $\mathcal{I}_3$, $16\mathcal{I}_3$ and $64\mathcal{I}_3$ respectively. All outputs bar the bottom right heatmap (which refers to the test data) refer to the training data.

however, there are two specific priors which lead to poor results, namely the priors $\mathcal{N}\left((0,0,0)^T, \mathcal{I}_3\right)$ and $\mathcal{N}\left((0,10,5)^T, \mathcal{I}_3\right)$. Indeed, both these priors have a lower Bayesian evidence, FMI value and AUC value (for both training and test data) when compared to priors with a well-placed prior mean or even diffuse ill-placed priors. Additionally, the prior $\mathcal{N}\left((0,10,5)^T, \mathcal{I}_3\right)$ failed to even produce 4 clusters as an output (a 3 cluster model was selected instead). This is a consequence of the smaller sample size, where if a prior is concentrated on an ill-placed prior mean the number of observations cannot correct for this by allowing the likelihood to dominate the prior. It should be noted however that the well-specified prior mean

(for any prior variance specification) and the ill-specified prior means with diffuse prior variances all performed well in spite of the reduced number of observations.

For the 400-observation dataset, the results of the nine priors are shown in figure 4.14. When the dataset is this size the true clusters do not contain enough obser-

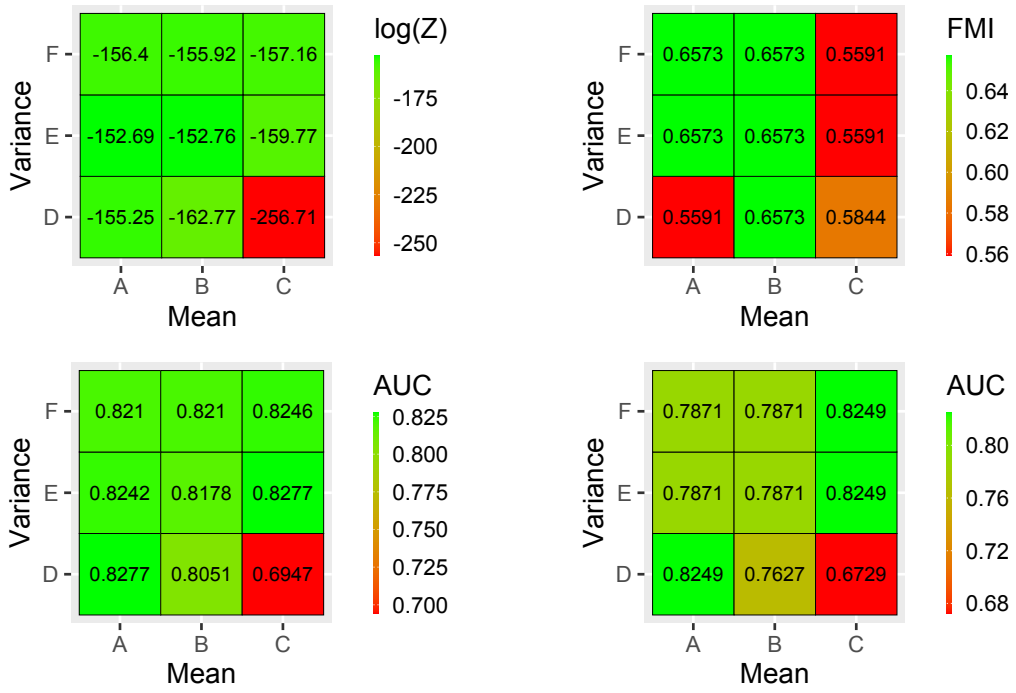


Figure 4.14: Heatmaps of various metrics of the UNCOVER algorithm output at differing prior specifications, for the 400-observation spiral dataset. These are; the natural logarithm of the Bayesian evidence (top left), the FMI values (top right) and the AUC values (bottom). The factors A, B and C refer to the means $(0, 0, 0)^T$, $(0, -2, 2)^T$ and $(0, 10, 5)^T$ respectively. The factors D, E and F refer to the variances $\mathcal{I}_3$, $16\mathcal{I}_3$ and $64\mathcal{I}_3$ respectively. All outputs bar the bottom right heatmap (which refers to the test data) refer to the training data.

vations to present a strong regression signal. As a consequence, the optimal cluster assignment for UNCOVER is unlikely to be the true clustering, and the prior mean $(0, -2, 2)^T$ may not be as well placed here as it was for the larger spiral datasets. Indeed, with weaker signals present, the standard prior mean option has more appeal and performs stronger than previously seen. Despite it's placement in covariate space being less of an asset, the prior mean $(0, -2, 2)^T$ also performs well, as does the prior mean $(0, 10, 5)^T$ when the prior variance is sufficiently diffuse. For these priors,

one of two 3-cluster solutions is selected[13], both of which have a similar performance in terms of Bayesian evidence and AUC values for training and test data. Obviously as the cluster assignments differ the FMI values will vary slightly, but as we would not expect UNCOVER to return the true clustering here the FMI values have less relevance. The only prior yet to be discussed is the prior with mean $(0, 10, 5)^T$ and variance $\mathcal{I}_3$. This prior performs poorly as the prior mean is still ill-placed here and the sample size of the dataset is such that this poorly chosen prior has a large impact on the Bayesian evidence and subsequently the cluster assignment (this prior is the only prior selected to produce a 2-cluster output). As stated previously, making the prior more diffuse can help alleviate the choice of prior mean and in this specific setting allows the output of UNCOVER to return to a better performing 3-cluster solution.

In summary, for optimal results one should opt for a prior well centered and diffuse enough to be able to capture the true cluster's coefficients with relative ease. This of course is by no means a trivial task, but prior knowledge of the problem and possible clustering structure can help with specification (see appendix B.1 for more details). This being said, all prior choices in the original spiral dataset resulted in the correct number of clusters being specified and so with a deforestation criterion such as minimum number of observations, which encourages posteriors to rely less on the prior, even ill specified priors can perform well with a large sized dataset.

4.7 Summary

UNCOVER offers an output that contains many features designed for stakeholder interpretability. Examples of this are; the hard clustering output achieved through the graphical representation of data, the flexible deforestation criteria to match stakeholder needs and the use of covariate structure through minimum spanning forests (which ensures observations with vastly different attributes are unlikely to belong to the same cohort). Guiding edge removal and reintroduction through the

[13]Which output is selected depends on both the prior chosen and the variation in estimation of the Bayesian evidence.

model's Bayesian evidence also allows for the relationship between the covariates and the response to govern the formation of the cohorts, offering an improvement upon unsupervised methods. UNCOVER also offers various additional benefits, such as; the ability to return a single cluster if no clustering structure is present, the lack of pre-specified parameters which are unlikely to be known a priori (e.g. number of clusters) and the ability to dictate the clustering structure through a sub-selection of the covariates whilst simultaneously allowing all covariates to contribute to the model.

CHAPTER 5

Implementation of UNCOVER

With interpretability forming a core principle for UNCOVER (and indeed this thesis in general), a thorough consideration of the practicalities of implementation must be carried out. Specifically, this refers to the computational bottleneck of implementing multiple Sequential Monte Carlo (SMC) samplers. In principle UNCOVER relies on a new run of an SMC sampler for each model considered (in order to obtain the Bayesian evidence of said model), so the cost of comparing models becomes extremely large if naïvely implemented. This is evident even in the first iteration of UNCOVER, which requires $n-1$ posteriors to be sampled in order to discover the optimal split of the initial one cluster model. Whilst this is a large improvement on the $2^{n-1}-1$ posteriors that would have to be sampled in an exhaustive search not utilising minimum spanning trees, this number of posteriors is still significant for large n. Additionally, the greedy nature of the algorithm does ensure that the amount of subsequent model generations diminishes as the algorithm progresses, as even with a blunt recalculation of all possible models through edge removal with a K-cluster current model the number of generations is $n-K \leq n-1$. Edge reintroduction further compounds the number of SMC runs required, however, as does the type of deforestation criterion applied.

The result in any case is that a significant amount of computation time must be devoted to the vast amount of posteriors to be sampled through SMC, and so it is vital that this bottleneck is addressed in order to allow UNCOVER to feasibly be used on real-world datasets. A standard approach one may take is to abandon the use of SMC samplers entirely or attempt to use them sparingly (this will be discussed in section 5.4), however, the benefits of using SMC make it a desirable method for UNCOVER. Indeed, frequentist methods such as information criteria do not provide samples from the posterior for obvious reasons and come with large data assumptions that cannot always be met. Other methods which do provide posterior samples such as Markov Chain Monte Carlo [37] often cannot provide the Bayesian evidence easily, which is clearly not desirable for UNCOVER. Therefore, given that SMC samplers provide all the necessary information required for UNCOVER, before consideration of alternative approaches we shall take measures to ensure the use of SMC in UNCOVER is as efficient as possible.

What follows in this chapter is the approach made to help achieve a computationally efficient version of UNCOVER using SMC samplers, namely using memoisation and reverse iterated batch importance sampling. The viability of alternatives to SMC are then discussed in the form of replacing SMC with asymptotic approximations for large clusters. Incorporation of these techniques then results in the final contribution to this chapter, an `R` package which provides a user-friendly method for using UNCOVER on real data.

5.1 Memoisation

Whilst it is true that the UNCOVER algorithm requires the generation of a vast amount of different models for comparison, it is also true that the majority of models produced will be extremely similar to a previously generated model, as they are usually nested with respect to which observations are included. Consider as a simple example figure 5.1. Removal of the edge highlighted in blue gives the two vertex sets $\mathfrak{V}_1 = \{2\}$ and $\mathfrak{V}_2 = \{1, 3, \ldots, 10\}$. If we were then to remove the edge highlighted in green we would obtain the vertex sets $\mathfrak{V}'_1 = \{2, 9\}$ and $\mathfrak{V}'_2 = \{1, 3, \ldots, 8, 10\}$.

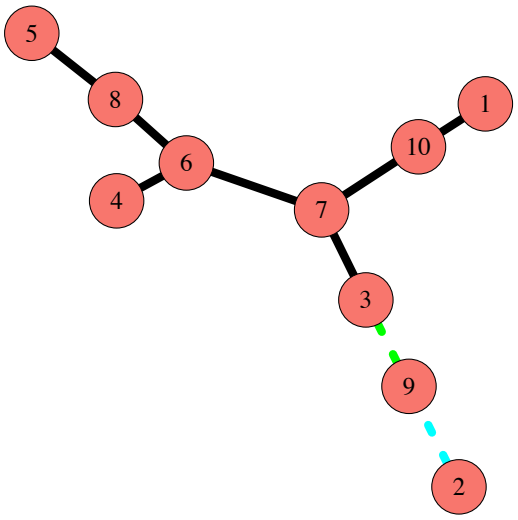


Figure 5.1: Minimum spanning tree of ten samples of $\mathcal{N}((0,0)^T, \mathcal{I}_2)$. Vertex labels correspond to the index of the observation. The majority of edges are given as black lines, with the exceptions being the green and blue dashed lines, used to highlight edges discussed in chapter 5.

Clearly the two partitions lead to similar vertex sets and therefore similar posteriors.

This similarity of models can be a major asset when using Chopin's Iterated Batch Importance Sampling (IBIS) method [9], as if the final posterior of model A contains observations that are a subset of observations that form the posterior of model B, then model A's posterior is a bridging distribution of model B's posterior. Therefore, storage of previous runs of a sequential Monte Carlo sampler (namely the weighted samples and the Bayesian evidence) could have potentially large computational savings, since model B's posterior simply requires further data updating of model A's posterior.

As such, we require a mechanism for both efficient storage and retrieval of IBIS sample sets from intermediate posteriors. In order to introduce this to UNCOVER we must first introduce two concepts, that of a cache and function memoisation.

Definition 5.1.1 (Cache)**.** *A cache is a storage object. Objects added to the cache are assigned a unique key such that these objects can be accessed without computation if the correct key is specified.*

Definition 5.1.2 (Function Memoisation)**.** *A memoised function is a function with a local cache such that when the function is called the output is stored in the cache with a unique key based on the function arguments. If the function is called again with the same arguments then the results of the first call are given without computation.*

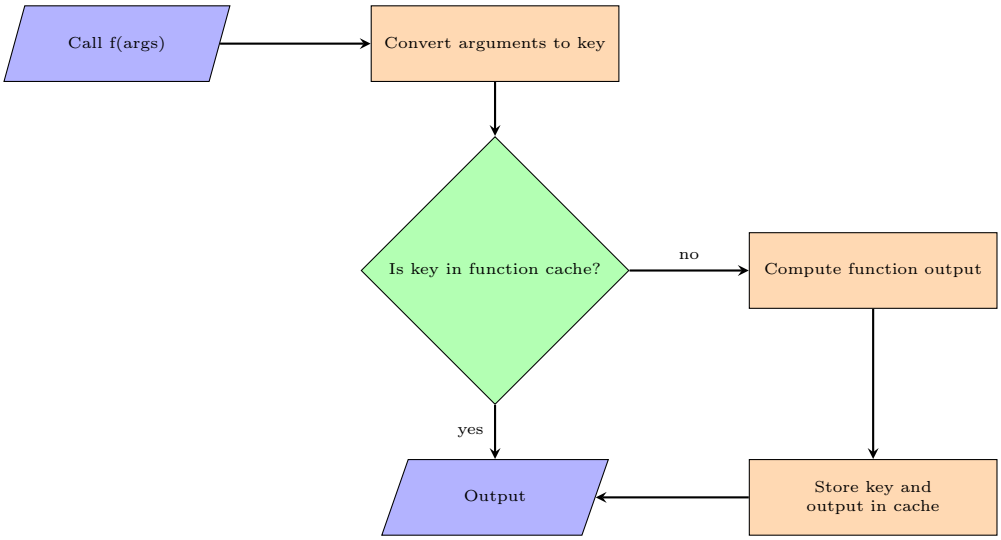


Figure 5.2: Flowchart detailing the memoisation process for a function.

It is important to note here that the cache does not possess infinite storage, and when the cache becomes full an eviction policy [82] may be required to retain the most important objects. Eviction policies determine which object to remove when a new object needs to be stored, and the policy adopted by the memoised functions in UNCOVER is Least Recently Used (LRU). The LRU policy time stamps cache objects when an action is performed, that being either when they enter the cache or when the object is called from the cache (for function memoisation this is when the function is called with arguments matching the object's key). When an object must be evicted due to insufficient space to store a new result, the object with the oldest time stamp is removed. This encourages the most useful objects to stay in the cache, where usefulness is determined by how recently the object has been used by the user.

A key feature of function memoisation is that the arguments of the function are stored (as a key) within the function's cache, meaning that when the function is called with specific arguments, only one check has to be made — if the argument

key is in the cache or not. In order to utilise memoisation for UNCOVER, however, we must introduce a new form of key lookup — for a specified argument $\mathfrak{A}$ (being the set of observation indices in the target posterior), we want to identify the object in the cache whose argument is the largest subset of $\mathfrak{A}$. This check is required as the output (being posterior samples and the Bayesian evidence) from the largest subset of $\mathfrak{A}$ (in the cache) can then be instantly accessed and subsequently used as a bridging distribution to obtain the target posterior.

The non-memoised version of IBIS used in UNCOVER requires the addition of $|\mathfrak{A}|$ batches, but for a subset $\mathfrak{A}' \subseteq \mathfrak{A}$ which is in the cache the memoised version of IBIS used in UNCOVER only needs to compute the addition of $|\mathfrak{A} \backslash \mathfrak{A}'|$ batches. Naturally when $\mathfrak{A}' = \mathfrak{A}$ no evaluation of the IBIS function is necessary and we revert back to standard memoisation after the cache check.

5.1.1 Look After the Pounds and the Pennies Look After Themselves — Cache Management

The computational gains of memoisation in this setting is not guaranteed, as the savings we make in not running the IBIS function from scratch might be offset by the time taken to discover the largest subset in the cache. It would appear a balance must be struck between having too large a cache, meaning that the cache checks are more time consuming than running the function from scratch, and too small a cache, meaning that checks are fast but rarely find a useful subset.

However, there is another option, that of an evaluation policy[1]. An evaluation policy is a system for determining when it is beneficial to check the cache and when it is beneficial to run the function from scratch. Letting $\mathfrak{A}$ be the target observation index set, we propose the evaluation policy be to only check the cache when

$$|\mathfrak{A}| \geq \rho \tag{5.1}$$

where $\rho \in \{1, \ldots, n\}$ is the evaluation threshold. The intuition behind this policy is

[1]Note that 'evaluation policy' is distinct from 'eviction policy', which is always LRU for UNCOVER.

that if $\mathfrak{A}$ is small enough than the amount of time spent checking the cache, finding a bridging distribution, and then running the remainder of the function will likely be more than simply running the function from scratch. However, if $\mathfrak{A}$ is deemed sufficiently large[2] then it is deemed worthwhile to 'take a risk' on checking the cache to find a large subset, as if one exists the computation saved could be substantial. An examination of the computational efficiency of UNCOVER for varying values of ρ is given in section 5.2.1.

5.1.2 Eviction Policy Optimisation

With only a finite cache option available one may wonder how detrimental object eviction can be to the UNCOVER algorithm. The LRU eviction policy is a generally efficient policy to promote useful subsets being kept in the cache, but this policy is only as effective as the model selection scheme used in UNCOVER. Explaining further, for a given iteration of UNCOVER, all edge removals must be considered; if these edge removals are selected at random, the sub-models observation index sets are not likely to contain significant overlap and therefore this presents an inefficient method. Figure 5.1 gives a prime example of this, as if the edge $\{2, 9\}$ was selected originally then selection of $\{5, 8\}$ as the next edge removal is inefficient, as neither of the two sets this forms are a subset of index sets in the cache.

If we can traverse the graph in an optimal manner, we can ensure that we are likely to be successful in finding an effective subset when checking the cache, whilst simultaneously evicting objects from the cache that are unlikely to be required in the future. The latter point can also be highlighted through figure 5.1, as removing edge $\{2, 9\}$ creates sub-models valuable to the removal of $\{3, 9\}$, but if $\{3, 9\}$ is the last edge removal considered then the models created by removing edge $\{2, 9\}$ may have been evicted. Graph traversal can be achieved in many ways [83], but we opt for creating an order of edges by first considering the diameter path (see section 3.3.1), and then subsequent branches encountered on this path in a sequential manner. This algorithm is given as algorithm 19. Note edges are added to the queue in

[2]Sufficiency here is determined by many factors such as general computation time of the function, cache size etc.

batches but the order of the edges within a batch can be random. The result is $\mathfrak{p}$, which is an ordered version of the Minimum Spanning Tree (MST). As algorithm 19 requires an MST as an edge set, this is applied only on the initial one-cluster graph in UNCOVER, with the ordering remaining fixed thereafter.

Algorithm 19: Depth First Search of a Minimum Spanning Tree

Input : *Minimum Spanning Tree Graph* — $\mathcal{G}_{\mathfrak{T}} = (\mathfrak{V}, \mathfrak{T})$
Initialisation : Let $\mathfrak{p} = \emptyset$. Get the diameter path $\mathfrak{p}^{\star}$. Let $\mathfrak{p}_1^{\star} = \{i, j\}$. Add i to $\tilde{\mathfrak{V}}$. Let $\mathcal{Q}$ be a queue containing all edges which have i as an endpoint.
Step 1 : Let j be the first vertex not in $\tilde{\mathfrak{V}}$ seen as an endpoint in the queue $\mathcal{Q}$. Let $\{a, j\}$ be the corresponding edge in $\mathcal{Q}$. Update $\mathfrak{p} = \mathfrak{p} \cup \{a, j\}$, $\tilde{\mathfrak{V}} = \tilde{\mathfrak{V}} \cup \{j\}$ and add all edges which include j and another vertex not in $\tilde{\mathfrak{V}}$ as endpoints to the *front* of $\mathcal{Q}$.
Step 2 : **if** $\tilde{\mathfrak{V}} = \mathfrak{V}$ **then**
| Stop.
else
| Go to Step 1.
end
Result : *Ordered Edge Set* — $\mathfrak{p}$

Applying algorithm 19 on the example given in figure 5.1 would give the ordered set

$$\mathfrak{p} = \{\{2,9\},\{3,9\},\{3,7\},\{6,7\},\{6,8\},\{5,8\},\{4,6\},\{7,10\},\{1,10\}\} \tag{5.2}$$

This depth first approach allows the full exploration of a branch of the tree before the exploration of another, optimising the use of the current objects in the cache. Another possible method is a breadth first search [83], which aims to explore branches simultaneously. However, if the number of branches is large then for small caches this risks important objects being evicted whilst cycling through the edges in each separate branch.

In summary, memoisation is a powerful tool for fast implementation of UNCOVER, when suitable policies are chosen. In order to showcase the effect that memoisation with this policy has on the computation time of UNCOVER, we first introduce a concept which further enhances the potential benefits of storing previous results — reverse iterated batch importance sampling.

5.2 RIBIS: Reverse Iterated Batch Importance Sampling

Memoisation of the Iterated Batch Importance Sampling (IBIS) function has allowed the computational challenge of large cluster Bayesian evidence estimation to be mitigated significantly in situations where a previous index set is a subset of the new target index set — by starting from the bridging distribution provided by the former and adding observations. However, a natural question then arises as to whether it is conversely possible to utilise information from Sequential Monte Carlo (SMC) samplers whose associated index set is a *superset* of the new target index set? This has a clear use within UNCOVER, as the example shown in figure 5.1 highlights. Indeed, as previously stated the removal of the edge highlighted in blue requires the computation of the posteriors containing observations belonging to the sets $\mathfrak{V}_1 = \{2\}$ and $\mathfrak{V}_2 = \{1, 3, \ldots, 10\}$. If the edge highlighted in green is then subsequently removed, we obtain index sets $\mathfrak{V}'_1 = \{2, 9\}$ and $\mathfrak{V}'_2 = \{1, 3, \ldots, 8, 10\}$. We have seen the benefit of memoisation for obtaining the posterior containing $\mathfrak{V}'_1$ from the posterior containing $\mathfrak{V}_1$; however, it must be noted that $\mathfrak{V}'_2$ and $\mathfrak{V}_2$ also differ by a single observation, yet through the current use of memoisation the information we have on the posterior containing $\mathfrak{V}_2$ is not utilised.

Starting from an initial posterior distribution $\pi^0 = \pi(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}, \mathfrak{A}')$, where $\mathfrak{A}' \subseteq \{1, \ldots, n\}$ is an index set containing the observations used in the posterior, we seek to remove (rather than add) observations until we reach the target distribution $\pi^\Upsilon = \pi(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}, \mathfrak{A})$, where $\mathfrak{A} \subset \mathfrak{A}'$. Assuming we adopt Chopin's IBIS method (with a batch size of one) as previously done for adding observations, the reverse bridging distributions are obtained by removing an observation at each step, giving

$$\pi_t = \pi\left(\boldsymbol{\beta} \,\middle|\, \mathbf{X}, \mathbf{y}, \mathfrak{A}' \backslash \bigcup_{s=1}^{t} \mathfrak{B}_s\right) = \left[\prod_{i=1}^{n} \left[\pi(y_i \mid \boldsymbol{\beta}, \mathbf{x}_i)\right]^{\mathbb{I}\left(i \in \mathfrak{A}' \backslash \bigcup_{s=1}^{t} \mathfrak{B}_s\right)}\right] \pi(\boldsymbol{\beta}) \tag{5.3}$$

where $\mathfrak{B}_s \in \mathfrak{A}' \backslash \mathfrak{A}$ such that $(\mathfrak{B}_1, \ldots, \mathfrak{B}_\Upsilon)$ is a permutation of the ordered set of $\mathfrak{A}' \backslash \mathfrak{A}$. This results in the following updates to the importance weights given in

equation (3.25)[3]:

$$\tilde{w}_r^{\{t+1\}} = \frac{\gamma_{t+1}(\boldsymbol{\beta}_r^{\{t\}})}{\gamma_t(\boldsymbol{\beta}_r^{\{t\}})} = \left[\prod_{i=1}^{n} \left[\pi(y_i \mid \boldsymbol{\beta}_r^{\{t\}}, \mathbf{x}_i)\right]^{\mathbb{1}(i \in \mathfrak{B}_{t+1})}\right]^{-1} \tag{5.4}$$

As seen previously we can measure the degeneracy of the weights through the Effective Sample Size (ESS). However, whilst use of the ESS as an indicator of weight degeneracy is still valid, it is important to take into consideration the subtle differences in how the weights become degenerate in Reverse Iterated Batch Importance Sampling (RIBIS).

When moving from prior to posterior in the typical way (through addition of observations), we expect the posterior to contract. Indeed, under certain conditions which can be met through specification of a suitable prior, the Bernstein von Mises theorem [41] shows that

$$\sqrt{n}(\beta - \boldsymbol{\beta}_0) \xrightarrow{\mathcal{D}} \mathcal{N}(0, I(\boldsymbol{\beta}_0)^{-1}) \tag{5.5}$$

where $\boldsymbol{\beta}_0$ is the true value of β and $I(\cdot)$ is the Fisher information matrix. This implies that the variance of the posterior contracts at a rate of n^{-1}.

Therefore, in the traditional application of IBIS, as we transition from one bridging distribution to the next the distributions contract, and the weights of the samples in the tail of the partial posteriors degenerate rapidly relative to the weights of samples towards the center of the distribution. Subsequently, in the resampling step, 'tail' samples are much more likely to be removed. This does not necessarily hold when considering RIBIS, however. Returning to equation (5.4), note that the weight here is determined by the observation $\mathfrak{B}_{t+1}$, such that samples of the posterior π_t which explain this observation well (i.e. result in a high likelihood) are now given a small weighting as $\mathfrak{B}_{t+1}$ is to be removed. However, for large $|\mathfrak{A}' \setminus \bigcup_{s=1}^{t} \mathfrak{B}_s|$ where the likelihood has a dominating effect, there is a high chance that the center of this posterior will explain observation $\mathfrak{B}_{t+1}$ well (noting that $\mathfrak{B}_{t+1} \in \mathfrak{A}' \setminus \bigcup_{s=1}^{t} \mathfrak{B}_s$). So, in general, samples in the center will have low weighting (due to weights being

[3]Note here that subscript r refers to sample index and not cluster index.

determined by the inverse likelihood), but in the tails of π_t this is less likely. Again assuming $|\mathfrak{A}' \setminus \bigcup_{s=1}^{t} \mathfrak{B}_s|$ is large such that the Bernstein von Misses theorem holds, π_t will be centered in a similar position to the center of π_{t+1}. Therefore, for samples distributed around the centre of π_t, the weights will approximately be 1 (with 1 being the lowest achievable un-normalised weight here). For samples in the tail of π_t, however, the density of these samples will be significantly higher in π_{t+1} than π_t, because in reverse we undergo posterior *expansion* rather than posterior contraction. This then results in 'tail' samples being favoured when resampling.

This preference for samples in the tail of π_t is not explicitly detrimental to the process. Indeed, before resampling the weighted mean $\boldsymbol{\mu}$ and weighted covariance Σ are estimated to form a multivariate normal proposal. Due to the larger weights in every tail of the distribution, the theoretical form of the proposal distribution will have a similar center to π_{t+1} but with a highly inflated variance. The samples from this proposal used in the one-step independent Metropolis–Hastings sampler will then cover the center of the target distribution as well as the tails of the target distribution. Therefore, even with a high probability of only retaining tail samples after resampling, the 'move' step of RIBIS should provide proposals which give a wide coverage of the target distribution.

There is an issue present with RIBIS, however, as obtaining consistent tail information even with an inflated normal proposal is still challenging. Recall the acceptance stage of the one-step independent Metropolis–Hasting sampler is given as

$$\alpha(\boldsymbol{\beta}^{\{t\}}, \boldsymbol{\beta}^{\{t+1\}}) = \min\left\{1, \frac{\pi_{t+1}(\boldsymbol{\beta}^{\{t+1\}} \mid \mathbf{X}, \mathbf{y}, \mathfrak{A}' \setminus \bigcup_{s=1}^{t+1} \mathfrak{B}_s) q(\boldsymbol{\beta}^{\{t\}} \mid \boldsymbol{\mu}, \Sigma)}{\pi_{t+1}(\boldsymbol{\beta}^{\{t\}} \mid \mathbf{X}, \mathbf{y}, \mathfrak{A}' \setminus \bigcup_{s=1}^{t+1} \mathfrak{B}_s) q(\boldsymbol{\beta}^{\{t+1\}} \mid \boldsymbol{\mu}, \Sigma)}\right\} \tag{5.6}$$

where $q(\cdot)$ is the probability density function of the multivariate normal proposal. Samples which move towards the center of the target distribution have a high probability of acceptance. This is due to the large target distribution ratio $\pi_{t+1}(\boldsymbol{\beta}^{\{t+1\}} \mid \mathbf{X}, \mathbf{y}, \mathfrak{A}' \setminus \bigcup_{s=1}^{t+1} \mathfrak{B}_s) / \pi_{t+1}(\boldsymbol{\beta}^{\{t\}} \mid \mathbf{X}, \mathbf{y}, \mathfrak{A}' \setminus \bigcup_{s=1}^{t+1} \mathfrak{B}_s)$ and the proposal ratio $q(\boldsymbol{\beta}^{\{t\}} \mid \boldsymbol{\mu}, \Sigma) / q(\boldsymbol{\beta}^{\{t+1\}} \mid \boldsymbol{\mu}, \Sigma) \approx 1$, as the proposal will theoretically be hyper expanded compared to the target distribution so moves towards the center have a

bigger effect on the target than on the proposal. Consequently, moves towards the tails of π_{t+1} from the tails of π_t will have a low probability of acceptance due to the expansive proposal again resulting in the target ratio having a dominating effect (with clearly the preference being for the target distribution not to move to an area of low density[4]).

Additionally, we must consider the detrimental effect rejection has on the process. For IBIS, rejection of a proposed sample leads to a sample from a previous distribution remaining in the set of samples, and this sample will either contain information about a high density region of the new distribution or provide tail information about the new distribution. For RIBIS, it will be highly unlikely (as previous distributions are contracted posteriors) that the remaining sample will provide any tail information for the new distribution when the proposal is rejected. This effect will be compounded through subsequent resample-move steps, culminating in a biased estimation of the Bayesian evidence.

It is important to note here that this indication of bias is restricted to the setting where we take only a finite number of steps using the independent Metropolis–Hasting sampler. Indeed, for the weighted Gaussian proposal distribution used in IBIS, the Metropolis–Hastings algorithm asymptotically is guaranteed to provide samples from the target posterior, even in situations where this posterior is an expansion of the proposal. This guarantee is only asymptotic, however; for an efficient algorithm we aim to take as few Metropolis–Hastings steps as possible. As a result, this biasing effect will be present in RIBIS, and a less computationally expensive solution is required than simply insisting on a large number of Metropolis–Hastings steps when moving the samples.

Figure 5.3 showcases the bias issue through the generation of 3000 log Bayesian evidences. The red points are obtained through repetition of the following procedure 1000 times using the iris dataset: first 30 observations are added to the prior (a standard normal) using IBIS to obtain an estimate for the log Bayesian evidence at this particular partial posterior (plotted on the x-axis), then all remaining obser-

[4]Though of course moves into the tails of π_{t+1} are not impossible.

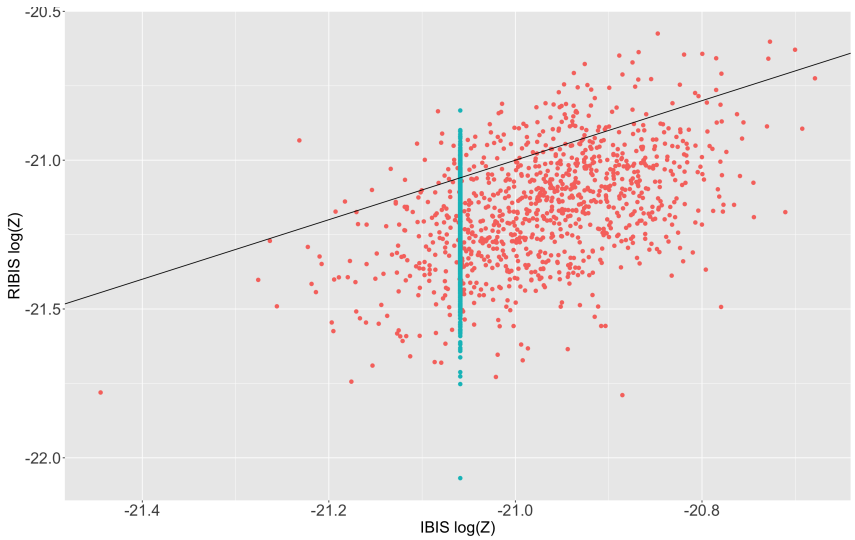


Figure 5.3: Multiple runs of IBIS and RIBIS (without transform) to obtain $\log(Z)$ for a partial posterior of the iris dataset containing 30 observations. The species response was altered to either 'versicolor' or 'not versicolor' to obtain a binary output. Red points represent runs with different initialisations whereas blue points represent runs with the same initialisation. The black line represents the scenario where the IBIS output is identical to the RIBIS output.

vations are added using IBIS, before being removed using RIBIS to obtain a second estimate of the 30 observation partial posterior log Bayesian evidence (plotted on the y-axis). The blue points are obtained by taking the samples from a single fixed full IBIS posterior and reversing to the 30 observation partial posterior (repeated 1000 times), enabling visualisation of the variation solely in the RIBIS procedure. To highlight the bias, the line $y = x$ has been added to the plot. As the two estimates are of the same quantity, the points should fall along this line, but there is a clear downward bias in the RIBIS estimate.

In order to combat this bias, we will assume that the Bernstein von Mises theorem holds and that the target distribution contains a large number of observations. This assumption is crucial, as the closer to the prior we get, the further away our proposal distribution becomes from the prior (if the prior is not a multivariate normal), and the suitability of the following proposed transformation weakens.

If the number of observations in the target distribution is large then all bridging distributions including the initial distribution will have a large number of observations as well (since we work in reverse), and so we can assume that all distribu-

tions in the RIBIS sequence are approximately $\mathcal{N}\left(\boldsymbol{\beta}_0, \frac{I(\boldsymbol{\beta}_0)^{-1}}{n}\right)$. Therefore, letting $|\mathfrak{A}' \setminus \bigcup_{s=1}^{t} \mathfrak{B}_s| = n^{\{t\}}$:

$$\begin{aligned} \boldsymbol{\beta}^{\{t\}} \sim \pi_t &\approx \mathcal{N}\left(\boldsymbol{\beta}_0, \frac{I(\boldsymbol{\beta}_0)^{-1}}{n^{\{t\}}}\right) \\ \implies \sqrt{\frac{n^{\{t\}}}{n^{\{t\}}-1}}(\boldsymbol{\beta}^{\{t\}} - \boldsymbol{\beta}_0) + \boldsymbol{\beta}_0 &\dot{\sim} \mathcal{N}\left(\boldsymbol{\beta}_0, \frac{I(\boldsymbol{\beta}_0)^{-1}}{n^{\{t\}}-1}\right) \approx \pi_{t+1} \end{aligned} \tag{5.7}$$

as $|\mathfrak{B}_s| = 1$ for all $s = 1, \ldots, \Upsilon$. Consequently, we propose applying the transformation in equation (5.7) to the current sample set in order to obtain an approximate sample from the next bridging distribution. This provides approximate tail samples for the next bridging distribution and as a result can counter the one-step biasing effect witnessed previously. Finally, we approximate $\boldsymbol{\beta}_0$ using the weighted mean which forms the mean for the proposal distribution. A simple illustration of this can be seen in figure 5.4. Here we can see that through the transformation of the

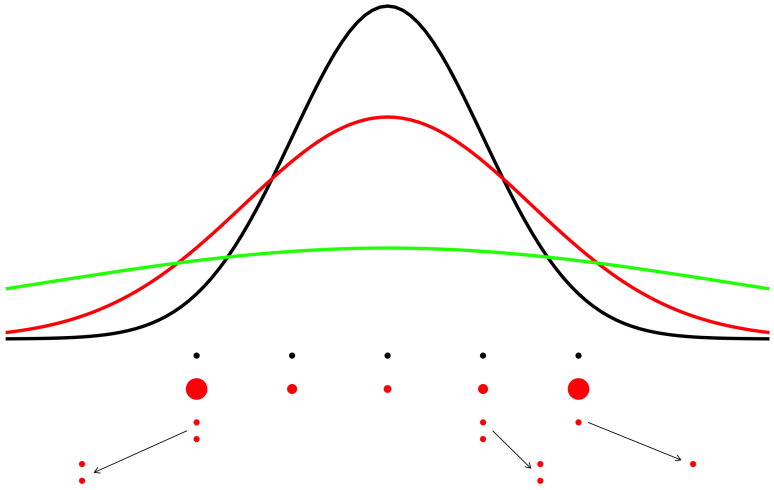


Figure 5.4: Densities of the current distribution (black), target distribution (red) and proposal distribution (green). Four rows of points are also given representing the position of distribution samples. The first row gives samples from current distribution, the second row gives weighted samples (with weight corresponding to size) of the target distribution, the third row is obtained by resampling the second row of points according to weight and the fourth row is obtained by applying the transformation in equation (5.7) to the third row of points.

samples before the move step, we have obtained tail information about the target distribution, which is clearly not available before the transformation and difficult to

obtain through one-step Metropolis–Hastings moves.

The full procedure is detailed in algorithms 20 and 21. The necessity for algorithm 20 comes from the fact that in order to apply the transformation in equation (5.7) we need un-weighted samples from the current distribution.

Algorithm 20: Resample-Move step for algorithm 21

Input : *Number of Samples* — N, *Distribution Samples* — $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T$, *Distribution Weights* — $\mathbf{w} = (w_1, \ldots, w_N)^T$, *Distribution Index Set* — $\mathfrak{A}'$, *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$

Step 1 : Let $(\tilde{\boldsymbol{\beta}}_1, \ldots, \tilde{\boldsymbol{\beta}}_N)^T = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T$. Let $q \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with

$$\boldsymbol{\mu} = \mathbf{w}^T(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T, \quad \Sigma = \left((\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T - \boldsymbol{\mu}\right)^T \operatorname{diag}\{\mathbf{w}\} \left((\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T - \boldsymbol{\mu}\right)$$

Step 2 : **for** $r = 1, \ldots, N$ **do**

  Sample $\boldsymbol{\beta}_r$ from $\{\tilde{\boldsymbol{\beta}}_1, \ldots, \tilde{\boldsymbol{\beta}}_N\}$, where $\mathbb{P}(\boldsymbol{\beta}_r = \tilde{\boldsymbol{\beta}}_s) = w_s$

  Update $\boldsymbol{\beta}_r$ using algorithm 3, replacing π_{t+1} with $\pi(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \mathfrak{A}')$.

end

Result : $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T$

We can witness the one-step bias correction by revisiting the iris example shown in figure 5.3, and applying the bias-correcting RIBIS algorithm. The results are shown in figure 5.5, where the biasing effect is clearly rectified. A real-world example can be seen for the mall customers dataset [84, 85], where we take as our binary response $\mathbb{1}\{\text{spending score} \leq 50\}$, with sex, age and income as the covariates (summary information is showcased in table B.2, given in appendix B.2.1). Again, the log Bayesian evidence is obtained from the partial posterior of a sample of 30 observations through either IBIS from a standard normal prior or RIBIS from the full posterior of all 200 observations. The results of this experiment can be seen in figure 5.6.

5.2.1 Implementation Within UNCOVER

Just as the benefits of memoisation have been highlighted for the IBIS algorithm, the same is true of RIBIS. One may not even need separate memoised functions, as the two processes can be combined through a wrapper algorithm (see algorithm 22).

Memoisation of this wrapper function has no effect on the evaluation policy but does have an effect on the eviction policy. To see this, we note that now when

Algorithm 21: RIBIS for target $\pi(\boldsymbol{\beta}|\mathbf{X},\mathbf{y},\mathfrak{A})$ from $\pi(\boldsymbol{\beta}|\mathbf{X},\mathbf{y},\mathfrak{A}')$

Input : *Standard IBIS Parameters* — $(\mathbf{X},\ \mathbf{y},\ \xi,\ N)$, *Target Index Set* — $\mathfrak{A}$, *Current Index Set* — $\mathfrak{A}'$, *Current Posterior Samples* — $(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_N)^T$, *Current Posterior Weights* — $\mathbf{w}=(w_1,\ldots,w_N)^T$, *Current Model's Bayesian Evidence* — Z

Initialisation : Obtain $(\boldsymbol{\beta}_1^{\{0\}},\ldots,\boldsymbol{\beta}_N^{\{0\}})^T$ from algorithm 20. Let $t=0,\tilde{t}=0,\Upsilon=|\mathfrak{A}'\backslash\mathfrak{A}|$ and $(\mathfrak{B}_1,\ldots,\mathfrak{B}_\Upsilon)$ be a permutation of $\mathfrak{A}'\backslash\mathfrak{A}$.

Step 1 : **for** $r=1,\ldots,N$ **do**

Let $\tilde{w}_r^{t+1}=\left[\prod_{i=1}^n\left[\pi(y_i\mid\boldsymbol{\beta}_r^{\{t\}},\mathbf{x}_i)\right]^{\mathbb{1}\left(i\in\bigcup_{s=\tilde{t}+1}^{t+1}\mathfrak{B}_s\right)}\right]^{-1}$

end

Let $\mathbf{w}^{\{t+1\}}=\frac{\tilde{\mathbf{w}}^{\{t+1\}}}{\sum_{r=1}^N\tilde{w}_r^{\{t+1\}}}$

Step 2 : Pool together identical samples: Let $\boldsymbol{\delta}^\dagger$ be the index vector of the unique elements of $(\boldsymbol{\beta}_1^{\{t\}},\ldots,\boldsymbol{\beta}_N^{\{t\}})^T$. Let $\boldsymbol{\delta}^{\dagger\dagger}\in\mathbb{N}^{|\boldsymbol{\delta}^\dagger|}$ be such that $\delta_i^{\dagger\dagger}=|\{r\in\{1,\ldots,N\}:\boldsymbol{\beta}_r=\boldsymbol{\beta}_{\delta_i^\dagger}\}|$.

Step 3 : **if**

$$\left(\sum_{r\in\boldsymbol{\delta}^\dagger}w_r^{\{t+1\}}\delta_r^{\dagger\dagger}\right)^2\Bigg/\sum_{r\in\boldsymbol{\delta}^\dagger}\left(w_r^{\{t+1\}}\delta_r^{\dagger\dagger}\right)^2<\xi$$

then

Let $(\tilde{\boldsymbol{\beta}}_1^{\{t\}},\ldots,\tilde{\boldsymbol{\beta}}_N^{\{t\}})^T=(\boldsymbol{\beta}_1^{\{t\}},\ldots,\boldsymbol{\beta}_N^{\{t\}})^T$. Let $q\sim\mathcal{N}(\boldsymbol{\mu},\Sigma)$ with $\boldsymbol{\mu}=(\mathbf{w}^{\{t+1\}})^T(\boldsymbol{\beta}_1^{\{t\}},\ldots,\boldsymbol{\beta}_N^{\{t\}})^T$,
$\Sigma=\left((\boldsymbol{\beta}_1^{\{t\}},\ldots,\boldsymbol{\beta}_N^{\{t\}})^T-\boldsymbol{\mu}\right)^T\operatorname{diag}\{\mathbf{w}^{\{t+1\}}\}\left((\boldsymbol{\beta}_1^{\{t\}},\ldots,\boldsymbol{\beta}_N^{\{t\}})^T-\boldsymbol{\mu}\right)$
For $r=1,\ldots,N$ sample $\boldsymbol{\beta}_r^{\{t\}}$ from $\{\tilde{\boldsymbol{\beta}}_1^{\{t\}},\ldots,\tilde{\boldsymbol{\beta}}_N^{\{t\}}\}$ (where $\mathbb{P}(\boldsymbol{\beta}_r^{\{t\}}=\tilde{\boldsymbol{\beta}}_s^{\{t\}})=w_s^{\{t+1\}}$), let $\boldsymbol{\beta}_r^{\{t\}}=\sqrt{\frac{|\mathfrak{A}'|-\tilde{t}}{|\mathfrak{A}'|-t-1}}(\boldsymbol{\beta}_r^{\{t\}}-\boldsymbol{\mu})+\boldsymbol{\mu}$ and sample $\boldsymbol{\beta}_r^{\{t+1\}}$ using algorithm 3.
Update $Z=Z\times\frac{1}{N}\sum_{r=1}^N\tilde{w}_r^{\{t+1\}}$. Let $\tilde{t}=t+1$.

else

$(\boldsymbol{\beta}_1^{\{t+1\}},\ldots,\boldsymbol{\beta}_N^{\{t+1\}})^T=(\boldsymbol{\beta}_1^{\{t\}},\ldots,\boldsymbol{\beta}_N^{\{t\}})^T$

end

Step 4 : **if** $t+1=\Upsilon$ **then**

if $\tilde{t}=\Upsilon+1$ **then**

Stop.

else

Update $Z=Z\times\frac{1}{N}\sum_{r=1}^N\tilde{w}_r^{\{t+1\}}$. Stop.

end

else

Update $t=t+1$. Go to step 1.

end

Result : *Target Distribution Output* — $[(\boldsymbol{\beta}_1^{\{t+1\}},\ldots,\boldsymbol{\beta}_N^{\{t+1\}})^T,\mathbf{w}^{\{t+1\}},Z]$

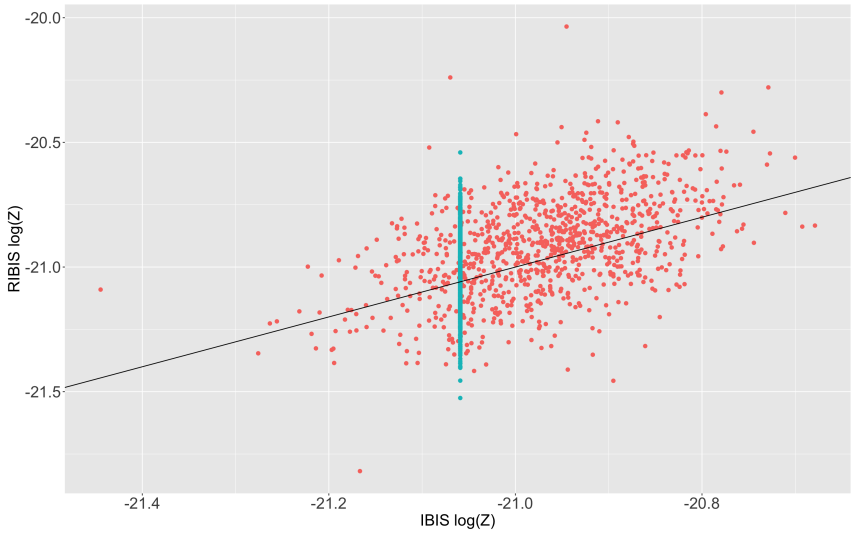


Figure 5.5: Multiple runs of IBIS and RIBIS (with transform) to obtain $\log(Z)$ for a partial posterior of the iris dataset containing 30 observations. Dataset and IBIS estimates are the same of those in figure 5.3. Red points represent runs with different initialisations whereas blue points represent runs with the same initialisation. The black line represents the scenario where the IBIS output is identical to the RIBIS output.

checking the cache, we search for either the largest subset of our target observation index set $\mathfrak{A}$ *or* the smallest set which contains $\mathfrak{A}$ as a subset. The additional choice one has for index sets in the cache pairs well with the depth first search used to order the edge removals. Recall the initial intention behind the search — to create a sequence of posteriors that differ from their previous model by only the addition of a single observation. However, note that the removal of an edge creates another vertex set, which previously had no benefits for the eviction policy, as they represent a sequence of posteriors which differ from their previous model by only the *removal* of a single observation. These posteriors can now be used through RIBIS.

We test the computational benefits of memoisation using a sample of 100 observations for each of the three Gaussians from the simulated dataset given in figure 4.4. As a result, for all runs a maximum number of clusters deforestation criterion was used, with $\kappa = 3$. All other flexible parameters for UNCOVER (for example $\varkappa$, ξ, N) are kept consistent throughout runs, with the adjustable parameter being the cache checking threshold. These results are given in figure 5.7, where a threshold value of 0 implies we always check the cache and a threshold value of 301 implies we never check the cache.

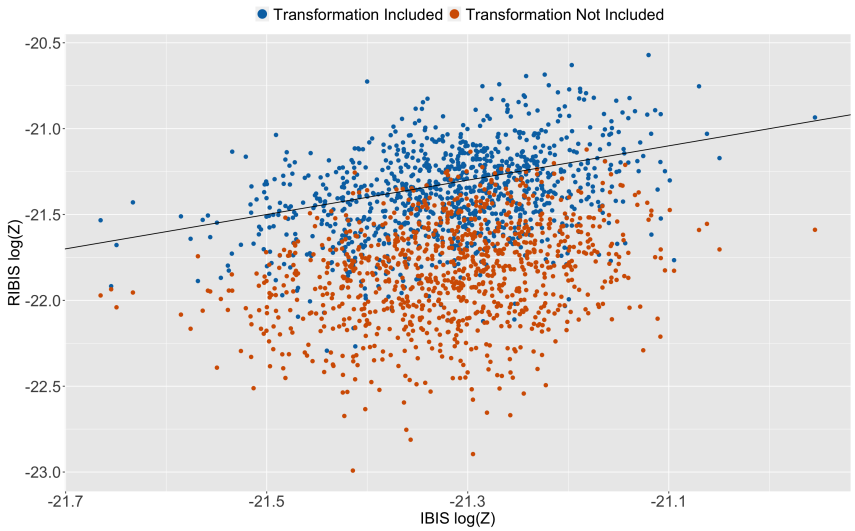


Figure 5.6: Multiple runs of IBIS and RIBIS to obtain $\log(Z)$ for a partial posterior of the mall customer dataset containing 30 observations. Blue points represent runs when the bias correction is in place and orange points represent runs when the bias correction is not in place (regarding the RIBIS algorithm).

Figure 5.7 showcases the subtly in specifying this threshold for computational speed up, as never checking the cache is costly due to never utilising the similar previous posteriors we have sampled from, but always checking the cache clearly is detrimental also as significant time is spent searching the cache for useful objects.

When considering the Bayesian evidences, what is notable is the consistency of output one can achieve through use of SMC estimation as we observe no obvious systematic drift as the threshold varies. There also appears to be (excluding one outlier) lower variability of output for the scenario where we always check the cache. This is to be expected as always checking the cache ensures that previous posteriors are utilised much more often, and as a result posterior samples are recycled to a much greater extent, reducing variability. The outlier is important, however, as it showcases that this constant recycling of samples can be detrimental if the initial samples (to be constantly re-used) are a poor representation of the posterior.

Finally, note that we introduced restrictions here which may have a detrimental effect for low values of the memoisation threshold, but are crucial for an acceptable output. Specifically, we highly recommend that any posterior that contains a

Algorithm 22: IBIS and RIBIS Wrapper

Input : *Covariate Matrix* — $\mathbf{X}$, *Response Vector* — $\mathbf{y}$,
Target Index Set — $\mathfrak{A}$, *Current Index Set* — $\mathfrak{A}'$, *ESS Threshold* — ξ,
Current Model's Bayesian Evidence — Z', *Number of Samples* — N,
Current Posterior Samples — $(\boldsymbol{\beta}'_1, \ldots, \boldsymbol{\beta}'_N)^T$,
Current Posterior Weights — $\mathbf{w}'$

Step 1 : **if** $\mathfrak{A} \subset \mathfrak{A}'$ **then**

Obtain $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T$, $\mathbf{w}$ and Z from algorithm 21.

else

Let $\mathfrak{B}_s = \sigma(s)$ for $s \in \mathfrak{A} \backslash \mathfrak{A}'$. Obtain $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T$, $\mathbf{w}$ and Z from a modified version of algorithm 4, where instead of initialising with the prior we initialise with the posterior $\pi(\boldsymbol{\beta} \mid \mathbf{X}, \mathbf{y}, \mathfrak{A}')$.

end

Result : *Target Distribution Posterior Samples* — $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)^T$,
Target Distribution Posterior Weights — $\mathbf{w}$,
Target Model's Bayesian Evidence — Z

number of observations below a certain threshold[5] should not be obtained through RIBIS. This is to ensure that the asymptotic properties of the posterior (which the transformation relies upon) are upheld, however, this will have a knock-on effect of reducing the usefulness of the cache for posteriors with small numbers of observations. Therefore, the threshold at which one excludes the use of RIBIS should be a key consideration when selecting a memoisation threshold. Additionally, we also note that if the optimal posterior selected from the cache requires using RIBIS, and the number of observations to remove is more than the number of observations in the target posterior, then we insist on applying IBIS from prior samples instead for obvious computational reasons.

It is important to mention that this is just one example of memoisation for a specific problem and a specific cache size. The conclusion one should draw from this experiment is that the extreme thresholds (i.e. always or never checking the cache) do not necessarily result in the fastest algorithms. Regarding quality of output, it is advised to avoid checking the cache in at least some situations to avoid the risk of poor initial samples having a knock-on effect for the rest of the algorithm[6].

[5]This can be specified by the user, with an ad-hoc recommendation being 30, although the appropriate threshold is problem specific (see appendix B.1 for more details).

[6]See appendix B.1 for further details.

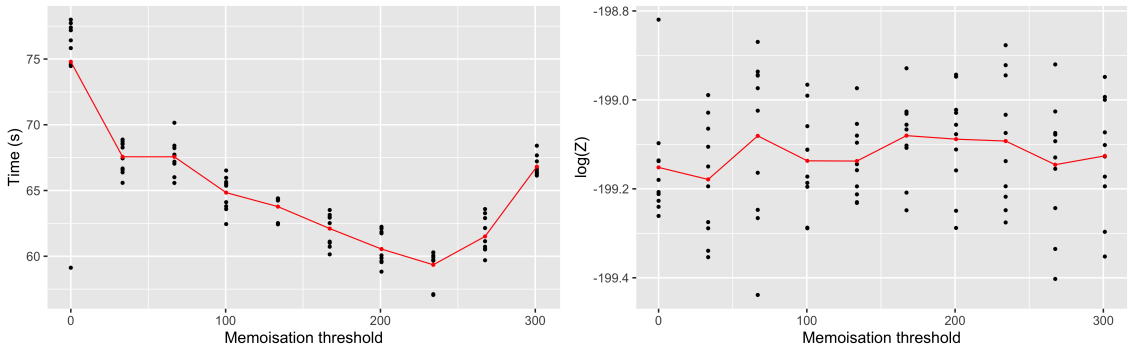


Figure 5.7: An UNCOVER algorithm's performance for different cache checking thresholds (i.e. if the number of observations in a posterior exceeds this threshold we check the cache for similar index sets). The left plot shows the algorithms performance with respect to computational time whereas the right plot shows the logarithm of the algorithm's Bayesian evidence output. Black dots represent individual runs while red dots represent the mean for that threshold.

5.3 Save States

Section 5.1 gave an insight into a non-conventional use of memoisation, due to the similarity sub-models are likely to have with previous computations. However, the main usage of function memoisation outside of UNCOVER, to remember results of a function evaluation for later use, has yet to be thoroughly discussed.

Referring specifically to sub-models, we have seen previously that actions based on a specific edge only affect the clusters which contain the endpoints of said edge in the current graph. Therefore, if in the current iteration a cluster is not selected to be spilt or merged with another cluster, then repeating all the edge actions that were performed on said cluster in the next iteration will give the same sub-posteriors discovered in the current iteration. A simple example of this can be seen in figure 5.8, where here we have two clusters, red and blue. Suppose that in one iteration we assess all edge removals in the blue and red clusters, consequently finding the optimal edges to remove for both the blue cluster and the red cluster. Then suppose we opt to remove an edge in the red cluster, highlighted in green. For the next iteration, the optimal edge to remove in the blue cluster will be the same, as the sub-posteriors created through removing the edges in the blue cluster will be the same as the previous iteration.

This hints at not necessarily using memoisation, as this risks optimal edge re-

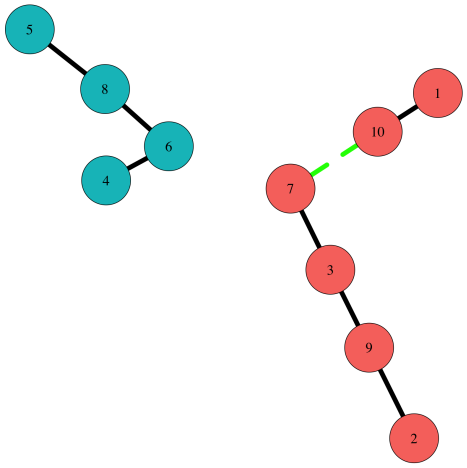


Figure 5.8: Minimum Spanning Forest of ten samples of $\mathcal{N}((0,0)^T, \mathcal{I}_2)$. Vertex labels correspond to the index of the observation, with colour corresponding to cluster. The majority of edges are given as black lines, with the exception being the green dashed line, highlighted for discussion in section 5.3.

moval sub posteriors getting evicted from the cache, but instead a separate storage system $\mathcal{S}$ for retained optimal edge actions.

The first instance of this is when removing edges. As this is done per cluster assuming a subgraph of just the connected component in question (see algorithm 7), we simply save the two Bayesian evidences of the clusters produced from the optimal edge removal in this component. So for cluster k and optimal edge $\{i,j\} \in \mathfrak{T}_k$ we store $Z_{k1}^{\{i,j\}^-}$ and $Z_{k2}^{\{i,j\}^-}$ with their associated index sets as an object in $\mathcal{S}$. If an edge removal benefits the system, the cluster which contains the overall optimal edge to be removed no longer needs to be stored in $\mathcal{S}$ (as it is retained in the updated iterations graph), so we remove it from $\mathcal{S}$. Then for the next iteration's edge removals the only information required is the optimal edges for the two clusters that are not represented in $\mathcal{S}$.

The second instance of this is for edge reintroductions. Before any edge is removed, we save the Bayesian evidence of the current cluster to be split, along with that cluster's associated observation index set, as an object in $\mathcal{S}$. As a result, for each edge in $\mathfrak{R}$ we have an associated object in $\mathcal{S}$. So, when checking to see if this edge is beneficial to reintroduce, if the two clusters formed by splitting the stored cluster have not been further split, the object in $\mathcal{S}$ provides an instant evaluation

of that cluster's Bayesian evidence. Of course this may not always be the case and so, if the clusters have been further split, we would calculate the Bayesian evidence from scratch and then let this (alongside the new associated index set) replace the defunct object in $\mathcal{S}$. Additionally, if an edge is reintroduced, then the object associated with the reintroduced edge is removed from $\mathcal{S}$. The structure of $\mathcal{S}$ is given in 5.1. This then operates as a form of memoisation, albeit with (given the potential importance of the information stored in $\mathcal{S}$) a highly customised cache structure and eviction policy.

Edge Action	Reference	Object
Removal	1	Information on the optimal edge to remove in cluster 1 **or** Blank
	⋮	⋮
	K	Information on the optimal edge to remove in cluster K **or** Blank
Reintroduction	$\sigma(\mathfrak{R})_1$	Partial or full information on the cluster formed by reintroducing edge $\sigma(\mathfrak{R})_1$
	⋮	⋮
	$\sigma(\mathfrak{R})_{\lvert\mathfrak{R}\rvert}$	Partial or full information on the cluster formed by reintroducing edge $\sigma(\mathfrak{R})_{\lvert\mathfrak{R}\rvert}$

Table 5.1: The structure of $\mathcal{S}$.

5.4 Asymptotic Approximations

Previous sections in this chapter have focused entirely on ensuring that the use of Sequential Monte Carlo (SMC) samplers is as efficient as possible. However, even with the utilisation of previous results, it is likely that, for some stages in particularly large problems, SMC samplers will still be too computationally expensive. An example of this is for the first Bayesian evidence that is calculated for the one-cluster model. No previous posteriors have been obtained and so we must run Iterated Batch Importance Sampling (IBIS) from prior samples to the full posterior with all observations included, which could be an expensive process for large n. Therefore, in extreme situations such as this we must look to make asymptotic approximations that can be computed with little expense.

As discussed in section 3.2.1, for large enough n (where n is the number of

observations) a transformation of the Bayesian Information Criterion (BIC) gives a suitable approximation of the Bayesian evidence[7]. The exact suitability of this approximation depends on three main factors:

1. Number of observations — is n large enough that the error term in the approximation is negligible.

2. Choice of prior — does the prior either; violate any assumptions made for the derivation of the BIC or have significant influence on the posterior with the data provided.

3. Separability — Are the observations completely separable such that the maximum likelihood estimator is not finite.

Factor 2 is somewhat avoidable through the specification of a weakly informative prior. Factor 3, whilst not avoidable, is easily tested by whether or not an iteratively reweighted least squares algorithm converges; this can be done through the `glm` function in `R`. Factor 1 is less straightforward, having some dependence on factors 2 and 3, and as such an ad-hoc threshold is suggested as a flexible parameter such that this can be adjusted to suit each individual problem. Note that a high threshold is not inherently problematic in practice, as factor 1 dovetails precisely with the setting where IBIS is slow.

In order to assess the viability of BIC, we must first consider the maximum likelihood:

$$\hat{L} = \pi(\mathbf{y} \mid \mathbf{X}, \hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_K, \mathfrak{V}) = \prod_{k=1}^{K} \prod_{i=1}^{n} \pi(y_i \mid \mathbf{x}_i, \hat{\boldsymbol{\beta}}_k)^{\mathbb{1}(i \in \mathfrak{V}_k)} \tag{5.8}$$

Due to the hard clustering enforced independence of the components, we can express $\hat{L} = \prod_{k=1}^{K} \hat{L}_k$, where $\hat{L}_k$ is the maximum likelihood of the sub-model generated by observations in $\mathfrak{V}_k$. This represents a problem, however, as if there is separability in

[7]Laplace's approximation [86] is also a suitable approximation, and can provide a more accurate estimate of the Bayesian evidence. However, computationally it is more expensive to apply Laplace's approximation using the function `LaplaceApproximation` [87] than it is to calculate the BIC through the function `BIC` [88], and so we opt for a transformation of the BIC as our large n approximation.

one of the sub-models then at least one $\hat{\boldsymbol{\beta}}_k$ will have none finite values, resulting in the overall BIC value being a poor estimator of the Bayesian evidence. Unfortunately this scenario is not uncommon, as for every iteration of UNCOVER we shall consider edge removals which leave one observation in its own cluster — automatically giving a singular sub-likelihood.

The separation of the likelihood does not need to be viewed as limiting factor, however, as this can be used as an advantage due to the posterior being a product of sub-posteriors. Indeed, as has been seen before in equation (4.12), the Bayesian evidence is also separable (i.e. $Z = \prod_{k=1}^{K} Z_k$). Therefore, if $|\mathfrak{V}_k|$ is sufficiently large and the corresponding observations do not suffer from separability, then

$$Z_k \approx \text{BIC}_k = \hat{L}_k \times |\mathfrak{V}_k|^{-\frac{p}{2}} \tag{5.9}$$

where $p = |\boldsymbol{\beta}_k|$. As in UNCOVER the estimation of Z_k is done separately for each $k = 1, \ldots, K$, BIC_k can simply be dropped in as a replacement for IBIS (algorithm 4) if deemed appropriate. The algorithm for obtaining BIC_k is given in algorithm 23. This algorithm can be implemented in `R` [88] using the function `BIC`.

Algorithm 23: BIC Generation for $\mathfrak{V}_k$

Input : *Covariate Data for Cluster k* — $\mathbf{X}_{\mathfrak{V}_k,\cdot}$,
Response Vector for Cluster k — $\mathbf{y}_{\mathfrak{V}_k}$, *Convergence Threshold* — $\eta > 0$,
Starting Value Vector for Regression Coefficients — $\boldsymbol{\beta}_k$

Step 1 : Obtain the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_k$ through algorithm 5.

Step 2 : Let

$$\text{BIC}_k = \pi(\mathbf{y}_{\mathfrak{V}_k} \mid \mathbf{X}_{\mathfrak{V}_k,\cdot}, \hat{\boldsymbol{\beta}}_k) \times |\mathfrak{V}_k|^{-\frac{|\hat{\boldsymbol{\beta}}_k|}{2}}$$

Result : *BIC Value for Cluster k* — BIC_k

We can showcase the advantages and disadvantages of approximating the Bayesian evidence with a transformation of the BIC value with the following highly artificial example, designed to highlight the crucial behaviour of BIC estimation. Note that this example is not run through the whole UNCOVER algorithm. We simulate a dataset with covariates from a mixture of ten Gaussians with increasing numbers of

observations in each group, such that in total $\mathbf{X} \in \mathbb{R}^{25575\times 2} = (\mathbf{X}^1, \ldots, \mathbf{X}^{10})^T$

$$\mathbf{x}_i^j \sim \mathcal{N}\left((j,j)^T, \frac{1}{36}\mathcal{I}_2\right) \tag{5.10}$$

where $j = 1, \ldots, 10$, $\mathbf{X}^j \in \mathbb{R}^{(25\times 2^{j-1})\times 2}$ and $i = 1, \ldots, (25 \times 2^{j-1})$. We also insist that each of the 10 groups has a differing relationship with the response such that:

$$\boldsymbol{\beta}_j = \frac{6}{\cos(u_j) + \sin(u_j)}\left(j, \frac{\sin(u_j)}{\cos(u_j) - \sin(u_j)}, \frac{-\cos(u_j)}{\cos(u_j) - \sin(u_j)}\right) \tag{5.11}$$

$$u_j \sim \mathcal{U}(0, 2\pi) \tag{5.12}$$

$$y_i^j \sim \text{Bern}\left((1 + e^{-(1,(\mathbf{x}_i^j)^T)\boldsymbol{\beta}_j})^{-1}\right) \tag{5.13}$$

where $\mathbf{y} = (\mathbf{y}^1, \ldots, \mathbf{y}^{10})^T$, $j = 1, \ldots, 10$ and $i = 1, \ldots, (25 \times 2^{j-1})$. This dataset can be visualised in figure 5.9.

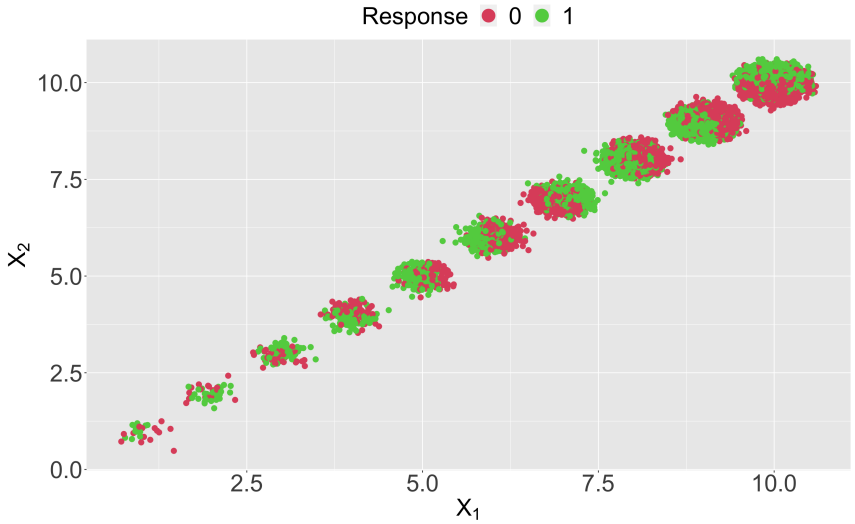


Figure 5.9: Ten cluster Gaussian dataset, with increasing cluster size as the center of the Gaussian increases from $(1,1)^T$ to $(10,10)^T$. Colours correspond to observation's associated responses.

Initially, we approximate the Bayesian evidence of each cluster using just BIC values and take the product of these values to be the total Bayesian evidence estimate. Then, we replace the Bayesian evidence estimate of the smallest (i.e. the computationally most tractable) cluster with an estimate using the standard IBIS algorithm (algorithm 4) with 1000 samples and an ESS threshold of 500. We repeat this process, successively estimating the next most computationally tractable cluster

by IBIS rather than BIC, until every cluster has their Bayesian evidence estimated by the IBIS algorithm. This process is repeated 10 times to assess the variability of our estimates. We also produce 10 estimates of the log Bayesian evidence using only the standard IBIS algorithm but with 10000 samples and an ESS threshold of 5000, in order to provide a reliable baseline for the true Bayesian evidence. Finally, we note that as the true coefficients are known we can use their mean and variance as parameters for the standard normal prior. Clearly this will not be available in real-world scenarios. However, as the true coefficients are varied (see table 5.2), the prior specification simply ensures that the prior mean is not centered around the true coefficient of any particular cluster.

Cluster	β_1	β_2	β_3
1	-1.081264	-3.6674217	4.7486859
2	4.937830	2.8246206	-5.2935355
3	-22.800575	1.9134925	5.6866991
4	28.538204	-1.2706377	-5.8639133
5	31.035329	-5.9962969	-0.2107690
6	34.704636	-5.9962485	0.2121424
7	-57.620843	5.1455318	3.0860173
8	64.148431	-5.3969724	-2.6215814
9	75.273299	-4.8974832	-3.4662168
10	-62.660040	0.2721807	5.9938233
Mean	**10.769097**	**-2.174598**	**0.640837**

Table 5.2: Coefficient Matrix for the true coefficients of the ten Gaussian example, along with the mean of these coefficients.

The results of this process in terms of log computational time and log Bayesian evidence are given in figure 5.10.

The left-hand plot of figure 5.10 showcases the exponential growth in time that occurs when replacing BIC estimation with estimation through the SMC method (i.e. IBIS), with the anomaly when all clusters are estimated by BIC occurring due to the fact that this estimation in some runs happened instantaneously, meaning that on a log scale this tends to towards minus infinity and therefore skews the result. On the other hand, the right-hand plot shows that using IBIS results in much more accurate estimation than using BIC as an estimator. This is particularly true for small clusters where the asymptotic assumptions made to justify the use of BIC do not hold. The pink dashed line shows this, as initially we make large gains in

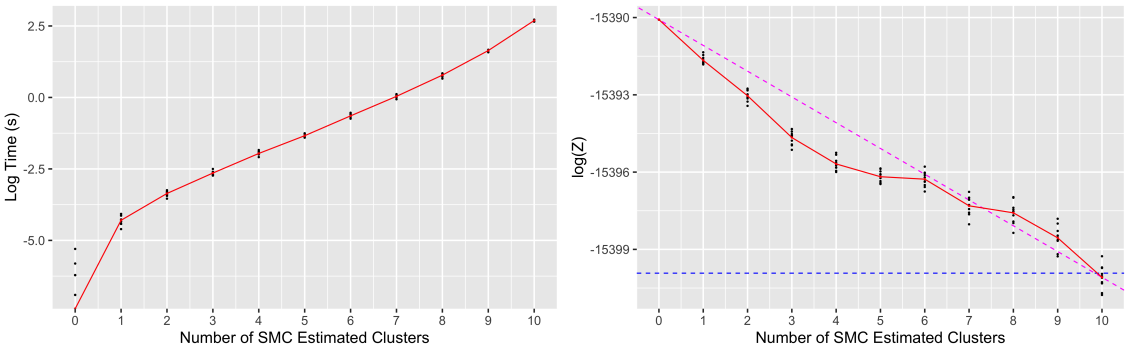


Figure 5.10: Log computation time and log Bayesian evidence for the ten Gaussian example shown in figure 5.9. A cluster not estimated through SMC is estimated by BIC. Black dots represent individual runs while red dots represent the mean. For the right-hand plot the blue dotted line represents the mean log Bayesian evidence for 10 SMC sampler runs on all ten Gaussians with 10000 samples and the pink dotted line represents the line passing through the mean $\log(Z)$ for 0 SMC clusters and 10 SMC clusters (with 1000 samples).

accuracy when using SMC samplers for the small clusters as BIC is quite a poor approximation. Incidentally for the large clusters (i.e. clusters on the right-hand side of figure 5.9) the gains made by replacing a BIC estimator with an SMC estimator are not as pronounced. This results in runs dropping below the pink line initially before eventually rising above, meaning that we gain more by replacing smaller clusters with a more accurate estimation method than we do replacing larger clusters with a more accurate estimation method. This justifies the asymptotic considerations one must make when attempting to use BIC as an estimator (see appendix B.1 for a further discussion on when it is appropriate to use a transformation of the BIC as an estimate for the Bayesian evidence) .

As a final note, having discussed the implementation of memoisation for SMC samplers (with the method having clear benefits for both IBIS and RIBIS), we must also do so for estimation using BIC values. Indeed, memoisation is possible for the BIC method as we can use the coefficients of similar models as the starting coefficients in algorithm 23. Whilst possible, the speed at which this algorithm is carried out in `R` through `BIC` (specifically through `glm`) makes it almost always detrimental to check the cache, and so only for extremely large datasets do we encourage the use of memoisation for BIC value calculation.

5.5 'UNCOVER' Package

The UNCOVER package [11] developed in R provides the UNCOVER algorithm as the primary function, with a separate options function for further specification of parameters. Additionally, the one batch Iterated Batch Importance Sampling (IBIS) and RIBIS wrapper for Bayesian logistic regression is also given as a function, again with a separate options function to allow for flexible parameter specification. These two main functions are implemented with all tools highlighted in this chapter available to the user and purposely held in a separate options function. The justification for this is that this allows the algorithms to be accessible to users at every level — it is possible for a novice to simply run the algorithm without knowledge of concepts such as memoisation (although this may result in a slow run time) whilst also allowing a user more comfortable with these ideas to increase the efficiency of computation.

5.5.1 Dependencies

The UNCOVER package requires several dependencies:

1. mvnfast [89] — This package is required for fast generation of samples from multivariate Gaussian distributions.

2. igraph [90] — This package is required for the graphical structures used within UNCOVER.

3. memoise [91] — This package is required for the memoisation of functions within UNCOVER.

4. stats [88] — This package is required for fast calculation of distances, sampling from uniform distributions and for fast calculation of weighted covariance matrices.

5. cachem [92] — This package is required for the generation of caches used within UNCOVER.

6. Other [93–98] — These packages are used for asthetics.

In order to utilise memoisation for UNCOVER we rely on function memoisation through `memoise` which itself relies on cache generation through `cachem`. However, it must be noted that the keys used for function caches are unique strings and not the arguments of the function. This creates a problem with checking the cache, as we no longer can retrieve the argument of an object in the cache, which is used to determine if the object is a suitable bridging distribution for IBIS or RIBIS. To bypass this problem, we simply insist that the IBIS function returns the observation index set as one of the outputs.

Additionally, we also need to consider function memoisation for situations where we do not manually check the cache and the problems that arise from having multiple arguments. For example, the IBIS and RIBIS wrapper function requires a target index set $\mathfrak{A}$ and a current index set $\mathfrak{A}'$ as arguments. However, after we have obtained the target posterior samples from the function we no longer have interest in which bridging distribution we started from (target posterior samples alongside the Bayesian evidence are valid regardless of whether the initial samples were from the prior or a bridging distribution based on $\mathfrak{A}'$). So when we store this object in the cache, the key which is generated should only be defined by the argument $\mathfrak{A}$, as that is the only argument of interest related to the output. Thankfully it is possible to only consider certain arguments for key generation when using the function `memoise` in the package `memoise`.

5.5.2 `UNCOVER` Function

The framework for the `R` function `UNCOVER` is given as:

```
UNCOVER(X, y, mst_var = NULL, options = UNCOVER.opts(), stop_criterion
= 5, deforest_criterion = "None", prior_mean = rep(0,ncol(X)+1),
prior_var = diag(ncol(X)+1), verbose = TRUE)
```

where the algorithm specific arguments are as follows:

`X` — Covariate matrix.

`y` — Response vector.

`mst_var` — Covariates used for construction of the minimum spanning tree ($\mathfrak{P}$). The default is to use all covariates.

`stop_criterion` — Stopping criterion $\varkappa$.

`deforest_criterion` — Deforestation criterion.

`prior_mean, prior_var` — The default prior for a cluster sub-model is a Gaussian. These specify the hyper-parameters for this prior.

Note how we allow the specification of no deforestation stage, with this being the default. Of course this is not generally advisable, however, this does allow for a novice user to run the function without specific knowledge of the deforestation criteria detailed in section 4.4. If a deforestation criterion is required, however, this can be specified through `deforest_criterion` and `options`. The `options` argument for `UNCOVER` can only specified through the function `UNCOVER.opts`, which actually allows for much more than deforestation criterion specification, and is given as:

```
UNCOVER.opts(N = 1000, train_frac = 1, max_K = Inf, min_size = 0, reg =
0, n_min_class = 0, SMC_thres = 30, BIC_memo_thres = Inf, SMC_memo_thres
= Inf, ess = N/2, n_move = 1, prior.override = FALSE, rprior = NULL,
dprior = NULL, diagnostics = TRUE, RIBIS_thres = 30, BIC_cache =
cachem::cache_mem(max_size = 1024 * 1024^2, evict = "lru"), SMC_cache =
cachem::cache_mem(max_size = 1024 * 1024^2, evict = "lru"), ...)
```

where the algorithm specific arguments are as follows:

`N` — Number of samples required for the Sequential Monte Carlo (SMC) sampler.

`train_frac` — What fraction of the data should be used as training data. Only required for validation data deforestation criterion.

`max_K` — Maximum number of clusters allowed in the final output. Only required for number of clusters deforestation criterion.

`min_size` — Minimum number of observations a cluster must have in the final input. Only required for size of clusters deforestation criterion.

`reg` — Maximum regret factor. Only required for maximal regret deforestation criterion.

`n_min_class` — Minimum number of observations whose response is in the minority class allowed per cluster. Only required for balanced response deforestation criterion.

`SMC_thres` — Threshold. Posteriors which contain a number of observations above this threshold attempt Bayesian evidence estimation through the Bayesian Information Criterion (BIC).

`BIC_memo_thres, SMC_memo_thres, BIC_cache, SMC_cache` — Thresholds and caches for the memoisation of the BIC transformation function and the SMC sampler. Posteriors which contain a number of observations above the thresholds check their respective caches for similar previous evaluations.

`ess` — Effective sample size threshold ξ.

`n_move` — `UNCOVER` allows for multiple Metropolis–Hasting steps to be carried out at a time to ensure the samples are more representative of the target distribution, which can lead to a better estimate of the Bayesian evidence. `n_move` specifies how many steps are taken.

`prior.override, rprior, dprior` — Arguments to specify a custom prior in terms of sampling function (`rprior`) and density function (`dprior`). Only holds if `prior.override` is `TRUE`.

`RIBIS_thres` — Threshold. If using an SMC sampler to calculate the Bayesian evidence of a posterior and the observations in this posterior form a subset of a previously generated posterior (which is in `SMC_cache`), then to consider using RIBIS the number of observations in the target posterior must be above this threshold.

`UNCOVER.opts` allows the user to specify deforestation criteria, several computational efficiency tools and allows the specification of differing prior beliefs than that of a Gaussian prior. One should be cautious with prior specification, however, as

certain choices can violate the assumptions made for the usage of BIC values and RIBIS. Certain choices of prior (for example a point mass) may even violate the similarity of neighbouring distributions which standard IBIS relies on to produce a suitable estimate of the Bayesian evidence.

The `diagnostics` argument for `UNCOVER.opts` being set to `TRUE` allows `UNCOVER` to collect diagnostic information as the algorithm progresses, which always includes the changes in the log Bayesian evidence when an action is performed and then can provide additional information on an aspect of interest depending on the deforestation criteria specified. For example, for the number of clusters criterion, diagnostics will track the number of clusters after each edge action.

5.5.3 `IBIS.logreg` Function

In addition to `UNCOVER`, if one wished to obtain posterior samples and the Bayesian evidence of a particular cluster, one could use `IBIS.logreg`:

```
IBIS.logreg(X, y, options = IBIS.logreg.opts(), prior_mean =
rep(0,ncol(X)+1), prior_var = diag(ncol(X)+1))
```

where the algorithm specific arguments follow that of the arguments for the function `UNCOVER`.

As with `UNCOVER`, we also provide an `options` argument to allow for further specification. This will always be an output from the function `IBIS.logreg.opts`:

```
IBIS.logreg.opts(N = 1000, ess = N/2, n_move = 1, weighted = FALSE,
prior.override = FALSE, rprior = NULL, dprior = NULL,...)
```

where again algorithm specific arguments follow that of the arguments for the function `UNCOVER`. The exception to this, however, is the argument `weighted`, which indicates whether the output posterior samples should be weighted or unweighted. If `weighted` is `FALSE`, we force the final samples to be progressed through `n_move` Metropolis–Hastings steps to get an unweighted sample.

5.5.4 Summary

The `UNCOVER` package allows for the direct application of the concepts introduced in chapter 4, specifically the UNCOVER algorithm. Chapter 5 highlights the possibilities for computational speed-up through various tools, and as such these ideas are formed within the package function. Additionally, the package also provides a specific version of Chopin's IBIS algorithm on Bayesian logistic regression problems. The two main functions also support the visualisation of their respective outputs through tailored plotting functions which showcase aspects such as cluster assignment (only relevant for `UNCOVER`), posterior samples, fitted values and diagnostics. Finally, prediction is also possible, again through tailor made functions which predict responses of new covariate data through estimation of the posterior predictive distribution. For more detail on the package including basic examples, documentation can be found on the Comprehensive R Archive Network (CRAN) — `https://cran.r-project.org/web/packages/UNCOVER/index.html`.

CHAPTER 6

Application of UNCOVER

In section 4.6 a synthetic spiral dataset was presented, which highlighted the flexibility of the UNCOVER model when modelling data with non-linear cohorts that were not clearly separable in covariate space. Whilst a useful tool in revealing the potential of the UNCOVER method, this example will rarely replicate in real-world scenarios as we would expect considerably more noise. Therefore, this chapter aims to provide an application of UNCOVER to real-world data and the various challenges that arise when analysing non-synthetic problems. In particular this chapter will showcase UNCOVER's suitability in scenarios where the data may possess a clustering structure less clearly defined than the clustering structure seen previously in the spirals example.

Before practical examples are showcased we first present an examination of UNCOVER in the presence of increasing noise, be that noise in covariate space or noise in both covariate space and the regression signal, for a simulated dataset. Following on from this we present three real-world datasets (obtained from the UCI Machine Learning Repository [84]) in which aspects of UNCOVER can be tested; a wine dataset which has known clusters based on wine type [99] to test noise interference, an abalone dataset [100] which assesses misleading clustering structure and a heart

disease dataset [101] which assesses the role of categorical variables in UNCOVER.

6.1 Colliding Gaussians

For UNCOVER's main competitor, mixture of experts models, it is clear that non-linear covariate structures present a challenge in discovering the most appropriate clustering pattern. This is due to specification of the gating network, which can be adapted for non-linear clusters, but this requires either a hierarchical model (which requires a large amount of clusters to capture the non-linearity) or a non-linear gating network (which requires prior knowledge of the covariate structure which may be unreasonable in high dimensions).

What perhaps is less clear is the extent to which unsupervised methods (combined with sequential predictive modelling) underperform when compared to UNCOVER. Indeed, in scenarios where the clustering structure in the relationship between the response and the covariates is not synonymous with the clustering structure present in covariate space, unsupervised methods clearly fail, as seen in section 2.1.3. In situations where the clustering structure in covariate space is synonymous with that of the clustering structure in the relationship between the response and the covariates, one may hypothesise that a correctly specified unsupervised method[1] will perform at a similar level to that of UNCOVER. This, however, is dependent on level of separability of the clusters.

In real-world scenarios even with synonymous clustering structures we would still expect noise in the observations to cause some degree of overlap between clusters in covariate space. In order to test the effect of this overlap on clustering methods, we present the following scenario. Let $\mathbf{X} \in \mathbb{R}^{2000\times 2}$ be a covariate matrix such that for

[1]Even in the synonymous clustering structure scenario an ill-chosen clustering method can still perform poorly. A clear example of this is K-means on non-linearly separable clusters.

observation i:

$$\mathbf{x}_i \sim \begin{cases} \mathcal{N}\left((-3,-3)^T, \begin{pmatrix} 0.75 & 0.7 \\ 0.7 & 0.75 \end{pmatrix}\right) & \text{if } i \in \{1,\ldots,1000\} \\ \mathcal{N}\left((3,3)^T, \begin{pmatrix} 0.75 & 0.7 \\ 0.7 & 0.75 \end{pmatrix}\right) & \text{if } i \in \{1001,\ldots,2000\} \end{cases} \tag{6.1}$$

This covariate data represents two Gaussians which are unlikely to overlap, therefore creating a clear clustering structure in covariate space. We also insist upon each Gaussian having a different relationship between the covariates and the response, such that

$$\boldsymbol{\beta}_1 = (6,1,1)^T \tag{6.2}$$

$$\boldsymbol{\beta}_2 = (-6,1,1)^T \tag{6.3}$$

$$y_i \sim \begin{cases} \text{Bern}((1+e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_1})^{-1}) & \text{if } i \in \{1,\ldots,1000\} \\ \text{Bern}((1+e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_2})^{-1}) & \text{if } i \in \{1001,\ldots,2000\} \end{cases} \tag{6.4}$$

The results of the dataset can be seen in figure 6.1. Having generated the co-

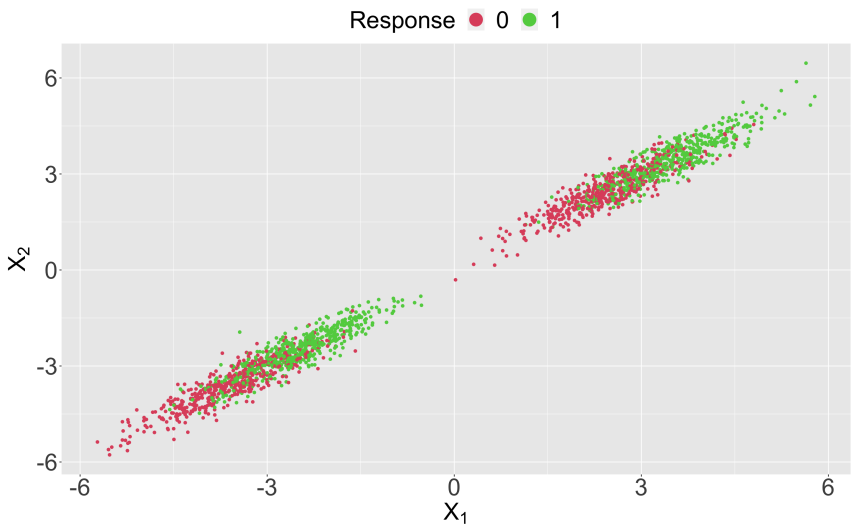


Figure 6.1: Base dataset for colliding Gaussians example. The response is shown through the colour of the points.

variates $\mathbf{X}$ and the response $\mathbf{y}$, we now fix these values and create a sequence of

datasets indexed by c, $(\mathbf{X}^c, \mathbf{y}^c)$ for $c \in [0, 3]$, through the following deterministic transformation on the original simulations:

$$\mathbf{x}_i^c = \begin{cases} \mathbf{x}_i + (c, c)^T & \text{if } i \in \{1, \ldots, 1000\} \\ \mathbf{x}_i - (c, c)^T & \text{if } i \in \{1001, \ldots, 2000\} \end{cases} \tag{6.5}$$

As $c \to 3$ the two Gaussians converge to completely overlap with each other, removing the clustering structure apparent in covariate space.

The specification of $\mathbf{y}^c$ is dependent on whether we consider noise in just the covariates (section 6.1.1) or in the covariates *and* the signal between the covariates and the response (section 6.1.2). For all examples, an 80 : 20 split of the data for each cluster is made to obtain training and test datasets[2]. This allows for the comparison of previously introduced unsupervised methods with UNCOVER, where we select a stopping criterion of $\varkappa = 4$ and a deforestation criterion of a minimum cluster size of 500 observations. In order for a fair comparison, for each unsupervised method we allow the number of clusters to range from one to five, with the optimal K selected through use of the gap statistic[3], using Tibrishani et.al's [17] method of selecting K.

6.1.1 Covariate Noise

As the noise we consider is only in the covariates, we retain the regression coefficients defined in equations (6.2) and (6.3). This, however, requires a resampling of the responses for each $c > 0$, and so we define $\mathbf{y}^c$ as:

$$y_i^c \sim \begin{cases} \text{Bern}((1 + e^{-(1,(\mathbf{x}_i^c)^T)\boldsymbol{\beta}_1})^{-1}) & \text{if } i \in \{1, \ldots, 1000\} \\ \text{Bern}((1 + e^{-(1,(\mathbf{x}_i^c)^T)\boldsymbol{\beta}_2})^{-1}) & \text{if } i \in \{1001, \ldots, 2000\} \end{cases} \tag{6.6}$$

A visualisation of the Gaussians when $c = 1.5$ is shown in the left-hand plot of figure 6.2.

[2]This split will be held constant throughout the sequence of datasets.
[3]This is valid here as a multivariate Gaussian distribution is log-concave.

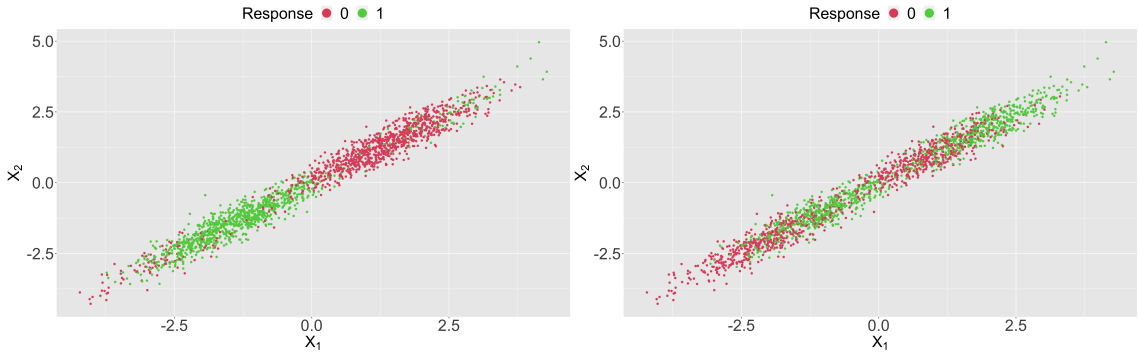


Figure 6.2: Covariates from the colliding Gaussian example when $c = 1.5$ (i.e one Gaussian has been translated by the vector $(1.5, 1.5)^T$ and one Gaussian has been translated by the vector $(-1.5, -1.5)^T$). The response $\mathbf{y}^c$ has either; been re-sampled for this value of c (left) or remain unchanged from the initial $\mathbf{y}$ representing a change in the coefficients (right). This is shown through the colour of the points.

The signal as a result remains despite the cluster collision in covariate space, and therefore as the data transitions from a synonymous clustering structure to a non-synonymous cluster structure (as when the Gaussians begin to overlap and there is no longer a clustering structure present in covariate space) the unsupervised methods should begin to fail. On the other hand, UNCOVER can utilise the response to still detect the true clustering structure even when there is some overlap present and therefore should outperform unsupervised methods for larger values of c. Note that for complete overlap it is unlikely that UNCOVER will perform well, as the minimum spanning tree structure used to initialise the method will not be able to distinguish between the two clusters. The Fowlkes–Mallows Index (FMI) for various methods, for increasing values of c, are given in figures 6.3 (for the training data) and 6.4 (for the test data). Note that for the one-cluster model $\frac{\text{TP}}{\text{TP+FP}} = 1$ and as the cluster sizes are equal TPR $= 0.5$, such that FMI $= \sqrt{0.5} \approx 0.707$. Therefore, what figures 6.3 and 6.4 reveal is that while all models can correctly identify the two clusters when they are trivially separable, the collapse to a one cluster model occurs much more rapidly when the response is not jointly modelled, leading to UNCOVER maintaining a high FMI value for larger values of c. The outlier here is when $c = 2.25$, however, the overlap between the two Gaussians at this point is substantial and so producing a two-cluster model which provides a better predictive model may result in a worse FMI value. Indeed, after this point the overlap is so

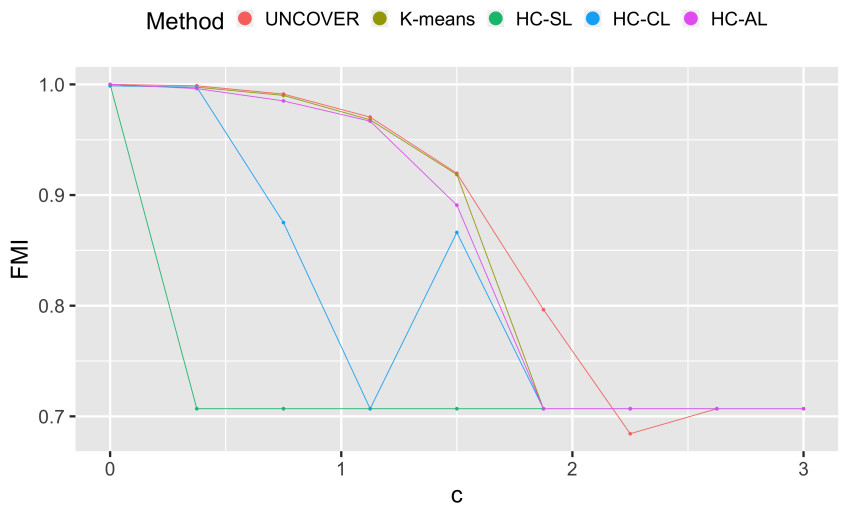


Figure 6.3: A comparison of unsupervised methods (K-means, Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), and Hierarchical Clustering — Average Linkage (HC–AL)) and UNCOVER's FMI values for differing values of c in the colliding Gaussian example with noise in just the covariates. FMI values are calculated for the training data.

considerable that even UNCOVER cannot detect a second cluster, and at $c = 3$ (complete overlap) there is no clear signal and therefore a one-cluster model is the most appropriate choice.

Regarding the AUC, the results are given in figures 6.5 (for the training data) and 6.6 (for the test data). As the two Gaussians collide, the difference in their relationship with the response remains intact, giving a clear two-cluster model. However, for a large enough value of c the overlap will be significant enough to distort the signal such that the two-cluster model can no longer be detected, resulting in a one cluster model being preferred. This occurs for $c \geq 2.625$ in this particular experiment[4]. This is not to say that a one cluster logistic regression model is accurately portraying the relationship between the response and the covariates. Indeed, as figures 6.5 and 6.6 show, the predictive power of a one cluster model for complete cluster overlap ($c = 3$) is much worse than the well separated scenario ($c = 0$) as when $c = 3$ the regression signals are completely distorted. As a result, an indica-

[4]Figures 6.5 and 6.6 appear to imply that a one cluster model is produced at $c = 2.25$. This is not the case, however, and is simply a result of the difference between a one cluster model and UNCOVER's two-cluster output for $c = 2.25$ being minimal in terms of the AUC.

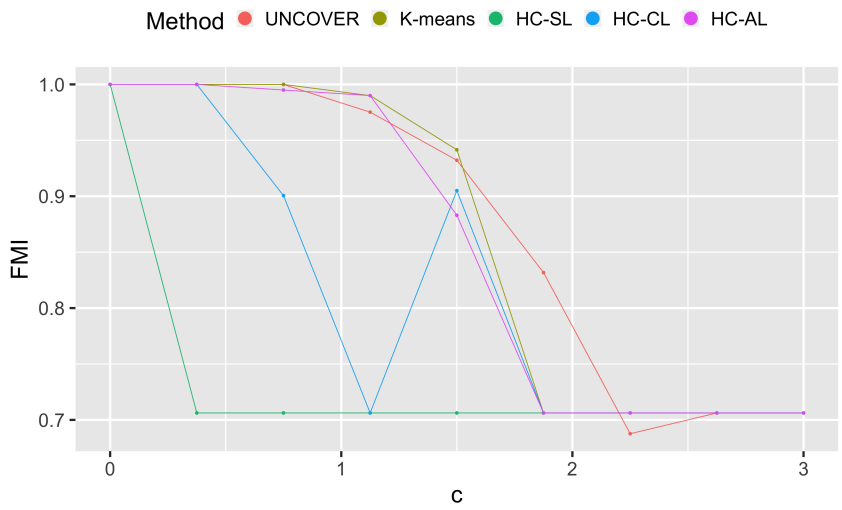


Figure 6.4: A comparison of unsupervised methods (K-means, Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), and Hierarchical Clustering — Average Linkage (HC–AL)) and UNCOVER's FMI values for differing values of c in the colliding Gaussian example with noise in just the covariates. FMI values are calculated for the test data.

tion of high performance is a method's ability to resist collapsing to a one cluster model for large values of c. In this regard UNCOVER can produce a two-cluster model (due to the incorporation of the response in cluster assignment) at values of c higher than any of the competing unsupervised methods (as these methods ignore the response for cluster assignment).

It is important to consider in this scenario the impact of the deforestation criterion. Indeed, selecting a minimum cluster size of 500 requires the regression signal for either cluster to be detectable for over half of the observations in said cluster (as the training data contains 800 observations of each true cluster). Therefore, whilst the collapse to a one cluster model occurs at $c = 2.625$ in this particular experiment, by reducing the minimum cluster size we could reasonably expect the value of c at which collapse occurs to be much closer to 3, with the natural trade-off being that there will be many misassigned observations for each 'true' cluster. One may even expect with a lower minimum cluster size for three clusters to form (two representing the two areas that are not overlapped and one completely overlapped cluster); however, this is rarely the case. If three clusters form, the 'overlapping' third cluster will have such a distorted signal that the Bayesian evidence will be very low. Given

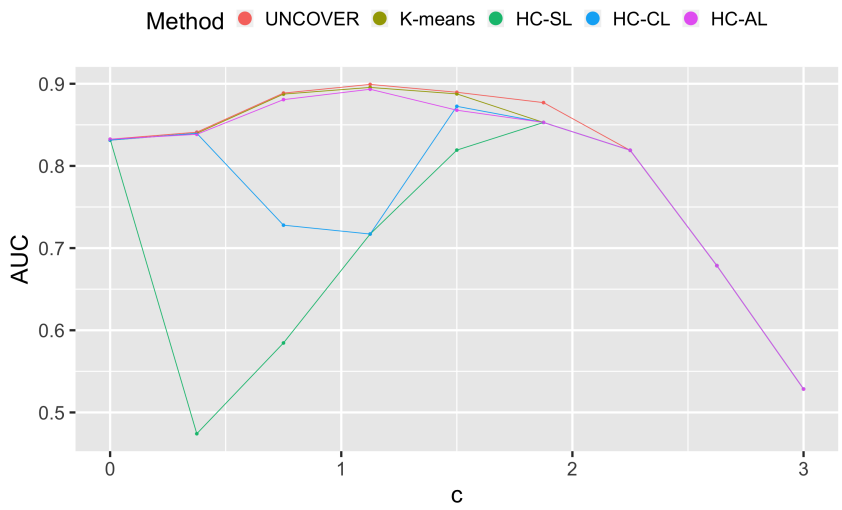


Figure 6.5: A comparison of unsupervised methods (K-means, Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), and Hierarchical Clustering — Average Linkage (HC–AL)) and UNCOVER's AUC values for differing values of c in the colliding Gaussian example with noise in just the covariates. AUC values are calculated for the training data.

this would occur when there is significant overlap, the number of observations in this 'bad' cluster would be large. Therefore, the formation of this cluster is unlikely to be beneficial to the entire model.

Finally, note that when c is large, the Gaussians operate in areas of the covariate space such that it is possible for the responses of a cluster to be of only one type (i.e. $y_i^c = 0$ if i corresponds to an observation in the first cluster and $y_i^c = 1$ if i corresponds to an observation in the second cluster). In this setting one may have concerns that the true cluster signals are lost due to the lack of response diversity. This setting does not occur in this experiment, however, if the clusters did only have one response type one would expect UNCOVER to return a one cluster model as it could not detect any regression signal in the true clusters. It should be highlighted, however, that this is only likely to occur at large values of c, and so a loss of signal due to response diversity is likely to coincide with a loss of signal due to overlapping. Therefore, this response diversity scenario results in similar outputs to the experiment showcased in figures 6.3—6.6.

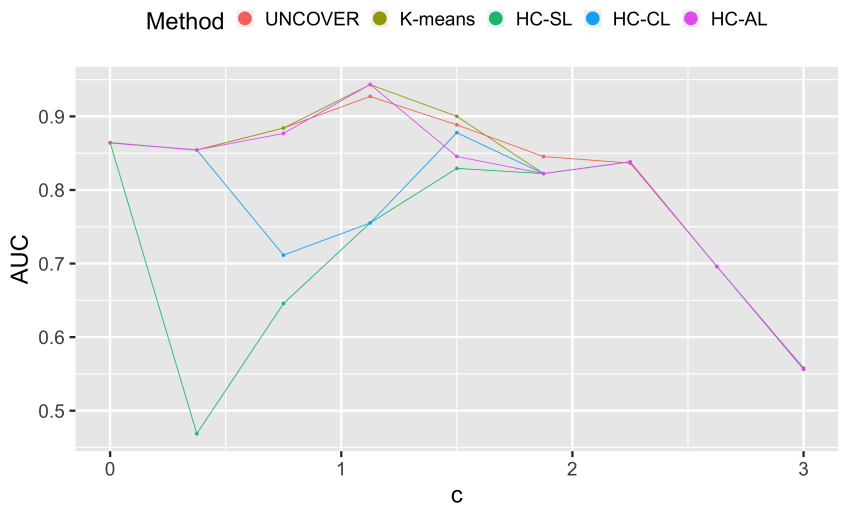


Figure 6.6: A comparison of unsupervised methods (K-means, Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), and Hierarchical Clustering — Average Linkage (HC–AL)) and UNCOVER's AUC values for differing values of c in the colliding Gaussian example with noise in just the covariates. AUC values are calculated for the test data.

6.1.2 Covariate & Signal Noise

Noise need not just be present in covariate space. Indeed, a separate form of noise can be found in the signal between the covariates and the response, and these forms of noise are not mutually exclusive. Therefore, the addition of signal noise is considered in the colliding Gaussian example through fixing the response as c increases, i.e.

$$y_i^c = y_i \quad \forall c, i \tag{6.7}$$

As $\mathbf{y}$ remains fixed, the regression coefficients must change with c in the following manner:

$$\boldsymbol{\beta}_1^c = (6 - 2c, 1, 1)^T \tag{6.8}$$

$$\boldsymbol{\beta}_2^c = (-6 + 2c, 1, 1)^T \tag{6.9}$$

A visualisation of the Gaussians when $c = 1.5$ is shown in the right-hand plot of figure 6.2.

This specification of the regression coefficients combined with the covariate shift

described in equation (6.5) allows for an experiment similar to the experiment showcased in section 6.1.1 to be conducted as $c \to 3$.

The results of this experiment, regarding the FMI, are in fact identical to the FMI results when there is just noise in the covariates. For the unsupervised methods this is to be expected as cluster assignment is not derived through consideration of the response for these methods, and the response is the only changing factor for these two experiments. Conversely, UNCOVER does rely on the response for cluster assignment so a change in output may be expected. However, this does not occur as although complete overlap has a different meaning for each experiment (with just covariate noise complete overlap represents two clusters occupying the same covariate space whereas when there is covariate and signal noise this represents a single cluster model) UNCOVER can only output a single model regardless. As a result, the transition from a two-cluster model to a one-cluster model is identical for the two experiments with regards to UNCOVER.

Whilst the FMI values are identical for both experiments, this is not the case when considering the AUC. This is due to a one cluster model clearly having a greater predictive power than a two-cluster model where the clusters are completely overlapping distorting the regression signals. Therefore, we showcase the AUC results for the covariate and signal noise experiment in figures 6.7 (for the training data) and 6.8 (for the test data).

As the two Gaussians collide, the difference in their relationship with the response lessens, leading to the one-cluster model at $c = 3$. Therefore, the strength of the joint clustering and cohort detection method here is how smoothly it can detect that transition, as a remnant of the differing regression signals will be apparent up until $c = 3$. In this aspect UNCOVER outperforms the entirety of the unsupervised methods, detecting the differing signals for large values of c (for example $c = 1.875$). Indeed, for large values of c UNCOVER is able to produce two clusters which perform better than the one-cluster model at predicting the response of both the training data and the test data. Out of the unsupervised methods considered here, K-means clearly is the superior method, which is due to the shape and uniformity of cluster size in this setting encouraging K-means to produce a split along the line $X_2 = -X_1$.

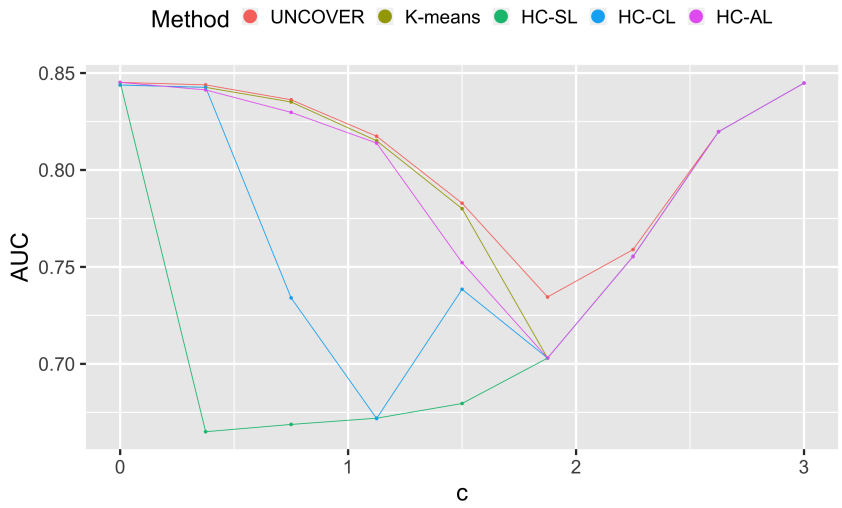


Figure 6.7: A comparison of unsupervised methods (K-means, Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), and Hierarchical Clustering — Average Linkage (HC–AL)) and UNCOVER's AUC values for differing values of c in the colliding Gaussian example with noise in the covariates and regression signals. AUC values are calculated for the training data.

However, even in this scenario which aids K-means clustering, the disregard of the response results in K-means producing a one cluster model before the clustering signals become undetectable (again the clear example of this is $c = 1.875$, when all methods bar UNCOVER produce a one cluster model). Finally, note that whilst we included the option of producing at most four clusters in the final output, no model's final output was more than 2 clusters at any point.

6.2 Wine Quality

The effect of overlapping clusters in covariate space can be detrimental for UNCOVER when attempting to detect underlying clustering structures, as seen in section 6.1. Whilst the example given in said section is hypothetical, there exists many real-world scenarios where this situation occurs. A prime of example of this is the wine quality dataset, where several physicochemical attributes of a sample of wine are given, with the task to determine the quality of sample. Quality is measured on a scale of 1 to 10, and so to reduce this to a binary response we state that wine samples given a score of 7 or above are classed as 'good' quality wine and samples

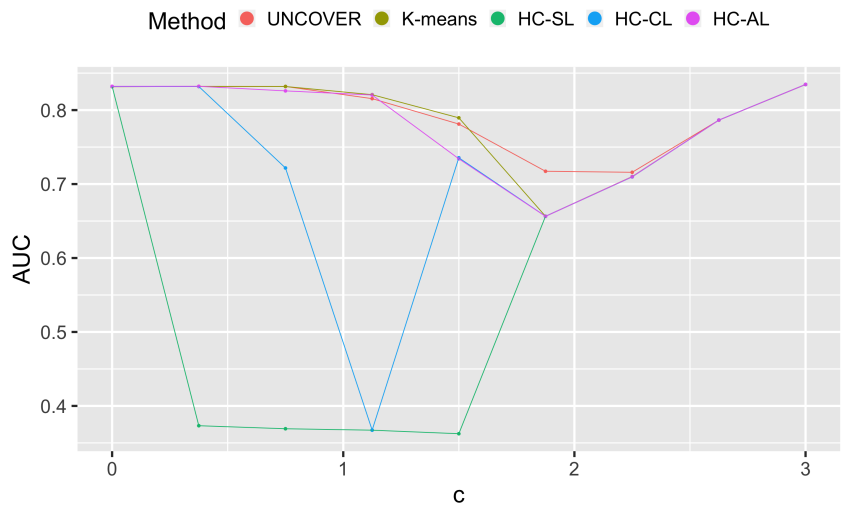


Figure 6.8: A comparison of unsupervised methods (K-means, Hierarchical Clustering — Single Linkage (HC–SL), Hierarchical Clustering — Complete Linkage (HC–CL), and Hierarchical Clustering — Average Linkage (HC–AL)) and UNCOVER's AUC values for differing values of c in the colliding Gaussian example with noise in the covariates and regression signals. AUC values are calculated for the test data.

given a score less than 7 are classed as 'bad' quality wine. Other variables in this dataset are summarised in table B.3, given in appendix B.2.2. The interpretation of colour from this table is that it should be considered as a covariate. This is a slight misnomer, as technically the attribute is derived from the fact that the wine quality dataset is formed from two datasets — one dataset for red wine samples and one dataset for white wine samples. In actuality, colour should be treated as a variable defining two clusters. Indeed, the behaviour between the physicochemical attributes and wine quality differs significantly depending on the colour of the wine. To highlight this point, figure 6.9 shows (for a selection of attributes) the difference in the posteriors for each colour when a Bayesian logistic regression model (with a standard normal prior) is chosen[5].

If access to the colour variable was not available, such that the clustering structure was no longer known a priori, one would hope that the UNCOVER method would be able to recover the wine colour clustering structure through use of the

[5]For the remainder of this section posteriors from the true clustering refers to this Bayesian logistic regression model with standard normal prior.

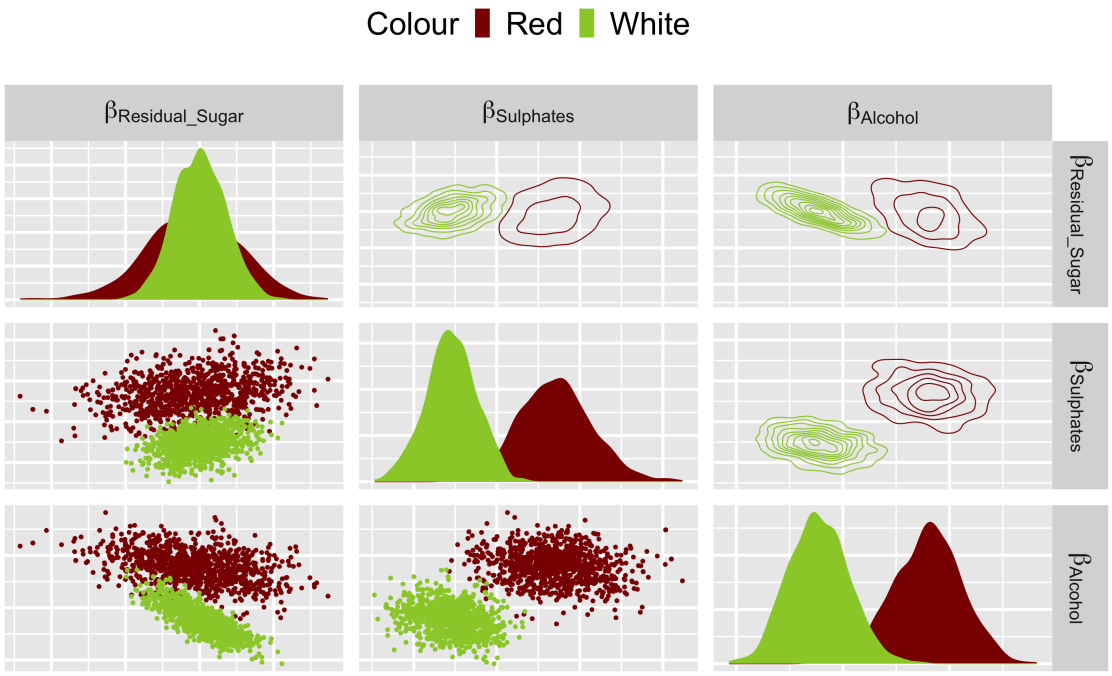


Figure 6.9: Posterior samples for the red wine dataset and the white wine dataset. Coefficients associated with covariates residual sugar, sulphates and alcohol are shown.

other variables in the dataset. This is not automatically guaranteed, however, as the remaining attributes may not be able to sufficiently separate the two types of wine in covariate space. This therefore represents a real-world scenario of cluster overlap or noise within the covariates.

As the Minimum Spanning Tree (MST) dictates the possible clusters formed in UNCOVER, the sub-selection of the attributes which are used to construct the MST (i.e. $\mathfrak{P}$) can be carefully selected to alleviate the potential overlap of the two true clusters. One possible method of selecting $\mathfrak{P}$ is to make use of the cut property (Lemma 3.3.1). Explaining further, for a complete graph cut such that the two sets created by the cut correspond to the two wine colour sets, the edge with the minimum weight (determined by distance) in the cut set must be in the MST. That is not to say, however, that other edges in this cut set will then be excluded from the MST, and indeed if more than one edge from the cut set is present in the MST then it is impossible for UNCOVER to produce clusters which directly correspond to the wine colour clusters. Therefore, if we can reduce the number of 'cut set' edges

in the MST through specification of $\mathfrak{P}$ we then will increase the separation of the two wine colour sets.

The number of possible combinations of $\mathfrak{P}$ is considerable ($2^{p=11} = 2048$) so, given an MST must be constructed for each combination, a less excessive method will be required instead of assessing all possible constructs of $\mathfrak{P}$. The method adopted is similar to that of forward selection [30]; we initially begin with $\mathfrak{P} = \emptyset$, then for each step we assess which variable's inclusion would reduce the amount of 'cut set' edges in the MST the most and add that variable to $\mathfrak{P}$. This method is formally given in algorithm 24. The end result of applying this greedy method is that seven attributes are selected to form $\mathfrak{P}$ (volatile acidity, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density and alcohol) which reduced the number of 'cut set' edges included in the MST to 32.

This specification of $\mathfrak{P}$ clearly fails to completely separate the two wine colour clusters entirely, presenting a challenge for the UNCOVER method. To examine the behaviour of UNCOVER in this setting, we specify a prior to be a multivariate normal whose mean and variance are derived from the mean and variance of the combined set of posterior samples showcased in figure 6.9. This ensures both true cluster posteriors are easily obtainable from the prior. Furthermore, we specify the deforestation criterion to be maximal regret, with a log maximal regret parameter of $\log(\nu) \approx 8$ (i.e. in the deforestation stage we will not restrict the reintroduction of edge $\{i, j\}$ if the resulting Bayesian evidence $Z^{\{i,j\}^+}$ satisfies the inequality $\log(Z^{\{i,j\}^+}) > \log(Z) - 8$ when compared to the current Bayesian evidence Z). Additionally, the stopping criterion is specified as $\varkappa = 5$.

The resulting output from this run was the uncovering of three clusters, summarised in table 6.1. The size and response distribution of the clusters indicate that overfitting has not occurred, and the colour distribution of the clusters clearly shows each cluster has a dominant colour, be that white for clusters 1 and 3 or red for cluster 2.

If the relationship between the covariates and the response are the same for clusters 1 and 3 this would represent a constraint of the UNCOVER method imposed through the structural restrictions of the MST. Indeed, whilst construction of an

Algorithm 24: Variable Selection for the Wine Quality Dataset

Input : *Covariate Matrix* — $\mathbf{X} \in \mathbb{R}^{n \times p}$, *Response Vector* — $\mathbf{y} \in \{0,1\}^n$,
Red Wine Observation Index Set — $\mathfrak{V}_{\text{red}}$,
White Wine Observation Index Set — $\mathfrak{V}_{\text{white}}$

Initialisation : Let $\mathfrak{P} = \emptyset$.

Step 1 : **for** $a = 1, \ldots, p$ **do**

Let $\mathcal{G}^a = (\mathfrak{V}, \mathfrak{E}^a)$ such that

$$\mathfrak{V} = \{1, \ldots, n\}$$
$$\mathfrak{E}^a = \{\{i,j\}; i \in \mathfrak{V}, j \in \mathfrak{V}, i \neq j\}$$

with edge $\{i,j\}$ having weight $e_{ij}^a = \|\mathbf{x}_{i,a} - \mathbf{x}_{j,a}\|_2$. Obtain minimum spanning tree $\mathfrak{T}^a$ from algorithm 6 and let Γ^a be the cardinality of the cut set, i.e.

$$\Gamma^a = |\{\{i,j\} \in \mathfrak{T}^a; i \in \mathfrak{V}_{\text{red}}, j \in \mathfrak{V}_{\text{white}}\}|$$

end

Add $b = \arg\min_{a=1,\ldots,p}\{\Gamma^a\}$ to $\mathfrak{P}$ and let $\Gamma = \Gamma^b$.

Step 2 : **for** $c \in \{1, \ldots, p\} \backslash \mathfrak{P}$ **do**

Let $\mathcal{G}^c = (\mathfrak{V}, \mathfrak{E}^c)$ such that

$$\mathfrak{V} = \{1, \ldots, n\}$$
$$\mathfrak{E}^c = \{\{i,j\}; i \in \mathfrak{V}, j \in \mathfrak{V}, i \neq j\}$$

with edge $\{i,j\}$ having weight $e_{ij}^c = \|\mathbf{x}_{i,\mathfrak{P}\cup\{c\}} - \mathbf{x}_{j,\mathfrak{P}\cup\{c\}}\|_2$. Obtain minimum spanning tree $\mathfrak{T}^c$ from algorithm 6 and let Γ^c be the cardinality of the cut set, i.e.

$$\Gamma^c = |\{\{i,j\} \in \mathfrak{T}^c; i \in \mathfrak{V}_{\text{red}}, j \in \mathfrak{V}_{\text{white}}\}|$$

end

Let $d = \arg\min_{c \in \{1,\ldots,p\}\backslash\mathfrak{P}}\{\Gamma^c\}$ and $\Gamma^d = \min_{c \in \{1,\ldots,p\}\backslash\mathfrak{P}}\{\Gamma^c\}$.

Step 3 : **if** $\Gamma^d \leq \Gamma$ **then**

Add d to $\mathfrak{P}$ and let $\Gamma = \Gamma^d$. **if** $\mathfrak{P} = \{1, \ldots, p\}$ **then**

Stop.

else

Go to Step 2.

end

else

Stop.

end

Result : *Covariate Indices Subset* — $\mathfrak{P}$

MST is necessary to reduce the number of possible partitions of the data and to capture the structure within the covariates, this process ensured that a complete separation of the true colour clusters was not possible at any stage of the algorithm.

Cluster	Size	Successes	Failures	Red	White
1	991	54	937	1	990
2	1344	175	1169	1324	20
3	2983	779	2204	34	2949

Table 6.1: Cluster summary information for the wine quality UNCOVER run. Successes and failures refer to the number of observations in the cluster whose associated quality score was good or bad respectively. Red and White refer to the number of observations in the cluster of said colour.

A consequence of this is a true cluster may be represented by several clusters in the output of UNCOVER. This is potentially problematic in scenarios where there is a cost associated with cluster-tailored intervention plans, as we then are creating an unnecessary cost for each additional cluster to the true clusters. However, if the relationship is different for clusters 1 and 3 then it is possible that UNCOVER may have detected further clustering structure within the white colour cluster.

To investigate this we first inspect the posterior distributions of the UNCOVER output, focusing on the same coefficients showcased in figure 6.9. This is shown in figure 6.10. Comparing the posteriors of the true clusters to the posteriors of the

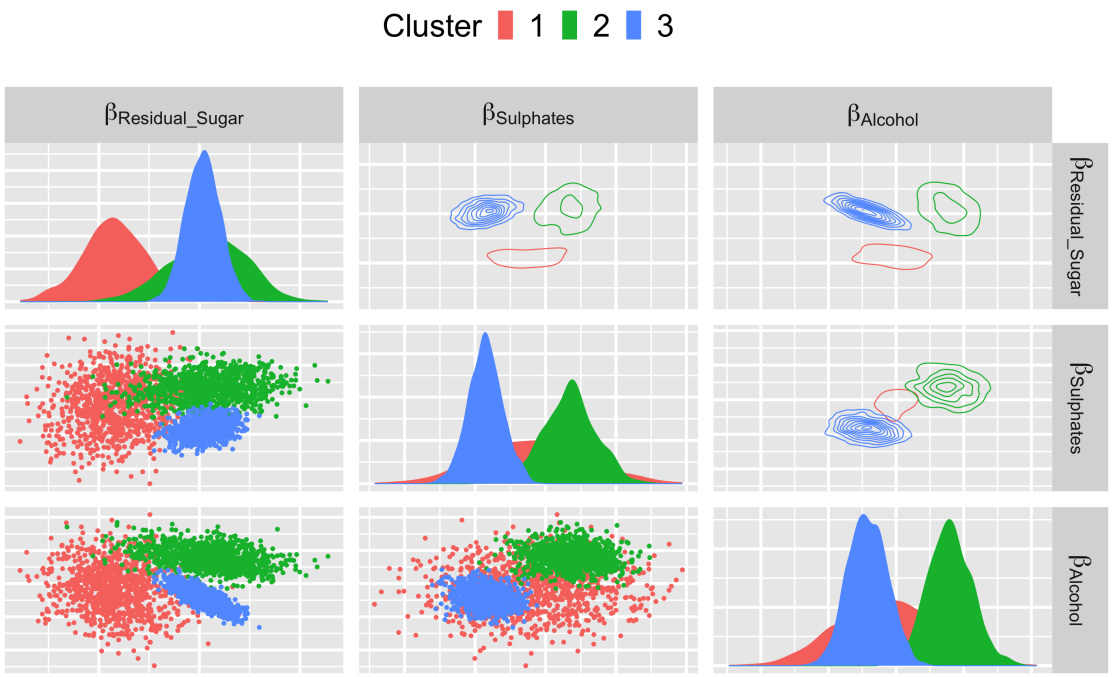


Figure 6.10: Posterior Samples for the clusters produced by UNCOVER on the wine quality dataset. Coefficients associated with covariates residual sugar, sulphates and alcohol are shown.

UNCOVER output, we note that clusters 2 and 3 appear to represent the red and white posteriors respectively, with cluster 1 showcasing different behaviour. From these figures though it is not clear if cluster 1 was formed to create a stronger concurrence between clusters 2 and 3 and the true clusters or if cluster 1 was formed as UNCOVER detected a genuine differing relationship. This can be evaluated through the predictive power of cluster 1 — using the posteriors from both the true clusters and the UNCOVER output we predict the probability of success (i.e. $Y = 1$) for observations assigned to cluster 1, then using these predictions we calculate the AUC for both sets of predictions. The AUC for the UNCOVER method being significantly higher for these observations suggests cluster 1 is a genuine cluster whilst the contrary suggests cluster 1 is only generated to further benefit one of the other two clusters. The results for this cluster as well as the other two clusters are found in table 6.2. From this table we can confidently state that we have detected a

Observations Assigned To	**UNCOVER AUC**	**True Clustering AUC**
Cluster 1	0.7433	0.6178
Cluster 2	0.8862	0.8858
Cluster 3	0.7956	0.7925

Table 6.2: AUC comparison table for the true clustering and the clustering produced by UNCOVER. The first column partitions the observations into the three UNCOVER clusters, then the AUC value for these observation's predictions against the response are given for both the UNCOVER method posterior (second column) and the true clustering posterior (third column).

further clustering structure within the white cluster, as we see a clear improvement in the AUC for cluster 1. This is further justified by the fact that neither of cluster 2 nor cluster 3 have benefited significantly from the generation of cluster 1.

In summary, whilst the presence of noise does remove the possibility for complete concurrence with true clustering structures, the incorporation of the response does allow for the UNCOVER method to attempt to produce the closest representation of the true clusters it can achieve with the structure given, and in some scenarios can even detect further clustering structure. This presents the UNCOVER method as a desirable choice in comparison with its supervised competitors. Indeed, one may assume that for noisy datasets finite mixtures of logistic regression models represent a suitable choice as they are not restricted by any covariate structure. However, as

discussed previously, clustering for these models is utilised as a mechanism within the algorithm and therefore this clustering cannot be applied to new data. Mixture of Expert (MoE) models on the other hand requires further integration.

In order to compete with UNCOVER, MoE models must produce a model with high predictive power and clusters which seemingly relate to the true clustering (this aids interpretability to stakeholders as wine colours are clearly distinguishable cohorts). To allow for a fair comparison with respect to the latter point we insist the clustering be derived in the same covariate space as was derived for UNCOVER (i.e. only using the covariates in $\mathfrak{P}$). This hinders MoE models predictive power as the 'experts' cannot use other covariates outside $\mathfrak{P}$ and so highlights an advantage of UNCOVER models as they can use different sets of covariates for different stages of the algorithm.

The results of a standard MoE model and a hierarchical MoE model are given in table 6.3 (the number of clusters is selected to be 3 for both models, with $\mathbf{K} = (2, 1)^T$ for the hierarchical model). From this table we can see that the clusters generated

MoE Method	Cluster	Red	White	AUC
Standard	1	670	2412	0.8412
	2	463	639	
	3	226	908	
Hierarchical	1	764	2099	0.8306
	2	524	501	
	3	71	1359	

Table 6.3: Cluster summary information for the wine quality MoE runs. Red and White refer to the number of observations in the cluster of said colour. AUC refers to the AUC when considering predictions for all observation responses.

by the MoE models do not bare any resemblance to the true clusters — potentially a result of the linearly separable property of these models not being able to sufficiently cope with the structure of the two-colour clusters. Furthermore, the AUC values of the two methods fall below the AUC value of the UNCOVER method (0.8418), albeit not substantially. In essence, by switching to a MoE model we remove the interpretability of the clusters for a slightly worse predictive model. Therefore, on this basis UNCOVER appears to be the most suitable choice for this noisy dataset — though in general one should be cautious of noise affecting the MST, as this

has the potential to force UNCOVER to produce multiple clusters with the same behaviour.

6.3 Abalone Age

One of the interesting features of the UNCOVER model is the automatic selection of the number of clusters and more specifically the possibility of outputting a one cluster model. This ability has been seen previously in section 6.1.2, where completely overlapping Gaussians in this scenario required a one cluster output. Outside of simulated examples, this setting occurs frequently, with many existing clustering algorithms failing to recognise an absence of clustering structure if the number of clusters must be pre-specified.

A dataset which highlights this issue of cluster selection clearly is the abalone dataset. Here the target variable Y is number of rings on an abalone's shell. The covariates X used in this analysis[6] are detailed in table B.4, given in appendix B.2.3.

Without background knowledge on the significance of rings on an abalone's shell, one may initially suspect that the sex covariate may contain clustering information. This can be visually examined through figure 6.11. From figure 6.11 it is clear that there is not a substantial difference in either the covariates or response when comparing males with females, which therefore does not suggest any difference between these two groups. However, the difference between infants and adults (with adults being male and females) in the covariates is substantial and therefore suggests a potential clustering structure.

We now make the assumption that sex in general is difficult to determine for abalones, and that future data would not necessarily contain information about sex. This prompts the omission of sex as an attribute in the model, which in turn gives a potential clustering structure which is not explicitly given but can be discovered through the other covariates.

[6]There are weight attributes other than whole weight contained within the abalone dataset, however, we make the assumption that due to ethical concerns (the measurement of the other weights require the abalone to not be alive) these attributes are unobtainable for future abalones and we therefore omit these covariates from our analysis.

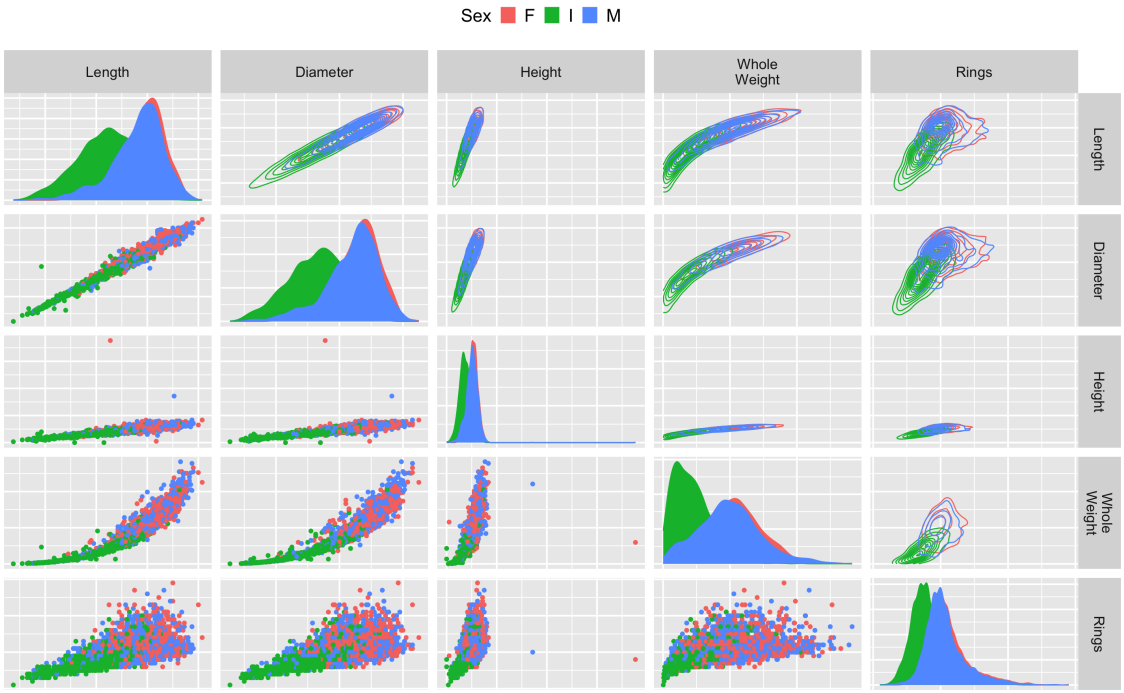


Figure 6.11: Abalone covariates and response (Rings). Points are coloured according to sex, with red points representing females, green points representing infants and blue points representing males.

In addition to this assumption, we would also have to create a binary response by classifying rings into two groups (for example this could be achieved by classifying based on whether the number of rings is greater than or less than a particular threshold). After obtaining binary response, we may naïvely conclude based on initial assessment that there are two clusters present in this dataset. For existing methods mentioned previously which do not automatically select the number of clusters (i.e. K-means clustering, hierarchical clustering, finite mixtures of logistic regressions and mixture of experts) a pre-selection of $K = 2$ would be the natural choice if we were to produce a model for this data. However, for this dataset the selection of any $K > 1$ would be unnecessary due to the context of the response. Indeed, the number of rings on an abalone's shell is directly related to age[7], as is the grouping of the sex attribute into infants and adults. The consequence of this is that for any response threshold chosen both infants and adults will have the same relationship between the covariates and the response; as the continuous attributes

[7] An abalone's number of rings plus 1.5 gives the abalone's age.

in the dataset all increase as an abalone grows with age, the relationship for both clusters will be that as an abalone psychically grows the probability of the abalone having a large number of rings increases. To highlight this point visually figure 6.12 shows the covariate's interaction with the response. From figure 6.12 it is clear

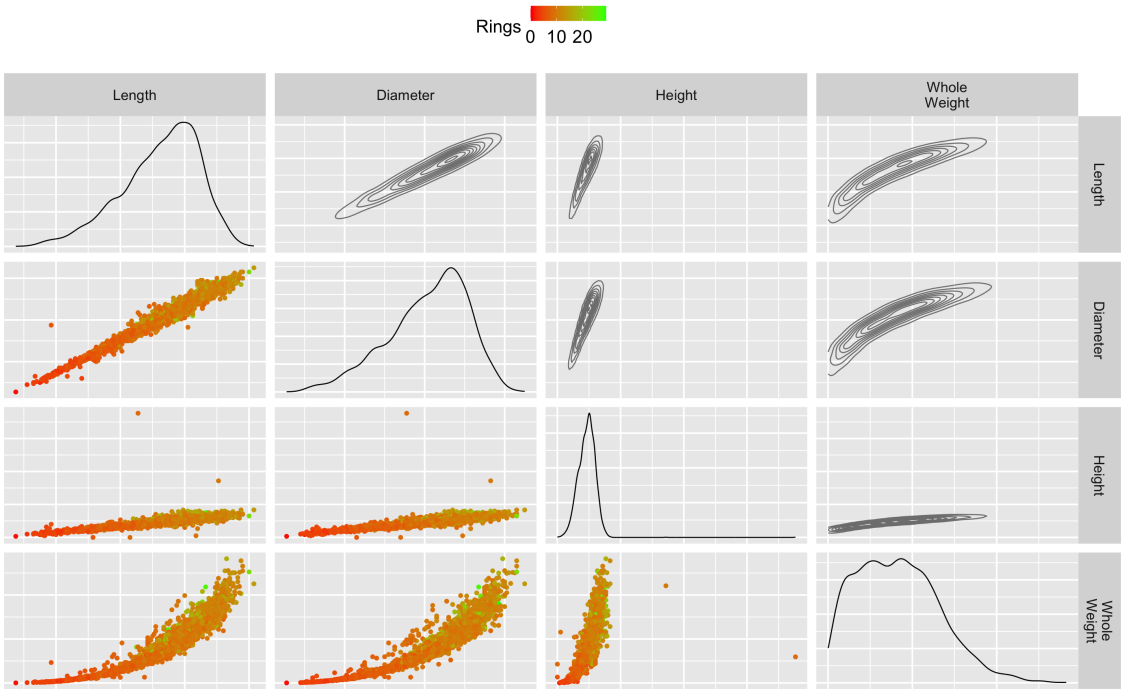


Figure 6.12: Abalone covariates. Points are coloured according to number of rings.

that the link between sex and ring number is such that a one cluster model is a better representation of the relationship between covariates and response[8] than a two-cluster model where the clusters correspond to infants and adults.

Having established intuitively that a one cluster model should explain the data better than a two cluster 'infant–adult' model, we can now test UNCOVER's ability to recognise this. The deforestation method selected is the response diversity criterion with minimum minority class factor $\upsilon = 100$[9]. In order to ensure a fair test of UNCOVER's capabilities, we set the number of rings threshold to 8, such that the

[8] Although the covariate structure is not strictly linear the increase in number of rings as the covariates increase appears linear and so a one cluster logistic regression model should perform well.

[9] The choice of υ is made to ensure that UNCOVER does not overfit the data, and indeed the both the infant (420) and adult (485) cluster's minimum number of observations with minority response class is well above this threshold.

response $Y = 1$ if the number of rings exceeds 8 and $Y = 0$ otherwise. Using this threshold allows the size of minimum minority response class to be 420 for infants and 485 for adults (both much larger than υ) and therefore if there are genuinely two clusters based on infants and adults, UNCOVER should be able to produce this as an output. We also specify that the stopping criterion $\varkappa = 4$ as opposed to $\varkappa = 2$ to allow UNCOVER to produce a $K > 2$ cluster model if it is deemed more beneficial than a one or two cluster model.

As hypothesised, with these specifications UNCOVER produces only a single cluster as an output, therefore recognising that a two cluster 'infant–adult' model, or indeed any model with a clustering structure, does not explain the data better than a no cluster model. Even the reduction of υ to only require a cluster to contain 10 observations with a minority response still produces a one cluster model. Therefore, we have confidence that the one-cluster model is genuinely the most appropriate model and not the result of an overly restrictive criterion.

6.4 Heart Disease & Incorporation of Categorical Variables

In previous sections categorical variables such as wine colour in section 6.2 or sex in section 6.3 are either treated as cluster information or variables which are excluded from the model. Seemingly this suggests that UNCOVER is unable to integrate categorical variables in the model construction process, which is not the case. Bayesian logistic regression models can clearly incorporate categorical variables through an appropriate specification of the design matrix. There is, however, a caveat when considering the variables which form the Minimum Spanning Tree (MST), as the Euclidean distance used to weight the edges of the graph is an unsuitable dissimilarity metric for categorical variables.

The Euclidean distance can of course be replaced by another more suitable metric to incorporate mixed type variables. However, as stated in section 4.1.1, the selection of a metric for mixed variables is challenging for general datasets.

A practical example of a dataset containing a mix of continuous and categorical

variables is the heart disease dataset, whose binary response Y is the presence ($Y = 1$) or absence ($Y = 0$) of heart disease for a patient, with the patients attributes[10] detailed in table B.5, given in appendix B.2.4.

Several attributes are categorical and therefore cannot be included in $\mathfrak{P}$ with the Euclidean distance metric, however, this alone does not justify their removal entirely. Indeed, complete exclusion of the categorical variables suggests that either categorical variables do not provide any information on the behaviour of the response or that categorical variables only provide generalised information on numerical attributes already present in the data. Neither of these statements can be applied to all datasets. To labour the point with this particular dataset, we conduct two runs of the UNCOVER algorithm with the Euclidean distance metric — one excluding the categorical covariates completely (the 'exclusion' model) and one excluding the categorical covariates from just $\mathfrak{P}$ (the 'inclusion' model as we are including the categorical covariates in the logistic regression models). Keeping the exact same specifications of the parameters for each UNCOVER run, an improvement on the Bayesian evidence of the 'inclusion' model over the 'exclusion' model suggests that one cannot simply exclude categorical variables from an UNCOVER model.

The specifications we give to UNCOVER are the following: the deforestation criterion is a minimum cluster size of 30 observations, a stopping criterion $\varkappa = 10$ and the MST variables are selected to be age, thalach and oldpeak. The results of the two runs are shown in table 6.4, showing that the inclusion of categorical variables allowed UNCOVER to detect a two cluster system which was not apparent when the categorical variables were excluded from the data. Furthermore, the

	Inclusion Model	**Exclusion Model**
Number of Clusters	2	1
Log Bayesian Evidence	-155.6319	-170.4736
Minimum Cluster Size	49	297
Minimum Minority Class	11	137

Table 6.4: Information on two runs of UNCOVER for the heart disease dataset, one including categorical variables and one excluding categorical variables.

[10]Some attributes in this dataset are omitted to produce a setting in which UNCOVER can detect a clustering structure.

second cluster detected does not appear to be a result of overfitting as there is a sufficient number of observations within the cluster (49) and a sufficient amount of observations whose corresponding response is in the minority response class (11). Regarding the Bayesian evidences, if one were to carry out a Bayesian hypothesis test that the inclusion model is an improvement upon the exclusion model using the Bayes factor and Jeffreys Scale [51, 53], then the result would be decisive support for the inclusion model over the exclusion model.

As a final point, note that the detection of a clustering structure is not the only benefit one can obtain from including categorical variables. Indeed, even in settings where both inclusion and exclusion models produce one cluster outputs, the inclusion of categorical variables may simply provide a model which explains the data more accurately[11]. This does not suggest, however, that all variables should be included in every situation, as UNCOVER is not exempt from the benefits of variable selection. Ultimately, the impracticality of including categorical variables in $\mathfrak{P}$ (when using the Euclidean distance metric) does not justify an ad-hoc selection of variables solely based on type (i.e. selection of only numerical attributes).

6.5 Summary

In real-world settings, UNCOVER is well adapted to provide a suitable output for a plethora of challenging situations. In this chapter, this has been demonstrated on one synthetic and three diverse real-world problems, which highlight effective performance in the face of a range of different challenges.

The most common amongst these challenges was addressed in the first two examples, that being the presence of noise causing overlap between the true clusters. Whilst UNCOVER is partially hindered by the distorted covariate structure due to the noise, the utilisation of the response ensures that the clusters outputted will still attempt to resemble the true clusters. This may result in additional clusters being generated which have the same behaviour as an existing cluster (but these

[11] This is the conclusion one arrives at if all attributes (i.e. introduction of the variables 'ca', 'cp' and 'thal' to the dataset) are utilised.

clusters cannot be combined due to the constraints of the MST structure). Indeed, UNCOVER was also shown to enable detection of clustering structure long after unsupervised methods collapse to a singular cluster. Another common challenge is the possibility of no clustering structure being present in the data, and in this case UNCOVER can automatically produce a one cluster model which accounts for this.

Finally, for datasets which contain a mixture of numerical and categorical variables, UNCOVER can proceed with a model which takes into account all variables by either restricting $\mathfrak{P}$ to only contain numerical covariates or (in some circumstances) by replacing the Euclidean distance metric (which defines similarity between observations) with a suitable alternative such as Gower's distance.

CHAPTER 7

Optimal Hold-out Sets: An Application in Updating Risk Scores

In this chapter we present the final contribution of this thesis, concerning an important and related topic to the previous chapters. This contribution addresses a more general problem, that being the practical implications of updating predictive models (like UNCOVER) where the previous incarnation has already been deployed and had an effect on the population. The problem this may have on model updating is discussed in section 7.1, with one solution to this being the use of hold-out sets.

A method for the specific size one should select for a hold-out set is proposed and a practical implementation through the use of emulation [102] is detailed. Given that interpretability is a core aim of UNCOVER and that the intention of this method is to be used in practical settings, it is important to consider the scenario of UNCOVER replacing an existing predictive model. This is done in section 7.4, as part of a wider consideration of the consequences of a clustering structure being present in the data used to construct these models.

The work done in this chapter formed part of a larger collaborative project with Louis Aslett, James Liley and Sami Haidar on the discovery of an optimal hold-out set size in the context of updating models [12]. However, the contributions detailed

after section 7.1 is the sole work of the author, unless stated otherwise.

For clarity of presentation, the notation in this chapter should be treated as self-contained.

7.1 Problem Outline

The use of UNCOVER was primarily motivated by uncovering cohorts of patients for the Scottish Population At Risk of Readmission and Admission (SPARRA) model, introduced in section 1.2. These statistically modelled cohorts could then be compared to expertly identified cohorts which were utilised in version 3 (v3) of the model. Note that version 4 (v4) of SPARRA did not include an explicit cohort clustering structure, and therefore implementation of UNCOVER within SPARRA would give a fifth version of the model.

This model updating may present a challenge, however, as the deployment of previous models is likely to have led to interventions being made by clinicians, on the basis of the predictions given by the most recent model they had available at the time. This creates a causal pathway between the covariates and the response through the previous model's predictions, confounding the output of the current model. An example of this would be the ASPRE score [103], a predictive model to detect the probability of a patient developing pre-eclampsia during pregnancy. For patients with a high risk of developing pre-eclampsia, aspirin can be prescribed to lower said risk. However, it is not advised that aspirin is universally applied as this treatment itself contains a slight risk of detrimental effects[1]. In light of this score, for high risk patients, medical professionals will intervene to prescribe aspirin. Now assume a new version of the model, say ASPRE 2, is developed (either through more accurate modelling techniques or population drift [12]). Due to the interventions made based on the previous model, individuals whose covariate values previously suggested a high risk of developing pre-eclampsia are now unlikely to have a corresponding response of pre-eclampsia developed, as they have been treated with

[1]This risk of these detrimental effects occurring is less than the risk of developing pre-eclampsia, but clearly in situations where a patient is extremely unlikely to develop pre-eclampsia then the risk of taking aspirin would outweigh the risk of developing pre-eclampsia.

aspirin. As a result, strong indicators of pre-eclampsia in the initial model are no longer present in the updated model, lowering the predictive power of the updated model even if the technique theoretically should lead to more accurate results. This particular case highlights a situation of 'better' models performing worse [13].

One could argue for using the same dataset to train the models for each new version, but there are scenarios in which this is not desirable, for example when distribution of the covariates and the response drifts [12] over time, independent of intervention effects. In this setting the data used for training the previous model would not be appropriate for training the current model. Letting $f_e(\cdot \mid \mathbf{X}_e, \mathbf{y}_e)$ be the predictor function from the version e model, trained on data $(\mathbf{X}_e, \mathbf{y}_e)$, when we come to build model $e+1$ we will in fact be modelling

$$Y_{e+1} \mid X_{e+1}, f_e(Y_{e+1} \mid X_{e+1}, \mathbf{X}_e, \mathbf{y}_e) \tag{7.1}$$

This highlights the fact that the response at the time model $e+1$ is being developed now depends on the previous model through interventions based on $f_e(\cdot \mid \mathbf{X}_e, \mathbf{y}_e)$. To further complicate matters, the actual intervention effects are unlikely to be known or quantifiable, and simply ignoring the previous model (known as naïve model updating) leads to a less accurate model [13].

A potential solution to this issue would be to build the models on a hold-out set in which no prediction is given, and so the responses for these observations when the next version of the model is constructed do not depend on any intervention effects from the current model. In more detail, we define an intervention set (X_e^i, Y_e^i) and a hold-out set (X_e^h, Y_e^h), which is used to create the predictor function $f_e(\cdot \mid \mathbf{X}_e^h, \mathbf{y}_e^h)$. This ensures that whilst Y_{e+1}^i will be dependent on $f_e(Y_{e+1}^i \mid X_{e+1}^i, \mathbf{X}_e^h, \mathbf{y}_e^h)$ as well as X_{e+1}^i, Y_{e+1}^h will only be dependent on X_{e+1}^h, i.e.

$$Y_{e+1}^i \mid X_{e+1}^i, f_e(Y_{e+1}^i \mid X_{e+1}^i, \mathbf{X}_e^h, \mathbf{y}_e^h) \tag{7.2}$$

$$Y_{e+1}^h \mid X_{e+1}^h \tag{7.3}$$

Therefore, construction of a new model based on the hold-out data will allow for the modelling of the desired system.

We make the fairly standard assumption here that all models produced provide some benefit to the population they are used on, and as such we intend to select as small a hold-out set as possible to minimise the number of individuals that do not benefit from the model. However, a hold-out set which is too small would mean we cannot accurately predict the response. Therefore, we require an optimal hold-out set size which can balance these two aspects. Letting $n^\star$ be the optimal hold-out set size, in order to devise a method to obtain $n^\star$ we must first define the costs associated with selecting a particular hold-out set size n. We let $C_1(X)$ be a random function of a random variable representing the cost for an individual who did not receive a prediction from the model (that is, the clinician acts only on other non-model information). The cost function $C_1(\cdot)$ is random as we assume there is not a deterministic approach all clinicians take for individuals. Therefore, the expected cost over the distribution of C_1 and X^h_{e+1} is defined as

$$k_1 = \mathbb{E}_{\pi(C_1, X^h_{e+1})}(C_1(X^h_{e+1})) \tag{7.4}$$

We also let $C_2(X \mid X^h_e, Y^h_e)$ be a random function of random variables representing the cost for an individual who did receive a prediction from the model (hence dependence on the hold-out data). As this cost now depends on the model built using the hold-out data, the cost is a function of the size of that set (only the size is relevant here as we take the expectation over the distribution of individuals in the hold-out set). Therefore,

$$k_2(n) = \mathbb{E}_{\pi(C_2, X^i_{e+1})}\left(\mathbb{E}_{\pi(X^h_e, Y^h_e)}(C_2(X^i_{e+1} \mid X^h_e, Y^h_e))\right) \tag{7.5}$$

Letting $\aleph$ be the total size of the population[2], the total cost of employing a hold-out set of size n, $\ell(n)$, is obtained through the addition of individual costs for members of the hold-out set and the intervention set, i.e.

$$\ell(n) = k_1 n + k_2(n)(\aleph - n) \tag{7.6}$$

[2]If the total population size is unknown but a specific dataset is available then we can take the $\aleph$ to be the size of the dataset.

The following two sections of this chapter will introduce a method of discovering $n^\star$, the minimiser of the function $\ell(n)$. This will be done by first making some assumptions on the system to enable a solution to be found, then attempting to minimise ℓ through the use of an emulator and the technique of expected improvement.

As justification for the use of emulation, we note that evaluation of $\ell(n)$ is likely to be expensive as many hold-out sets and models must be constructed to gain an accurate approximation of the costs, meaning we are limited in the number of evaluations one can make in search of the minimum. Additionally, both k_1 and $k_2(n)$ may require approximation through Monte Carlo estimation. This gives uncertainty in the values of k_1 and $k_2(n)$. Therefore, although $\ell(n)$ is deterministic, the process of evaluating $\ell(n)$ results in $\ell(n)$ giving possibly different results for the same value of n. Both of these issues can be tackled with an emulation framework.

7.2 Assumptions

In order to develop a method to discover the optimal hold-out set size, one must be convinced that such a size exists. Our solution space is restricted in the following ways; $n^\star \in \{0, \ldots, \aleph\}$. The scenario that we wish to eliminate here is that $n^\star = 0$ or $n^\star = \aleph$ as these solutions imply that either a hold-out set size should not exist or that the model should not exist. Note that at these extremes we have

$$\ell(0) = \aleph k_2(0) \tag{7.7}$$

$$\ell(\aleph) = \aleph k_1 \tag{7.8}$$

and therefore we can separate this issue into three cases.

7.2.1 $k_2(0) < k_1$

This setting corresponds to the scenario where the cost incurred by giving a member of the intervention set a prediction using a model trained on no data (for example a logistic regression model where the parameters are derived through expert opinion) is

less than the cost incurred by a member of the hold-out set not receiving a prediction at all. Here we need to assume the following:

Assumption 1: There exists $0 < ג < א$ such that $\frac{א-ג}{א}(k_1 - k_2(ג)) > k_1 - k_2(0)$.

Assumption 1 states that for some amount of data less than the entire population, the difference in expected cost between an individual who receives a prediction from this model and an individual who receives no prediction at all will be greater than the difference in expected cost between an individual who receives a prediction from a model based on no data and an individual who receives no prediction at all, by at least a factor of $\frac{א}{א-ג} > 1$. This appears to be a reasonable assumption as it simply insists that the model at some point justifies its hold-out size. This results in the following lemma, which proves the existence of an optimal hold-out set size:

Lemma 7.2.1 (Optimal Hold-out Set Size Existence for $k_2(0) < k_1$)**.** *Let Assumption 1 hold and let $k_2(0) < k_1$. Then there exists $0 < ג < א$ such that $\ell(ג) < \ell(0) < \ell(א)$.*

Proof. Note $k_2(0) < k_1 \implies אk_2(0) < אk_1 \implies \ell(0) < \ell(א)$. Also, note that $\frac{א-ג}{א}(k_1 - k_2(ג)) > k_1 - k_2(0) \implies גk_1 + (א - ג)k_2(ג) < אk_2(0) \implies \ell(ג) < \ell(0)$. As a result we have $\ell(ג) < \ell(0) < \ell(א)$. □

7.2.2 $k_2(0) = k_1$

This setting corresponds to the scenario where the cost incurred by giving a member of the intervention set a prediction based on no data is the same as the cost incurred by a member of the hold-out set not receiving a prediction at all. On the surface this situation appears rare but this can occur when personnel utilising the predictions simply ignore predictions based on no data or when members of the hold-out set are given no-data predictions as standard.

Theoretically this set-up is extremely similar to that seen before, and so again we rely on assumption 1, which reduces to the much weaker assumption that there exists $0 < ג < א$ such that $k_1 > k_2(ג)$. Indeed, all that is required here is that there is some amount of data less than the entire population where the expected cost to an individual who receives a prediction from the resulting model is less than the

expected cost to an individual who receives no prediction (or a prediction from a no-data model). The following lemma in this situation proves the existence of an optimal hold-out set size:

Lemma 7.2.2 (Optimal Hold-out Set Size Existence for $k_2(0) = k_1$)**.** *Let Assumption 1 hold and let $k_2(0) = k_1$. Then there exists $0 < ℶ < ℵ$ such that $\ell(ℶ) < \ell(ℵ) = \ell(0)$.*

Proof. Note $k_2(0) = k_1 \implies ℵk_2(0) = ℵk_1 \implies \ell(0) = \ell(ℵ)$. Also, note that $k_2(ℶ) < k_1 \implies (ℵ - ℶ)k_2(ℶ) < (ℵ - ℶ)k_1 \implies ℶk_1 + (ℵ - ℶ)k_2(ℶ) < ℵk_1 \implies \ell(ℶ) < \ell(ℵ)$. As a result we have $\ell(ℶ) < \ell(ℵ) = \ell(0)$. □

7.2.3 $k_2(0) > k_1$

This setting corresponds to the scenario where the cost incurred by giving a member of the intervention set a prediction based on no data is more than the cost incurred by a member of the hold-out set not receiving a prediction at all. Again here all that is required is the weak assumption that there exists $0 < ℶ < ℵ$ such that $k_1 > k_2(ℶ)$. This then gives the final existence lemma:

Lemma 7.2.3 (Optimal Hold-out Set Size Existence for $k_2(0) > k_1$)**.** *Let there exists $0 < ℶ < ℵ$ such that $k_1 > k_2(ℶ)$ and let $k_2(0) > k_1$. Then we have $\ell(ℶ) < \ell(ℵ) = \ell(0)$.*

Proof. Note $k_2(0) > k_1 \implies ℵk_2(0) > ℵk_1 \implies \ell(0) > \ell(ℵ)$. Also, note that $k_2(ℶ) < k_1 \implies (ℵ - ℶ)k_2(ℶ) < (ℵ - ℶ)k_1 \implies ℶk_1 + (ℵ - ℶ)k_2(ℶ) < ℵk_1 \implies \ell(ℶ) < \ell(ℵ)$. As a result we have $\ell(ℶ) < \ell(ℵ) < \ell(0)$. □

7.2.4 Summary

We have provided a thorough examination of the robustness of the assumption that an optimal hold-out size exists, in various settings depending on one's interpretation of $k_2(0)$. However, $k_2(0)$ is governed by the specification of the initial model and so in essence there is control over which case one may find oneself in. Indeed, one can force $k_1 = k_2(0)$ by giving members of the hold-out set a prediction using a model trained on no data instead of no prediction at all. Assuming this same model is

used for members of the intervention set when $n = 0$ we force the equality. This would allow for a much weaker assumption to be made, as discussed in section 7.2.2; that we do not require the entire population as training data to produce a beneficial model.

Aside from assumption 1, there is an implicit assumption we have made throughout this, that being:

Assumption 2: k_1 does not depend on n.

This on the surface appears to be a reasonable assumption, as at the very least members of the hold-out set do not receive a prediction based on any data, and therefore should not be reliant on the number of observations in a model which is not used on said members. However, there are scenarios worth considering in which implicitly k_1 depends on n. For example, when the personnel who are able to provide interventions subconsciously learn aspects of the model used on the intervention set and apply this to members of the hold-out set. One could argue that it is very unlikely that a human could completely learn a model's behaviour simply through exposure to the predictive element of the model. Indeed, it is worth noting that interventions can take place without reference to a model, and so even if this phenomenon is taking place in a minimal way the link between intervention and model is tenuous enough to assume this is a natural intervention and therefore simply contributing towards population drift [12]. Nevertheless, the possibility of the assumption being broken necessitates a solution, which can be achieved through blind interventions. That is, personnel who can make interventions receive a prediction for a member regardless of the set the member belongs to (as alluded to earlier this can be a prediction from a model trained on no data for hold-out set members), but crucially they are not informed which set the member belongs to. This then does not encourage subconscious learning of the behaviour of the model as the 'intervener' has no knowledge of which model is making the predictions. Of course, a substantially worse model for the hold-out set would increase the likelihood of the intervener deciphering which set a given individual belongs to based on their prediction, and so in this setting it is recommended that the no-training data model is constructed using expert opinion.

Finally, note that we have just proved the existence of an optimal hold-out set

size $n^\star$, and not that $n^\star$ is unique. Proof of uniqueness requires further assumptions to be made alongside an extension of the total cost function $\ell(\cdot)$[3] [12].

7.3 Emulation of $\ell(n)$

With the existence of a non-trivial optimal hold-out set size justified, we now require a principled method of locating such a size. First we note that the intervention set cost $k_2(\cdot)$ is governed entirely by the learning curve of the model selected [104] to be used on the 'held out' data. In this setting, the learning curve of a particular model is formed by obtaining the expected prediction error of the model given the amount of the training data used to construct the model (i.e. n). The shape of learning curves for certain models have been studied previously, such as decision trees [105], but in general the form of a learning curve for a particular model and population is difficult to obtain exactly. This is due to the fact that for any training dataset of size n, there is likely to be a substantially large number of different combinations of individuals that can form such a set. Therefore, in order to evaluate the expected prediction error at n we must obtain the prediction error for all of the possible training dataset configurations, which for any reasonably sized population is impractical.

In well-behaved settings one could imagine the expected prediction error of a model decreasing as the amount of training data increases. An example of this would be specification of $k_2(n)$ as

$$k_2(n) = a_1 n^{-a_2} + a_3 \tag{7.9}$$

where $\mathbf{a} = (a_1, a_2, a_3)^T$ is a vector of constants with $a_2 > 0$. However, it is no guarantee that the learning curve (and hence $k_2(n)$) behaves in this manner. Indeed, learning curves can display a non-montonic behaviour, for example displaying a double descent [104].

With an unknown learning curve shape, one may be tempted to evaluate $k_2(n)$ for all $n \in \{1, \ldots, \aleph - 1\}$ to determine the shape of the curve and therefore locate

[3] Currently this is only defined for $n \in \{0, \ldots, \aleph\}$.

the minimum. This brute force strategy is likely to be too expensive to practically implement, however; even if the cost was simulated the amount of simulations required would be significant. Indeed, simulation would require multiple evaluations for differing training datasets for each of the $\aleph - 1$ sizes. Instead, one solution is to represent our uncertainty about ℓ as a Gaussian Process [106], such that:

$$\ell \sim \mathcal{GP}\left(m(n), c(n, n') = \sigma_u^2 \exp\left\{-\left(\frac{n - n'}{\zeta}\right)^2\right\}\right) \tag{7.10}$$

where $m(n)$ is a mean function which typically we shall assume takes the desired form of equation (7.9), and σ_u^2 and ζ are hyperparameters of the covariance function $c(n, n')$, which takes the form of a exponentiated quadratic to allow for a smooth output. Specification of σ_u and ζ should be made on the prior beliefs of the uncertainty of $m(n)$ as a surrogate for $\ell(n)$ at specific values of n and the strength of correlation between different hold-out set sizes respectively. Recommendations for σ_u and ζ are problem specific, however, σ_u should be specified with consideration of our belief in the mean function's ability to replicate the behaviour of the total loss function. In addition to this, ζ should be specified with consideration of the size of the population, as hold-out sets of similar size should have highly correlated outputs whereas hold-out sets of drastically different sizes should not necessarily have highly correlated outputs. Bower et.al [107] provide a useful discussion on selection of these hyperparameters in one-dimensional settings, although even a poor specification of these hyperparameters can be rectified with continued evaluation of the the total loss function at different hold-out set sizes.

For a given value of n, evaluating the total cost $\ell(n)$ then provides data which can be used to update our beliefs on the form of the function (i.e. updating the Gaussian process emulator). Whilst $\ell(n)$ is deterministic, evaluation of $\ell(n)$ requires approximations of expectations, which has inherent variability. Typically we evaluate $\ell(n)$ through simulation[4], where given a hold-out set size n one follows the procedure below to obtain an estimate for $k_2(n)$:

[4]Though in situations where this is too expensive one could rely on expert opinion or a literature review. In this setting, equations (7.11) and (7.12) will have a different specification.

1. Sample n individuals from the population to form the hold-out set.

2. Construct model f using the hold-out set as training data.

3. Sample an individual from the population that is not already in the hold-out set and calculate the cost.

Repetition of these steps multiple times gives several values, and taking the average gives the Monte Carlo estimator of $k_2(n)$ that can be used to evaluate $\ell(n)$. Note that as k_1 does not depend on n this can be derived through expert opinion on the behaviour of the cost when no model is in place (or a no-data model is in place). Notationally, we define $\mathbf{n}$ as a vector of hold-out set sizes which corresponded to an evaluation of the total cost, with $\mathbf{n}^1$ as the vector of unique values in $\mathbf{n}$. The corresponding evaluations we label $\mathbf{d}$ and $\mathbf{d}^1$ respectively, where

$$d_i^1 = \frac{1}{|\{j : n_j = n_i^1\}|} \sum_{j:n_j=n_i^1} d_j \tag{7.11}$$

and finally we represent the variability of our individual estimations with the standard error $\boldsymbol{\sigma}^1$, where

$$\sigma_i^1 = \sqrt{\frac{1}{(|\{j : n_j = n_i^1\}| - 1) \times |\{j : n_j = n_i^1\}|} \sum_{j:n_j=n_i^1} (d_j - d_i^1)^2} \tag{7.12}$$

Clearly this specification requires at least two evaluations to be made for a particular value of n_i^1 (i.e. $|\{j : n_j = n_i^1\}| > 1$)[5].

The standard treatment of variability in evaluation is treated through the introduction of a nugget term [106] to our emulator, which is a secondary Gaussian process with 0 mean function and constant variance. This is not appropriate, however, in this setting for two reasons. The first reason is that we would not expect the variance of our estimate to be constant in n (i.e. the variance is a function of the hold-out set size: $\text{Var}(d_i^1) = g(n_i^1)$). Indeed, proof of the variance of d_i^1 depen-

[5]If one had prior knowledge on the variance of evaluations, $\text{Var}(d_i^1)$ then two evaluations are not required as σ_i^1 would become $\sqrt{\frac{\text{Var}(d_i^1)}{|\{j:n_j=n_i^1\}|}}$.

dence on n_i^1 can be witnessed through consideration of the extreme values of n_i^1. For large values of n_i^1 model fits should be more stable, resulting in consistent model predictions which in turn results in total cost evaluations which have low variability. In contrast, small values of n_i^1 give less stable model fits, inconsistent model predictions and consequently higher variability in the total cost evaluations. Even allowing for non-constant variance in n, a nugget term would be misleading for a second reason; nugget terms account for noise in the evaluation due to the exclusion of explanatory variables, which here would be the hold-out data, intervention data and their respective costs. However, whilst inclusion of these would create a deterministic function, this function would not be $\ell(n)$ as clearly $\ell(n)$ only relies on the size of the hold-out set. In addition to these reasons, the use of a nugget term does not allow for multiple evaluations to reduce the variance in our evaluations. Indeed, the inherent variance in single evaluations of the total cost at n_i^1 is mitigated in our emulator by taking the average of multiple evaluations at n_i^1; therefore σ_i^1 decreases as $|\{j : n_j = n_i^1\}|$ increases. This technique of variance reduction is not available if we capture the variability with a nugget term.

Referring back to our evaluations $\mathbf{d}^1$, we represent this variability by stating that $d_i^1 = \ell(n_i^1) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, (\sigma_i^1)^2)$. This then gives the following joint distribution

$$\begin{bmatrix} \ell(n) \\ \mathbf{d}^1 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(n) \\ m(\mathbf{n}^1) \end{bmatrix}, \begin{bmatrix} c(n,n) & c(n,\mathbf{n}^1) \\ c(\mathbf{n}^1,n) & c(\mathbf{n}^1,\mathbf{n}^1) + \text{diag}\{(\boldsymbol{\sigma}^1)^2\} \end{bmatrix} \right) \tag{7.13}$$

where $m(\mathbf{n}^1)_i = m(n_i^1)$, $c(n,\mathbf{n}^1)_i = c(\mathbf{n}^1,n)_i^T = c(n,n_i^1)$, $c(\mathbf{n}^1,\mathbf{n}^1)_{ij} = c(n_i^1,n_j^1)$ and $\text{diag}\{(\boldsymbol{\sigma}^1)^2\}_{ij} = (\sigma_i^1)^2 \mathbb{1}(i=j)$. Equation (7.13) can then be used to specify the conditional distribution of $\ell \mid \mathbf{d}^1$:

$$\ell \mid \mathbf{d}^1 \sim \mathcal{GP}(\mu(n), \Psi(n)) \tag{7.14}$$

$$\mu(n) = m(n) + c(n,\mathbf{n}^1)\left[c(\mathbf{n}^1,\mathbf{n}^1) + \text{diag}\{(\boldsymbol{\sigma}^1)^2\}\right]^{-1}\{\mathbf{d}^1 - m(\mathbf{n}^1)\} \tag{7.15}$$

$$\Psi(n) = c(n,n) - c(n,\mathbf{n}^1)[c(\mathbf{n}^1,\mathbf{n}^1) + \text{diag}\{(\boldsymbol{\sigma}^1)^2\}]^{-1}c(\mathbf{n}^1,n) \tag{7.16}$$

$\mu(n)$ and $\Psi(n)$ are known as the Bayes linear update equations [106], and these two concepts will encapsulate the emulator ℓ when locating the minimum.

7.3.1 Expected Improvement

Given the Bayes linear update equations, we have a process for the addition of total cost evaluations to better understand the behaviour of the function $\ell(n)$. Indeed, each evaluation (i.e. addition to $\mathbf{d}$) improves our understanding of the true function by either observing the behaviour of the function at a new point or decreasing the standard error of a previously considered value of n.

The question that remains, however, is which value of n to evaluate next? Selection of n for evaluation clearly must aid in locating the minimum, and so we must balance exploration with exploitation. Exploration of the different sizes is important for gaining information about the total cost function, but equally important is the exploitation of values of n which give low total cost. A popular choice of metric [108] for assessing the viability of a point to be evaluated is the improvement function

$$\text{Imp}(n) = \max\{0, d^- - \ell(n)\} \tag{7.17}$$

where $d^- = \min\{\mathbf{d}^1\}$ represents a fixed 'known' minimum. Clearly maximising this function would be suitable for discovering the minimum if we could evaluate $\ell(n)$ for each n. However, we are in the scenario where we assume this is impractical and so the expected value of this function is required instead. Therefore, letting $\beth = d^- - \ell(n)$ and noting that for n we have $\beth$ under $\ell \mid \mathbf{d}^1$ to be $\mathcal{N}(d^- - \mu(n), \Psi(n))$, taking the expectation with respect to $\ell \mid \mathbf{d}^1$ gives:

$$\begin{aligned}\mathbb{E}_{\ell|\mathbf{d}^1}(\text{Imp}(n)) &= \int_0^\infty \beth \, \pi_{\ell|\mathbf{d}^1}(\beth) d\beth \\ &= \int_{\frac{-(d^- - \mu(n))}{\sqrt{\Psi(n)}}}^\infty \left[\sqrt{\Psi(n)} z + d^- - \mu(n)\right] \phi(z) dz \end{aligned} \tag{7.18}$$

where $z \sim \mathcal{N}(0,1)$ and $\phi(\cdot)$ is the pdf of a standard normal. This integral can be solved analytically [109], and gives the following 'expected improvement' function

$$\text{EI}(n) \coloneqq \mathbb{E}_{\ell|\mathbf{d}^1}(\text{Imp}(n)) = (d^- - \mu(n))\Phi\left(\frac{d^- - \mu(n)}{\sqrt{\Psi(n)}}\right) + \sqrt{\Psi(n)}\phi\left(\frac{d^- - \mu(n)}{\sqrt{\Psi(n)}}\right) \tag{7.19}$$

in which to maximise. Note that in rare situations where $\Psi(n) = 0$, $\text{Imp}(n)$ must be 0, and therefore $\text{EI}(n) = 0$.

We therefore proceed as follows; starting with an initial set of evaluations $\mathbf{d}^1$ and their respective standard errors $\boldsymbol{\sigma}^1$, calculate $\text{EI}(n)$ for $n = 1, \ldots, \aleph - 1$ and locate the maximum. Then evaluate the value of n which gave this maximum, updating $\mathbf{d}^1$ and $\boldsymbol{\sigma}^1$. As this process is repeated we gain more information on the true location of the minimum of the total cost. Whilst the infinite repetition of this process will guarantee the location of the minimum is found and that $\text{EI}(n) = 0$ for $n = 1, \ldots, \aleph - 1$ [12], this is clearly not practical and so we introduce a stopping criterion which is a threshold η_1 such that when $\max_{n=1,\ldots,\aleph-1}\{\text{EI}(n)\} < \eta_1$ we stop and take the current value $n^1_{\arg\min_i\{\mathbf{d}^1\}}$ as the optimal hold-out set size $n^\star$.

Introducing such a criterion may lead to the selection of a hold-out size n^1_i whose corresponding mean evaluation, d^1_i, has large standard error. This can also be true of other hold-out sizes, with large variability of output potentially meaning the true optimal hold-out set is not selected. To mitigate this problem, when the stopping criterion has been met, giving d^-, we insist on another criterion to bring an end to the process; that being that for all $n^1_j \in \mathbf{n}^1$, $d^1_j - 3\sigma^1_j > d^- \cup \sigma^1_j < \eta_2$ where η_2 is a constant. If this criterion is not met then we evaluate the process again at all n^1_j who failed the criterion, update d^1_j and restart the expected improvement process. This method is detailed in algorithm 25.

As mentioned previously, minimisation through expected improvement has the benefit of providing a balance between exploration and exploitation, justifying its use over other 'acquisition functions' (functions which aid in acquiring the minimum) such as the probability of improvement [108]. This balance is evident through consideration of what values of n produce large values of $\text{EI}(n)$. For values of n where the posterior mean is much lower than the posterior mean of our current known optimal size then $\text{EI}(n)$ will be large (i.e. we exploit values of n where we are confident of discovering an improvement). Alternatively, for unexplored values of n with high posterior variance $\text{EI}(n)$ will also be large, encouraging exploration. Generally if our current known optimal size has high variance this will also encourage exploration due to our lack of confidence in the accuracy of our current estimate.

Algorithm 25: Total Cost Minimisation Through Expected Improvement

Input : *Multiset of Sizes* — $\mathbf{n}$, *Set of Unique Sizes* — $\mathbf{n}^1$, *Total Cost Evaluations* — $\mathbf{d}$, *Total Cost Evaluations for Unique Sizes* — $\mathbf{d}^1$, *Standard Errors* — $\boldsymbol{\sigma}^1$, *Mean Function* — $m(n)$, *Stopping Threshold* — η_1, *Covariance Function Hyperparameters* — (σ_u, ζ), *Variance Threshold* — η_2, *Minimum Number of Evaluations Per Size* — $\tau > 1$

Initialisation : Let $c(n, n')$ be defined as in equation (7.10). Let

$$\mu(n) = m(n) + c(n, \mathbf{n}^1)\left[c(\mathbf{n}^1, \mathbf{n}^1) + \text{diag}\{(\boldsymbol{\sigma}^1)^2\}\right]^{-1}\{\mathbf{d}^1 - m(\mathbf{n}^1)\}$$
$$\Psi(n) = c(n, n) - c(n, \mathbf{n}^1)[c(\mathbf{n}^1, \mathbf{n}^1) + \text{diag}\{(\boldsymbol{\sigma}^1)^2\}]^{-1}c(\mathbf{n}^1, n)$$

Step 1 : Let $d^- = \min\{\mathbf{d}^1\}$.

for $n = 1, \ldots, \aleph - 1$ **do**

Calculate

$$\text{EI}(n) = (d^- - \mu(n))\Phi\left(\frac{d^- - \mu(n)}{\sqrt{\Psi(n)}}\right) + \sqrt{\Psi(n)}\phi\left(\frac{d^- - \mu(n)}{\sqrt{\Psi(n)}}\right)$$

end

Step 2 : **if** $\max_{n=1,\ldots,\aleph-1}\{EI(n)\} < \eta_1$ **then**

Let

$$\tilde{n} = \{n_j^1 \in \mathbf{n}^1 : d_j^1 - 3\sigma_j^1 \leq d^- \cap \sigma_j^1 \geq \eta_2\}$$

if $\tilde{n} = \emptyset$ **then**

Stop.

end

else

Let $\tilde{n} = \arg\max_{n=1,\ldots,\aleph-1}\{\text{EI}(n)\}$.

end

Step 3 : **for** $j \in \tilde{n}$ **do**

if $j \in \mathbf{n}^1$ **then**

Add j to $\mathbf{n}$, approximate the total cost at this size and add this to $\mathbf{d}$. Update $\mathbf{d}^1$ and $\boldsymbol{\sigma}^1$.

else

Add j to $\mathbf{n}$ τ times. Approximate the total cost at this size τ times and add this to $\mathbf{d}$. Calculate the average of these τ evaluations along with their standard error and add these to $\mathbf{d}^1$ and $\boldsymbol{\sigma}^1$ respectively. Update $\mathbf{n}^1$.

end

end

Step 4 : Update $\mu(n)$ and $\Psi(n)$. Go to step 1.

Result : *Minimum Hold-out Set Size* — $n^\star = n^1_{\arg\min_i\{\mathbf{d}^1\}}$

7.3.2 Random Forest Example

The application of optimal hold-out sets need not be restricted to medical settings or indeed be applied to data with clustering structure — the scope of hold-out sets as a solution to model updating considers all predictive models in which interventions are possible. To showcase this we take the car dataset [84], which contains 1728 observations and has covariates describing the features of a car detailed in table B.6, given in appendix B.2.5. The response variable describes the acceptability of the car, which can have possible values of 'Not Acceptable', 'Acceptable', 'Good' or 'Very Good'. However, for the purpose of this example we reduce this to a binary choice of either not acceptable or other, as the key response of interest is when a car's quality is not acceptable.

Constructing a model to predict acceptability will then allow for a pre-evaluation of future cars, with cars predicted as unacceptable intervened upon such that for the actual evaluation the majority of cars then get classified as acceptable or better (i.e. good or very good). An example of this process would be if safety was deemed a key predictor is car acceptability — for cars which are predicted to be not acceptable interventions can be made to improve the safety such that for the actual evaluation the car is deemed acceptable or better. With regards to model updating, without a hold-out set this presents an issue as in this example consistent intervention on safety results in the next model not being able to identify safety as a key predictor, as we no longer have data which is informative on poor safety leading to unacceptable cars.

This showcases the need for hold-out sets in this setting, and so the procedure detailed in section 7.3 can be carried out given a model. The model chosen for this example is a random forest model [30], an ensemble method which combines several decision trees. Random forests can be used in binary classification settings as a predictive model, as for a given observation $\mathbf{x}$ individual decision trees can predict the response to be either a success (i.e. $y = 1$) or a failure (i.e. $y = 0$), with the random forest prediction being the majority vote.

Taking a random initial set of sizes to form $\mathbf{n}^1$, we then replicate each of these

sizes 10 times to form $\mathbf{n}$. Evaluation of the system for $\mathbf{n}$ and then taking the average costs along with their respective standard errors will give $\mathbf{d}$, $\mathbf{d}^1$ and $\boldsymbol{\sigma}^1$. In order to do this, however, we require specification of k_1 and $k_2(n)$. We define cost to a particular individual in the intervention set who received a particular prediction through table 7.1. We assume no model is given to the hold-out set and that the

	Predicted Output	
Actual Output	True Positive (TP) = 0.5	False Negative (FN) = 1
	False Positive (FP) = 0.5	True Negative (TN) = 0

Table 7.1: Cost matrix for a member of the intervention set. Note this is based on the confusion matrix given in table 4.1.

expected cost is $k_1 = 0.5$. The mean function $m(n)$ we base on equation (7.9) and so has the form

$$m(n) = k_1 n + (a_1 n^{-a_2} + a_3)(\aleph - n) \tag{7.20}$$

where a_1, a_2 and a_3 are derived by fitting the curve $k_2(n) = (a_1 n^{-a_2} + a_3)$ to the evaluated expected cost to individuals in the intervention set. This gives $a_1 = 24.4335334, a_2 = 0.8805692$ and $a_3 = 0.2798915$. For the covariance function hyper-parameters[6] we specify $\sigma_u = 100$ and $\zeta = 150$. Finally, we specify $\eta_1 = 1$, $\eta_2 = 10$ and $\tau = 10$. The resulting emulator for the initial data is given in figure 7.1 alongside the expected improvement function for this initialisation.

As expected, our initial emulator behaves in a similar manner to that of the mean function for the initial evaluations, and these evaluations taken at certain values of n indicate that there does exist a hold-out set size that minimises the total cost. The effect of incorporating $\boldsymbol{\sigma}^1$ into our emulator is also showcased here, as the posterior emulator variance $\Psi(n)$ is greater than zero even at the evaluated sizes. For larger hold-out set sizes as discussed previously there is less variability in the evaluations due to the stability of model and so $\Psi(n)$ will be considerably smaller as $n \longrightarrow 1728$.

Regarding the expected improvement function, the initial need for exploration can be seen to dominate here as there is a large region of the input space yet to be

[6]These hyperparameters can be specified in this setting through a grid search of plausible values. In absence of a natural objective to optimise, evaluation of these values can be carried out through visual inspection (as in the left sided plot of figure 7.1) of the emulator's suitability to the initial data $\mathbf{d}^1$.

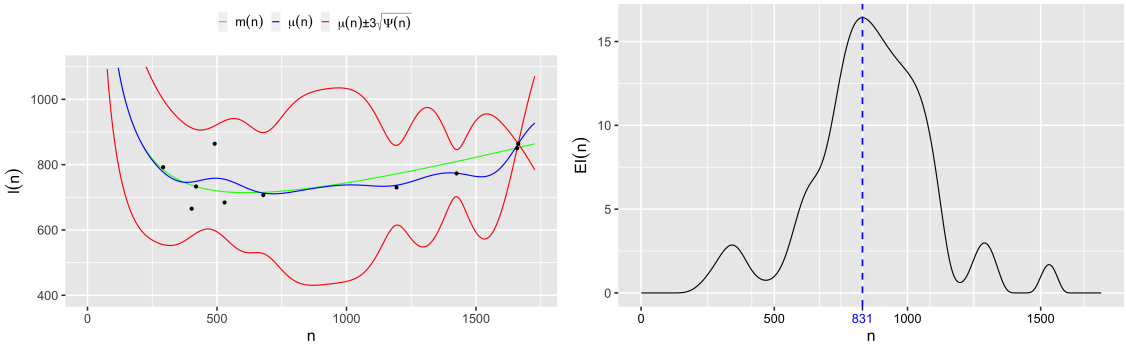


Figure 7.1: Initial emulator (left) of the car random forest emulation example along with the initial expected improvement function (right). For the left-hand plot black points represent $\mathbf{d}^1$ and for the right-hand plot the blue dashed line highlights the maximum of EI(n).

explored (namely the subset $\{680, \ldots, 1192\}$). Exploitation of current evaluations (or sizes near to current evaluations) will occur through maximising EI(n) after evaluations of the unexplored region have been made.

Running algorithm 25 then gives the emulator output seen in figure 7.2, which gives a final optimal hold-out size as $n^\star = 235$. Whilst the general behaviour of

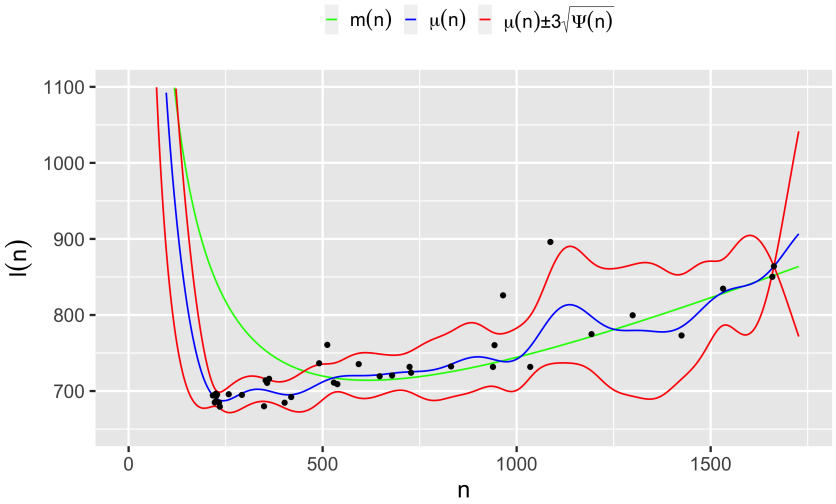


Figure 7.2: Outputted emulator from running algorithm 25 on the car random forest emulation example.

the emulator still resembles that of the mean function (in terms of a sharp decrease in total cost followed by a gradual increase), through continued evaluation it is clear that the true total cost function $\ell(n)$ deviates significantly from the initial assessment of total cost function (i.e. $m(n)$). This highlights the flexibility of the

emulation framework. One could argue that fitting a curve, using equation 7.20, through the final version of $\mathbf{d}^1$ would also give a reasonable representation of the true total cost function. However, for true total cost curves that differ in form (for example a double descent curve) from the specified function fit to $\mathbf{d}^1$, the results can be inaccurate.

In summary, by holding out a small percentage (13.6%) of the population we can ensure that key predictors of the model are not lost through intervention for the next model update.

7.4 The Effects of Clustering

Given the content of previous chapters, one may wonder how the presence of a clustering structure (regarding the relationship between the covariates and the response) effects the procedure of locating the minimum total cost. A general detrimental effect of a clustering structure can occur if the data we possess does not contain all clusters present in the population. Note here that whilst it is assumed we have population level data available for knowledge of the value $\aleph$, if this is not the case it is perfectly valid to take the size of the dataset available as $\aleph$. Taking this approach, however, does allow for the possibility of clusters in the population to have no representation in the sample dataset. Therefore, one must be mindful when collating samples of the population to ensure that the resulting dataset is sufficiently diverse.

Assuming that there is a clustering structure present and that all clusters are represented sufficiently in our dataset, the model choice has a large knock-on effect on the behaviour of the expected improvement process. If the model chosen is not designed to account for clustering structure, then the function $\ell(n)$ may behave in non-standard ways. Explaining further, typically we expect $k_2(n)$ to decrease as n increases as we learn more about the behaviour of the system. For a model that does not account for a clustering structure (e.g. standard logistic regression) we do not expect $k_2(n)$ to decrease as n increases as the clustering structure will distort the signal. The result of this is that $k_2(n)$ behaves in a similar fashion to k_1 as the model does not provide much insight to lower costs, violating our assumption

that the model is beneficial. Therefore, if a clustering structure is present, providing a model that cannot detect or manage this structure will fail to be beneficial and therefore render the concerns of intervention effects mute as the model will likely not have an effect.

Conversely the model chosen could be designed to account for clustering structure. Here we would expect for large values of n that $k_2(n)$ will be small, as all clusters should be represented in the hold-out set. For small values of n, however, it remains a possibility that large clusters are not represented sufficiently in the hold-out set and therefore individuals in the intervention set that belong to such unrepresented clusters will receive poor predictions, leading to a higher cost for those individuals. This will have a knock-on effect on the expected cost $k_2(n)$ and consequently the total cost. As a result, typically for data with a clustering structure present the optimal hold-out set will be larger to ensure each cluster is represented.

7.4.1 Clustering Examples

For the entirety of this section we shall keep the covariates $\mathbf{X} \in \mathbb{R}^{\aleph=200\times 2}$ consistent, namely

$$\mathbf{x}_i \sim \begin{cases} \mathcal{N}((-1,-1)^T, \mathcal{I}_2) & \text{if } i \in \{1,\dots,100\} \\ \mathcal{N}((1,1)^T, \mathcal{I}_2) & \text{if } i \in \{101,\dots,200\} \end{cases} \tag{7.21}$$

This low-dimensional simple dataset is chosen for complete clarity on when the data presents clustering structure and when it does not, as will be shown in the following examples.

First we consider the standard setting where there is no clustering structure present in the data. To mimic this setting we simulate the response in the following way:

$$\boldsymbol{\beta} = (0, -0.5, -0.5)^T \tag{7.22}$$

$$y_i \sim \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}})^{-1}) \quad \text{for } i = 1,\dots,200 \tag{7.23}$$

This dataset can be visualised in figure 7.3. As we are in the one-cluster setting,

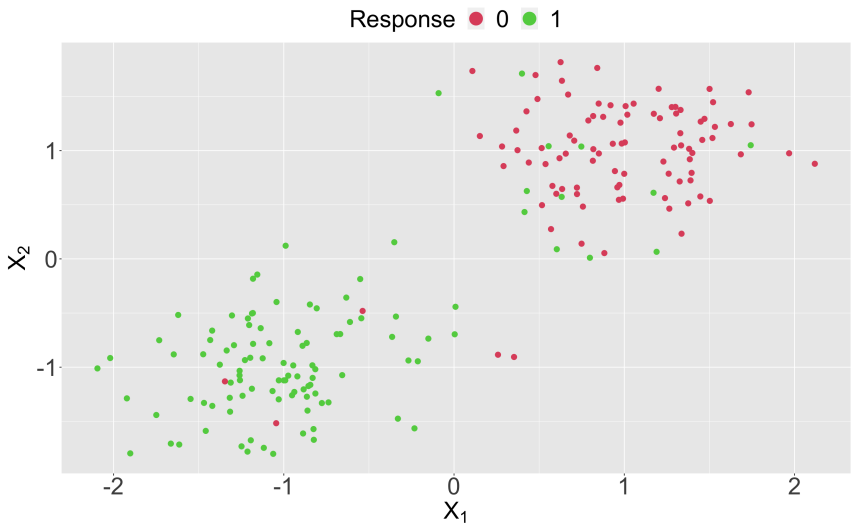


Figure 7.3: One-cluster dataset for the two-Gaussian emulation example. Colours correspond to response type.

a standard logistic regression model is a suitable choice for this data. As seen previously in section 7.3.2, we take a random initial set of sizes to form $\mathbf{n}^1$, we then replicate each of these sizes 10 times to form $\mathbf{n}$. Evaluation at these sizes gives $\mathbf{d}$, $\mathbf{d}^1$ and $\boldsymbol{\sigma}^1$. In order to do this, however, we again must specify k_1 and $k_2(n)$. We define cost to a particular individual in the intervention set who received a particular prediction through table 7.1. Note that here a hard prediction of success is given if the estimated probability of success is greater than 0.5. As before, we assume no model is given to the hold-out set and that the expected cost is $k_1 = 0.5$. The mean function $m(n)$ has the form given in equation (7.20), where a_1, a_2 and a_3 are derived by fitting the curve $k_2(n) = (a_1 n^{-a_2} + a_3)$ to the evaluated expected cost to individuals in the intervention set. This gives $a_1 = 1107.8762717, a_2 = 3.1861977$ and $a_3 = 0.2556594$. For the covariance function hyperparameters[7], we specify $\sigma_u = 10\sqrt{3}$ and $\zeta = 15$. Finally, we specify $\eta_1 = 1$, $\eta_2 = 6$ and $\tau = 10$. The resulting emulator for the initial data is given in figure 7.4 alongside the expected improvement function for this initialisation.

[7]As with the random forest example, these hyperparameters can be specified in this setting through a grid search of plausible values. In absence of a natural objective to optimise, evaluation of these values can be carried out through visual inspection (as in the left sided plot of figure 7.4) of the emulator's suitability to the initial data $\mathbf{d}^1$.

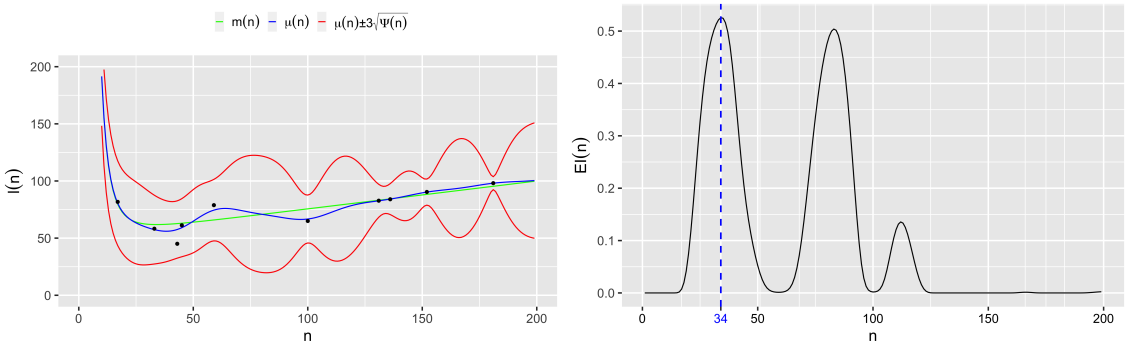


Figure 7.4: Initial emulator (left) of the one-cluster emulation example along with the initial expected improvement function (right). For the left-hand plot black points represent $\mathbf{d}^1$ and for the right-hand plot the blue dashed line highlights the maximum of EI(n).

Even from the initial few points evaluated we can see that the total cost is behaving in a standard way. Small hold-out set sizes do not learn the behaviour of the system and therefore have large total cost, but the total cost decreases as we learn more about the system. Finally, when enough data is collected to fully understand the behaviour of the system, the total cost increases as n increases due to the fact that we are unnecessarily adding individuals to the hold-out set. Also note that although $EI(n) < 1 \ \forall \ n = 1, \ldots, \aleph$ the algorithm will not necessarily immediately terminate as the standard error of our evaluated points may still be large. Indeed, from the left-hand plot in figure 7.4 we can see that although large values of $n \in \mathbf{n}^1$ have little posterior variability[8] (due to the consistency of model output), the same cannot be said for smaller values of $n \in \mathbf{n}^1$, suggesting more evaluations may be required for an accurate approximation of the total cost at these smaller values.

Running algorithm 25 then gives the emulator output seen in figure 7.5, which gives a final optimal hold-out size as $n^\star = 23$. Figure 7.5 clearly shows that the process of discovering the minimum total cost did indeed require evaluating further values of n. This was due to the high standard error associated with suspected optimal hold-out set sizes, and so initially these standard errors were reduced through further evaluations which then in turn altered the emulator allowing for un-evaluated

[8] As the red curves are narrow for black points at large values of $n \in \mathbf{n}^1$ but wide for black points at small values of $n \in \mathbf{n}^1$.

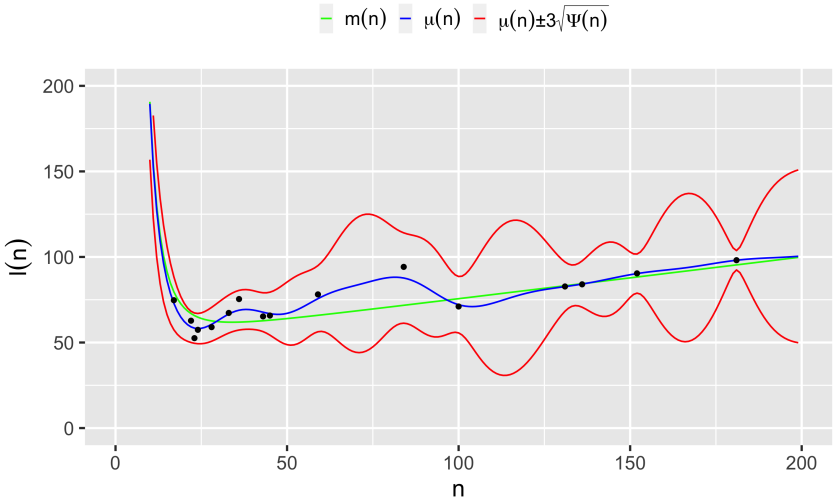


Figure 7.5: Outputted emulator from running algorithm 25 on the one-cluster emulation example. Black points here represent $\mathbf{d}^1$.

sizes to potentially be selected. Additionally, note that the posterior variance is greatly reduced at certain small values of $n \in \mathbf{n}^1$ as the algorithm evaluated these values many times to ensure an accurate approximation of the total cost was made in promising areas. For larger values of $n \in \mathbf{n}^1$, such as $n = 152$, the standard error is still large but we are reasonably confident that further evaluations would not alter d_i^1 (where i is such that $n_i^1 = 152$) enough that it would be lower than the current known minimum.

We now alter the responses such that a two-cluster scenario is present, namely:

$$\boldsymbol{\beta}_1 = (-6, -3, -3)^T \tag{7.24}$$

$$\boldsymbol{\beta}_2 = (6, -3, -3)^T \tag{7.25}$$

$$y_i \sim \begin{cases} \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_1})^{-1}) & \text{if } i \in \{1, \ldots, 100\} \\ \text{Bern}((1 + e^{-(1,\mathbf{x}_i^T)\boldsymbol{\beta}_2})^{-1}) & \text{if } i \in \{101, \ldots, 200\} \end{cases} \tag{7.26}$$

which can be visualised in figure 7.6.

We keep the parameters of our emulator the same as for the one-cluster dataset, with the exception that as we have different initial evaluations, the constants $(a_1, a_2, a_3)^T$ which govern the mean function (as seen in equation (7.20)) will differ, and these are specified as $a_1 = 0.18145298, a_2 = 0.00388301$ and $a_3 = 0.29676975$. The result-

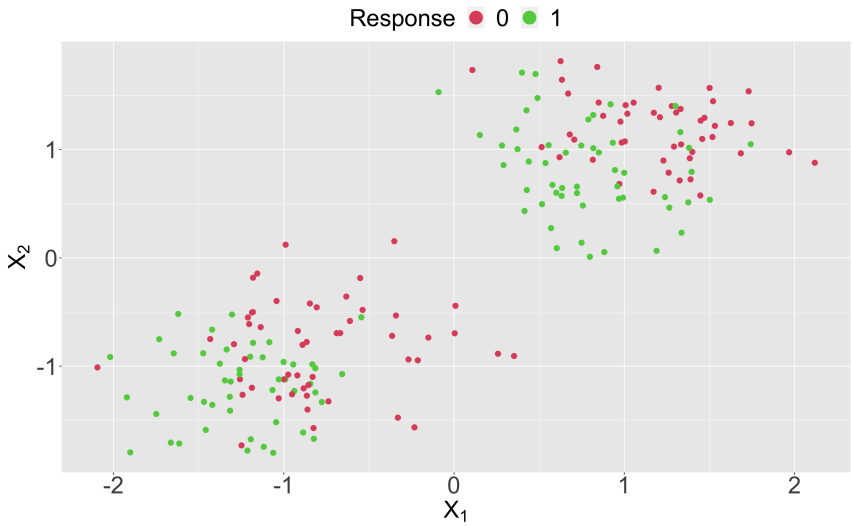


Figure 7.6: Two-cluster dataset for the two-Gaussian emulation example. Colours correspond to response type.

ing initial emulator, alongside the initial expected improvement function, are given in figure 7.7. In this setting the mean function $m(n)$ has interestingly been able

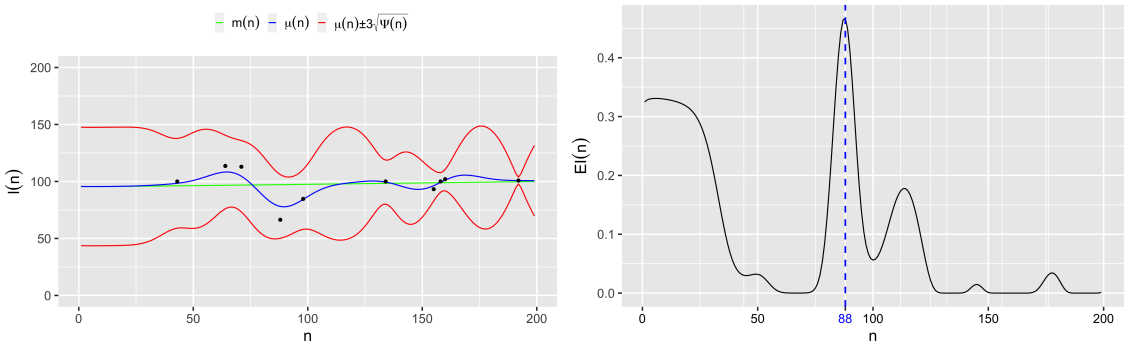


Figure 7.7: Initial emulator (left) of the two-cluster emulation example along with the initial expected improvement function (right). For the left-hand plot black points represent $\mathbf{d}^1$ and for the right-hand plot the blue dashed line highlights the maximum of EI(n).

to capture the constant total cost across n based only on the initial points. Comparatively $\mu(n)$ is distorted more towards the specific placements of these initial points, whose true total cost may not be accurate based on only $\tau = 10$ evaluations. Again, the expected improvement function implies that there is little improvement to be made on the current 'known' minimum, but as algorithm 25 progresses and we increase the certainty in our previous evaluations this may change. The results of running algorithm 25 can be seen is figure 7.8, which gives the optimal hold-out

set size of $n^\star = 98$. The resulting emulator, however, shows with more evaluations the total cost is likely to be fairly constant across n, as was our initial suspicion. This highlights the ineffectiveness of hold-out sets for non-beneficial models, as the cost to an individual in the hold-out set is equivalent to the cost to an individual in the intervention set when the model cannot learn the behaviour of the system.

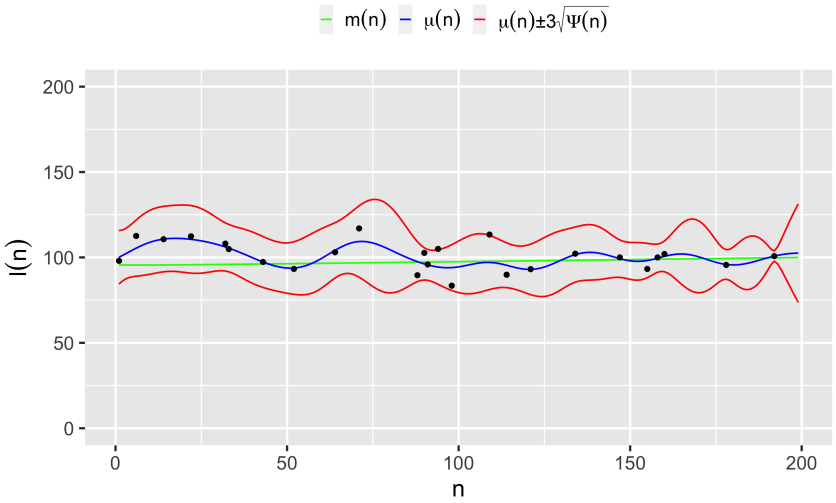


Figure 7.8: Outputted emulator from running algorithm 25 on the two-cluster emulation example. Black points here represent $\mathbf{d}^1$.

In order to compare this to a model which can detect clustering structure, we keep all parameters the same, expect we replace the standard logistic regression model with UNCOVER. Of course, this will give different initial evaluations and so $(a_1, a_2, a_3)^T$ will change again, giving $a_1 = 2.4169461, a_2 = 0.2711138$ and $a_3 = -0.2391386$. Additionally, we require specification for UNCOVER's parameters, and so we select a stopping criterion of $\varkappa = 2$ and a deforestation criterion of a minimum cluster size of $\min\{n, 25\}$ observations[9]. The resulting initial emulator along with the initial expected improvement function are given in figure 7.9.

The use of a model which can handle a clustering structure then allows us to revert to the usual setting where $k_2(n)$ decreases as n increases, as seen in figure 7.9 through the decrease and subsequent increase of both $m(n)$ and $\mu(n)$ as n increases.

[9]This stopping criterion results in the algorithm making only one edge removal before deforestation, which saves a lot of computation time and is appropriate as we know there are only two clusters present.

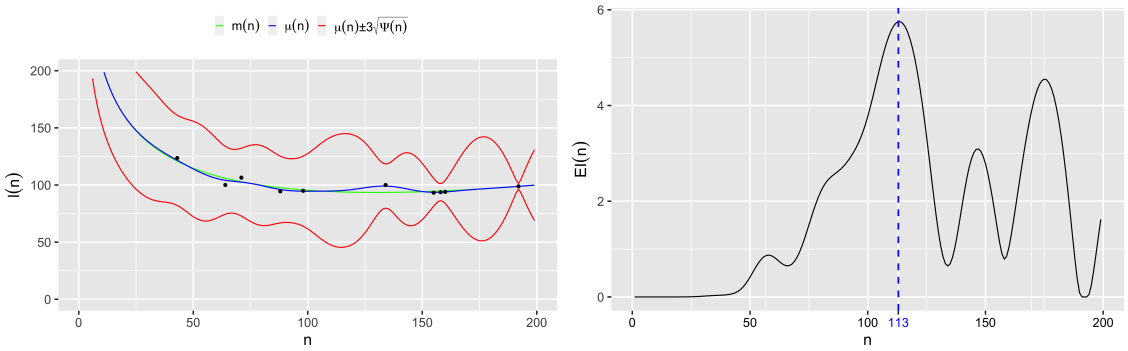


Figure 7.9: Initial emulator (left) of the two-cluster emulation example, using an UNCOVER model, along with the initial expected improvement function (right). Black points represent $\mathbf{d}^1$ and the blue dashed line highlights the maximum of $EI(n)$.

Of course, this is just the initial emulator, and further evaluations may reveal different behaviour. Therefore, we run algorithm 25 to obtain the optimal hold-out set. The resulting emulator is given in figure 7.10, which gave an optimal hold-out size of $n^\star = 71$. Figure 7.10 shows that with multiple evaluations applied to retrieve $n^\star$ the

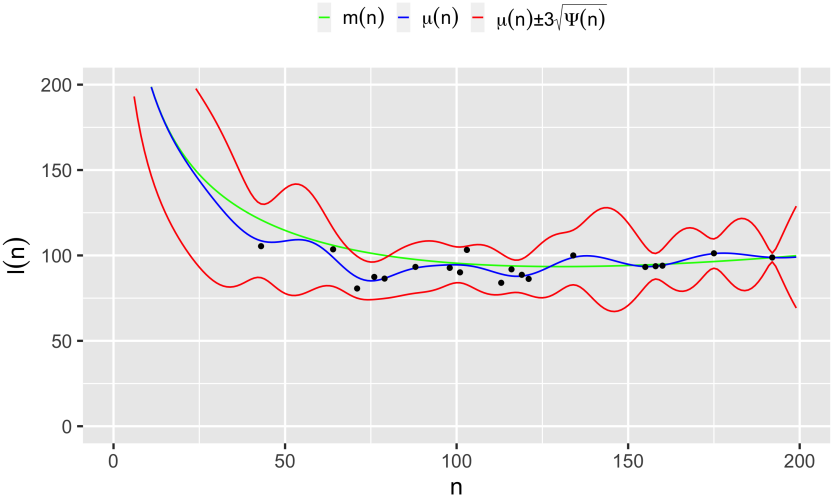


Figure 7.10: Outputted emulator from running algorithm 25 on the two-cluster emulation example where UNCOVER was the model used. Black points here represent $\mathbf{d}^1$.

total cost still behaves in the expected fashion, which we would expect given that the model choice does result in beneficial outputs. The process using UNCOVER is similar to that of the one-cluster dataset using a standard logistic regression model, although due to the presence of two clusters more data is required for the optimal hold-out set in order to guarantee both clusters are sufficiently represented.

7.5 Summary

In general the problem of model updating can be solved through the use of a hold-out set, the size of which can be optimised through specification of the various costs one encounters through implementation of a model. Locating this optimum can also be achieved with Bayesian optimisation through emulation of the total cost function, which will be between 1 and $\aleph - 1$ provided certain assumptions are met. Key among these assumptions is that the model is beneficial and therefore members of the intervention set benefit from an accurate model. When there is clustering present in the population, however, this assumption can be violated by a model which cannot handle such a clustering structure.

UNCOVER clearly is a model that can handle clustering structure and so meets the assumptions required for the existence of an optimal hold-out set. Interestingly, setting certain parameters of UNCOVER can showcase the power of emulation for discovering the minimum. For example, consider the UNCOVER example in section 7.4.1, where we can see from figure 7.10 that the emulator behaves in a similar fashion to the mean function $m(n)$ which acts as the emulator prior. However, the total cost function does not behave like $m(n)$ for $n < 50$. This is due to the deforestation criterion being the minimum size of a cluster must be greater or equal to $\min\{n, 25\}$, when n is the training data size (i.e. the hold-out set size). Therefore, in order to obtain a two-cluster output we require $n = 50$. As a result, assuming that a one cluster model learns very little about the behaviour of the system, $k_2(n) \approx k_1$ for $n < 50$ and so the total cost should be roughly constant for this range of n. Consequently, the total cost for $n < 50$ should approximately be 100, a large deviation away from $m(n)$. This can be shown by adding the point $n = 25$ (initially with 10 evaluations, then continued evaluation until the standard error is below 6) to the emulator. This can be seen in figure 7.11.

Clearly the point $n = 25$ deviates significantly from our prior mean function $m(n)$, but our posterior mean function $\mu(n)$ can account for this. This is in a setting where we have a solid understanding of the total cost function, but for other models where the behaviour of the total cost is unknown, this flexibility is crucial

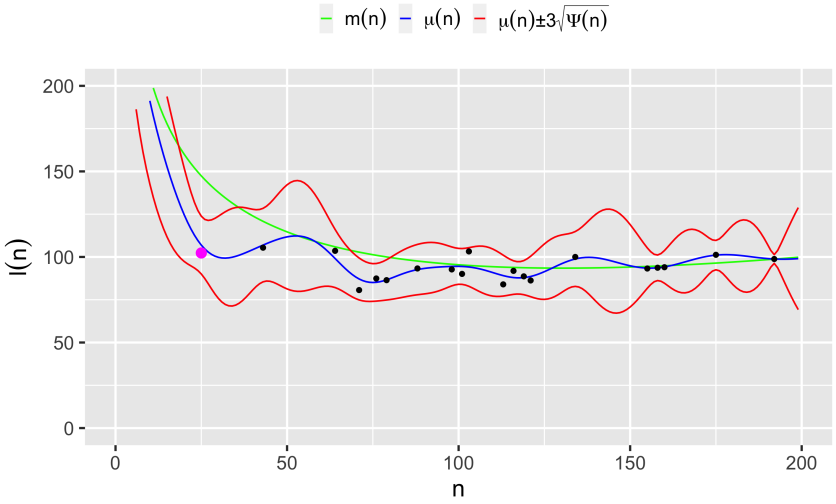


Figure 7.11: Outputted emulator from running algorithm 25 on the two-cluster emulation example (where UNCOVER was the model used) and then adding the point $n = 25$ (pink) through evaluation 113 times. Black points here represent $\mathbf{d}^1$.

in determining the optimal hold-out set and not relying too heavily on prior beliefs.

As a final point on the use of UNCOVER in safe model updating, if a population is suspected to contain a clustering structure, then an initial UNCOVER model using the entire population could be beneficial in reducing the size of the hold-out set. In more detail, if an initial run using UNCOVER produced a $K > 1$ cluster model, for subsequent hold-out sets a stratified sampling approach can be taken to form the set such that each cluster is represented in the training data for the model. The hard clustering output from UNCOVER makes this possible and would result in less variability in the evaluations at lower set sizes as well as the model potentially learning the behaviour of the system at much smaller sizes[10]. If UNCOVER initially detected no clustering structure then this gives confidence in using the previous method of random sampling to form hold-out sets. Whilst this approach has the potential to greatly reduce the hold-set size, there may be ethical issues to consider — for example a member of a small cluster will have a much greater chance of being in the hold-out set than a member of a large cluster.

[10] Assuming a clustering structure is present and UNCOVER is the model in question.

CHAPTER 8

Conclusion

The use of clustering in predictive modelling requires careful consideration of stakeholder's priorities. Typically focus has been heavily weighted either towards interpretability of the resulting cohorts (at the expense of model accuracy) or towards predictive power of the model (at the expense of solid cohort descriptions). UNCOVER aims to provide a method of predictive modelling which gives equal weight to the detection of clear cohorts, such that separate action plans can be developed for cohort members based on their cohort's particular relationship with the response.

With respect to cohort detection, UNCOVER insists potential cohorts adhere to the structure of the covariates, realised through a Minimum Spanning Tree (MST), but this structure can be formed through a subset of the covariates. This gives the user the ability to select the subset of covariates most beneficial in uncovering a clustering structure. The use of MSTs also allows for few assumptions to be made on the shape or properties of the clusters (that may not be known a priori).

For model accuracy, the inclusion of a response is incorporated from the beginning of the algorithm, with a Bayesian product of logistic regression models being selected as the overall model framework. This allows a comparison of models with different partitions of the data through the Bayesian evidence — an appropriate

metric given the small amounts of data one might encounter for a particular cluster. This also gives a key property to UNCOVER if one has doubts about the presence of any clustering at all, in that UNCOVER allows the inclusion of a one cluster model as an acceptable output. Indeed, an advantage of the UNCOVER method is that the number of clusters is not required to be known prior to running the algorithm. Additionally, we also utilise the Bayesian evidence to produce a methodical approach to meeting pre-specified 'deforestation' criteria similar to that of pruning.

As utilisation of this method is clearly encouraged in general settings, the implementation of UNCOVER is made available through the `R` package `UNCOVER`. This package has been specifically designed to allow for ease of use, with default settings put in place for accessibility to novices but also deep specifications available for users more familiar with the UNCOVER methodology. The specifications may range from theoretical parameters, such as a threshold for use of the Bayesian information criterion and deforestation criteria, to more computational aspects such as the threshold for cache checking within a tailored memoisation framework.

Finally, application of UNCOVER in real-world settings may result in the replacement of a previously implemented model, therefore necessitating discussion regarding the problems of model updating. In general, naïve model updating may result in better models performing worse due to interventions effects, with a possible solution to this being the notion of a hold-out set. A hold-out set requires an optimal size such that the cost to individuals in the population in minimised, and this can be tackled through the minimisation of a total cost function using a surrogate model and the method of expected improvement. Referring back to the clustering context, UNCOVER can be of significant use here in determining if there is any clustering present in the population as well as giving clarity to the cohorts that need to be represented in the hold-out set for accurate modelling.

8.1 Future Work

The research conducted in this thesis has opened many interesting avenues of further research that can be applied in the fields of theory, computation and application.

8.1.1 Seeing the Wood Through the Trees

In terms of theory, an interesting question surrounds the selection of the subset of covariates $\mathfrak{P}$. The intention of specifying $\mathfrak{P}$ was to allow for either the stakeholder or statistician to reduce the dimensionality of the covariate space, which could be done to focus on key demographics where the detection of cohorts would be beneficial to the stakeholder, or to discover a covariate structure in which a supervised clustering structure is most apparent. This requires a certain level of knowledge by the user, however. Therefore, a more automatic selection of $\mathfrak{P}$ could be desirable. A standard solution would be to run UNCOVER multiple times for different selections of the covariates and select the $\mathfrak{P}$ which gives the largest model Bayesian evidence. This solution represents an attempt at optimising $\mathfrak{P}$. Another view one could take is not which $\mathfrak{P}$ gives the clustering structure with the best model (according to the Bayesian evidence), but which clustering structure gives a consistently good model[1] regardless of $\mathfrak{P}$. For small training data this viewpoint will avoid a potentially misleading selection of $\mathfrak{P}$ which gives a model that generalises poorly. Additionally, a solution for this viewpoint can be incorporated into a single run of UNCOVER, by changing $\mathfrak{P}$ after every edge removal. Changing $\mathfrak{P}$ will then update the Minimum Spanning Forest (MSF) with K components back to a Minimum Spanning Tree (MST) with one component. To get back to a K-component MSF, for each edge $\{i, j\}$ in the set of the $K-1$ previous graph removed edges, we remove the longest edge in the path from i to j in the new graph. We would then update the cluster index sets $\mathfrak{V}_1, \ldots, \mathfrak{V}_K$ accordingly and then consider edge reintroduction based on the new $\mathfrak{P}$-determined graph. The theoretical properties (for example, is the algorithm guaranteed to reach a natural stopping point with a changing structure?) alongside implementation factors would be of great interest to explore.

[1] A 'good' model here refers to a model that makes accurate predictions of new and existing data.

8.1.2 Beyond Logistic Regression

The entirety of this thesis has focused on the special case of logistic regression models as a basis for UNCOVER. However, the framework developed for UNCOVER is not specific to logistic regression, and in many cases is easily replaceable with another base model. The simplest example of this would be probit regression, another model for a binary response. If the response was continuous then the standard regression model would represent a direct replacement for logistic regression.

The aspect that each of these base models have in common is that they are all parametric models, and therefore posteriors along with Bayesian evidences can all be obtained from these models. An interesting avenue to explore, however, is how non-parametric models can be incorporated within UNCOVER. One could change the metric from the Bayesian evidence to a metric that does not require probabilistic modelling (such as the AUC) or a more challenging prospect would be to attempt to derive parametric counter-parts to the non-parametric model such that a posterior and Bayesian evidence could be derived. An example of this would be a classic decision tree, where the unknown parameters in which to construct a posterior are the split parameters, but crucially here the observations would not be i.i.d. and so this presents further challenges. Tackling this problem would then allow for promising connections to be made between UNCOVER and the area of Bayesian inference which replaces the likelihood with a loss function [110].

8.1.3 Batched Spanning Trees

Whilst several techniques have been implemented (as discussed in chapter 5) in order to improve the computation time of UNCOVER, a significant procedure which has not yet been discussed is the construction of the initial MST. For large data this will take considerable time, and this problem is further exacerbated when considering the techniques discussed in section 8.1.1. Several algorithms aside from Prim's algorithm have been devised to tackle the computational burden of MST construction [63–65], of which it is future work to implement within UNCOVER. For extremely large datasets, however, the computational burden will still remain as all such algorithms

depend on the number of vertices n.

Interesting considerations to make here are the advantages and disadvantages of parallelisation of this process. If we were to spilt the vertices into batches, we could compute the MSTs for each of these batches in parallel. Then, to obtain the final graph we would construct the overall spanning tree by only considering edges that are in the individual MSTs and edges between batches. The result would be a spanning tree but this is not guaranteed to be an MST, therefore the gains we make in computation time may be potentially lost in not obtaining an accurate representation of the covariate structure. A parallelisable method such as this which gives spanning trees with a structure close to that of an MST would be a key area to explore to ensure UNCOVER has wide use in large data settings.

8.1.4 Cluster Caches

When considering the eviction policy for the caches involved in UNCOVER, we use the standard least recently used policy alongside a careful edge ordering to produce an efficient use of memoisation. In addition to this, we also have made use of specialised save states, which are deemed too important to risk eviction from the cache. An alternative option is to combine these two concepts into cluster specific caches. In this setting, we produce a cache per cluster and a cache per edge removed. Initially starting with one cluster cache, labelled cache 'Cluster – 1', we would discover the optimal edge ϵ to remove in the standard way. Then, before splitting the cluster, we would re-categorise the current cluster cache as an edge cache, now labelled 'Edge – ϵ', and then split the cluster. The two new clusters would then each receive a new cluster cache labelled cache 'Cluster – 1' and cache 'Cluster – 2'. If an edge ϵ was reintroduced, we would first remove the two cluster caches of the clusters to be combined and then re-categorise the ϵ edge cache to be the cluster cache of the newly formed combined cluster.

As hard clustering dictates observations are confined to their own cluster there would be no loss of information by having specific cluster caches. However, given that we may have a restriction on the size of a cache and the number of clusters at any one point of UNCOVER is unknown, determining the size of multiple caches

presents a challenge. We of course will know the maximum number of clusters through specification of the stopping criterion, so we can state that there will be at most $\varkappa + \varkappa - 1$ caches at any time. Simply dividing the allotted total cache size by $\varkappa + \varkappa - 1$ for the individual cache sizes, however, is inefficient as we only reach this number of clusters at most once in the algorithm. An adaptive cache sizing policy is an interesting solution to this problem, but careful consideration must be made on how the caches are modified when they are 'downsized' by the presence of a new cluster or 'upsized' by the combination of clusters.

8.1.5 Influential Observations

A paper by Broderick et.al [111] raises an important issue of a phenomenon that occurs when a small group of observations within a dataset are highly influential on the inferences one makes when utilising the data. In circumstances where this is not due to finite data or outliers, removal of such observations drastically changes the conclusions drawn, which could have a detrimental impact if actions are taken on the basis of these conclusions. This has an extremely large impact on the problem of optimal hold-out sets seen in chapter 7. If we assume that inclusion of this small group of observations gives a much more accurate model, then the hold-out set is much more sensitive to its contents as opposed to simply its size. Ignoring the presence of a small influential group will lead to a total cost function which is highly variable and favours much larger sizes of the optimal hold-out set to increase the possibility of capturing this small group.

If we could identify the individuals in the population which are influential prior to determining the hold-out set, we can make adjustments to ensure this group is represented, therefore lowering the variability and allowing for a smaller hold-out set size. Broderick et.al suggest such a method in their paper, but an alternative would be to use an adaptation of UNCOVER. Instead of taking the product of Bayesian evidences as our metric we take the maximum Bayesian evidence per observation[2]. Assuming the large majority of observations act as noise for the influential group,

[2]Such that for cluster k with observation index set $\mathfrak{V}_k$ and Bayesian evidence Z_k, the Bayesian evidence per observation would be $Z_k \backslash |\mathfrak{V}_k|$.

this version of UNCOVER will highlight the influential group or groups by considering them as an optimal cluster. Implementation as well as theoretical considerations would need to be made for this version of UNCOVER, as well as the ethical aspects of application (e.g. if observations corresponded to patients is it ethical to allow some patients to never receive an accurate prediction by always being in the hold-out set?).

Bibliography

[1] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, "A roadmap of clustering algorithms: finding a match for a biomedical application," *Briefings in Bioinformatics*, vol. 10, no. 3, pp. 297–314, 2009. 1.1

[2] O. V. Grishchenko and M. Rossi, "The role of heterogeneity in asset pricing: The effect of a clustering approach," *Journal of Business & Economic Statistics*, vol. 30, no. 2, pp. 297–311, 2012. 1.1

[3] J. Smith and P. B. Baltes, "Profiles of psychological functioning in the old and oldest old.," *Psychology and Aging*, vol. 12, no. 3, p. 458, 1997. 1.1

[4] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*, pp. 281–297, 1967. 1.1, 2.1.1

[5] G. J. McLachlan and D. Peel, *Finite Mixture Models*. John Wiley & Sons, 2004. 1.1

[6] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991. 1.1, 2.2.2

[7] J. Liley, G. Bohner, S. R. Emerson, B. A. Mateen, K. Borland, D. Carr, S. Heald, S. D. Oduro, J. Ireland, K. Moffat, *et al.*, "Development and assessment of a machine learning tool for predicting emergency admission in scotland," *medRxiv*, pp. 2021–08, 2021. 1.2

[8] NHS NSS Health and Social Care Information Programme, "Scottish Patients at Risk of Readmission and Admission (SPARRA): A report on the development of SPARRA version 3." `https://www.isdscotland.org/Health-Topics/Health-and-Social-Community-Care/SPARRA/2012-02-09-SPARRA-Version-3.pdf`, 2011. Accessed: 2023-03-16. 1.2

[9] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, 2002. 1b, 3.1.2, 3.1.2.1, 4.2, 5.1, 2

[10] D. Michie, ""memo" functions and machine learning," *Nature*, vol. 218, no. 5136, pp. 19–22, 1968. 2

[11] S. Emerson, *UNCOVER: Utilising Normalisation Constant Optimisation via Edge Removal (UNCOVER)*, 2023. R package version 1.1.0. 4, 5.5, 1

[12] S. Haidar-Wehbe, S. R. Emerson, L. J. Aslett, and J. Liley, "Optimal sizing of a holdout set for safe predictive model updating," *arXiv preprint arXiv:2202.06374*, 2022. 5, 7, 7.1, 7.2.4, 7.3.1

[13] J. Liley, S. Emerson, B. Mateen, C. Vallejos, L. Aslett, and S. Vollmer, "Model updating after interventions paradoxically introduces bias," in *International Conference on Artificial Intelligence and Statistics*, pp. 3916–3924, PMLR, 2021. 5, 7.1, 7.1

[14] D. Harada, H. Asanoi, T. Noto, and J. Takagawa, "Different pathophysiology and outcomes of heart failure with preserved ejection fraction stratified by K-means clustering," *Frontiers in Cardiovascular Medicine*, vol. 7, p. 607760, 2020. 2.1.1

[15] R. Soni and K. James Mathai, "An innovative 'cluster-then-predict' approach for improved sentiment prediction," in *Advanced Computing and Communication Technologies*, pp. 131–140, Springer, 2016. 2.1.1, 2.1.3

[16] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987. 2.1.1

[17] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001. 2.1.1, 6.1

[18] L. L. McQuitty, "Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies," *Educational and Psychological Measurement*, vol. 17, no. 2, pp. 207–229, 1957. 2.1.2

[19] T. J. Sørensen, *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons*. I kommission hos E. Munksgaard, 1948. 2.1.2

[20] R. R. Sokal, "A statistical method for evaluating systematic relationships.," *Univ. Kansas, Sci. Bull.*, vol. 38, pp. 1409–1438, 1958. 2.1.2

[21] K. Perrakis, T. Lartigue, F. Dondelinger, and S. Mukherjee, "Latent group structure and regularized regression," 2020. 2.1.3

[22] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894. 2.2.1

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977. 2.2.1

[24] G. J. McLachlan and D. Peel, *Finite mixture models*. Wiley Series in Probability and Statistics, 2000. 2.2.1

[25] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 46, no. 2, pp. 149–170, 1984. 2.2.1, 3.2.1

[26] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994. 2.2.2, 2.2.2.1

[27] Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph, "A mixture of feature experts approach for protein-protein interaction prediction," in *BMC Bioinformatics*, vol. 8, pp. 1–14, BioMed Central, 2007. 2.2.2

[28] M. Enzweiler and D. M. Gavrila, "A multilevel mixture-of-experts framework for pedestrian classification," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2967–2979, 2011. 2.2.2

[29] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012. 2.2.2.1

[30] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009. 2.2.2.1, 3.1, 4.1.1, 4.3.3, 4.4, 4.4.3, 6.2, 7.3.2

[31] Z. Liu and M. Barahona, "Graph-based data clustering via multiscale community detection," *Applied Network Science*, vol. 5, pp. 1–20, 2020. 3

[32] F. V. Jensen and T. D. Nielsen, *Bayesian networks and decision graphs*, vol. 2. Springer, 2007. 3

[33] P. M. Lukacs, K. P. Burnham, and D. R. Anderson, "Model selection bias and freedman's paradox," *Annals of the Institute of Statistical Mathematics*, vol. 62, pp. 117–125, 2010. 3.1

[34] T. Kloek and H. K. Van Dijk, "Bayesian estimates of equation system parameters: an application of integration by Monte Carlo," *Econometrica: Journal of the Econometric Society*, pp. 1–19, 1978. 3.1.1

[35] C. P. Robert and G. Casella, *Monte Carlo statistical methods*, vol. 2. Springer, 1999. 3.1.1

[36] C. Dai, J. Heng, P. E. Jacob, and N. Whiteley, "An invitation to sequential Monte Carlo samplers," *Journal of the American Statistical Association*, vol. 117, no. 539, pp. 1587–1600, 2022. 3.1.2, 4.2

[37] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov chain Monte Carlo*. CRC press, 2011. 3.1.2, 5

[38] P. Del Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, 2006. 3.1.2, 3.1.2.1

[39] M. Gerber, N. Chopin, and N. Whiteley, "Negative association, ordering and convergence of resampling methods," *The Annals of Statistics*, vol. 47, no. 4, pp. 2236–2260, 2019. 3.1.2

[40] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," 1970. 3.1.2.1

[41] B. J. Kleijn and A. W. van der Vaart, "The Bernstein-von-Mises theorem under misspecification," *Electronic Journal of Statistics*, vol. 6, pp. 354–381, 2012. 3.1.2.1, 5.2, B.1

[42] D. Freedman, "Wald lecture: On the Bernstein-von Mises theorem with infinite-dimensional parameters," *The Annals of Statistics*, vol. 27, no. 4, pp. 1119–1141, 1999. 3.1.2.1

[43] I. Castillo and R. Nickl, "Nonparametric Bernstein–von Mises theorems in Gaussian white noise," 2013. 3.1.2.1

[44] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, 1994. 3.1.2.1

[45] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory*, pp. 267–281, Akadémiai Kiadó Location Budapest, Hungary, 1973. 3.2.1

[46] J. E. Cavanaugh and A. A. Neath, "The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 11, no. 3, p. e1460, 2019. 3.2.1

[47] C. M. Hurvich and C.-L. Tsai, "Regression and time series model selection in small samples," *Biometrika*, vol. 76, no. 2, pp. 297–307, 1989. 3.2.1

[48] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, pp. 461–464, 1978. 3.2.1

[49] S. Konishi and G. Kitagawa, "Information criteria and statistical modeling," 2008. 3.2.1

[50] F. Llorente, L. Martino, D. Delgado, and J. Lopez-Santiago, "Marginal likelihood computation for model selection and hypothesis testing: an extensive review," *SIAM Review*, vol. 65, no. 1, pp. 3–58, 2023. 3.2.2, 4.6.1

[51] H. Jeffreys, "Some tests of significance, treated by the theory of probability," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 31, pp. 203–222, Cambridge University Press, 1935. 3.2.2, 6.4

[52] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the american statistical association*, vol. 90, no. 430, pp. 773–795, 1995. 3.2.2

[53] H. Jeffreys, *The theory of probability*. OuP Oxford, 1998. 3.2.2, 4.4.2, 6.4, B.1

[54] N. Friel and J. Wyse, "Estimating the evidence – a review," *Statistica Neerlandica*, vol. 66, no. 3, pp. 288–308, 2012. 3.2.2

[55] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995. 3.2.3, A.4

[56] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000. 3.2.3, A.4, A.4

[57] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901. 3.3

[58] A. Unwin and K. Kleinman, "The iris data set: In search of the source of virginica," *Significance*, vol. 18, 2021. 3.3

[59] D. B. West *et al.*, *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001. 3.3.1

[60] B. M. Moret and H. D. Shapiro, "An empirical analysis of algorithms for constructing a minimum spanning tree," in *Algorithms and Data Structures: 2nd Workshop, WADS'91 Ottawa, Canada, August 14–16, 1991 Proceedings 2*, pp. 400–411, Springer, 1991. 3.3.2

[61] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957. 3.3.2, 4.1.1

[62] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956. 3.3.2

[63] D. R. Karger, P. N. Klein, and R. E. Tarjan, "A randomized linear-time algorithm to find minimum spanning trees," *Journal of the ACM (JACM)*, vol. 42, no. 2, pp. 321–328, 1995. 3.3.2, 8.1.3

[64] B. Chazelle, "A minimum spanning tree algorithm with inverse-Ackermann type complexity," *Journal of the ACM (JACM)*, vol. 47, no. 6, pp. 1028–1047, 2000. 3.3.2, 8.1.3

[65] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," *Journal of the ACM (JACM)*, vol. 49, no. 1, pp. 16–34, 2002. 3.3.2, 8.1.3

[66] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022. 3.3.2

[67] S. Varma and R. Simon, "Iterative class discovery and feature selection using minimal spanning trees," *BMC Bioinformatics*, vol. 5, no. 1, pp. 1–9, 2004. 3.3.2

[68] C. Zhong, D. Miao, and P. Fränti, "Minimum spanning tree based split-and-merge: A hierarchical clustering method," *Information Sciences*, vol. 181, no. 16, pp. 3397–3410, 2011. 3.3.2

[69] A. Vathy-Fogarassy, B. Feil, and J. Abonyi, "Minimal spanning tree based fuzzy clustering," *Proc World Acad Sci Eng Tech*, vol. 8, pp. 7–12, 2005. 3.3.2

[70] J. C. Gower and G. J. Ross, "Minimum spanning trees and single linkage cluster analysis," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 18, no. 1, pp. 54–64, 1969. 3.3.2

[71] O. Grygorash, Y. Zhou, and Z. Jorgensen, "Minimum spanning tree based clustering algorithms," in *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pp. 73–81, IEEE, 2006. 3.3.2

[72] Z. T. Luo, H. Sang, and B. Mallick, "A Bayesian contiguous partitioning method for learning clustered latent variables," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1748–1799, 2021. 3.3.2

[73] J. Riordan, *Introduction to combinatorial analysis*. Courier Corporation, 2012. 4.1.1

[74] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, pp. 857–871, 1971. 4.1.1

[75] K. B. Athreya and S. N. Lahiri, *Measure theory and probability theory*, vol. 19. Springer, 2006. 4.1.2

[76] R. A. Jacobs, F. Peng, and M. A. Tanner, "A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures," *Neural Networks*, vol. 10, no. 2, pp. 231–241, 1997. 4.3.3, 4.4

[77] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 493–507, 2012. 4.4.3

[78] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006. 4.4.3

[79] S. Lotfi, P. Izmailov, G. Benton, M. Goldblum, and A. G. Wilson, "Bayesian model selection, the marginal likelihood, and generalization," in *International Conference on Machine Learning*, pp. 14223–14247, PMLR, 2022. 4.4.3

[80] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013. 4.6

[81] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, pp. 107–145, 2001. 4.6

[82] J. Reineke, D. Grund, C. Berg, and R. Wilhelm, "Timing predictability of cache replacement policies," *Real-Time Systems*, vol. 37, pp. 99–122, 2007. 5.1

[83] G. Valiente, *Algorithms on trees and graphs*, vol. 112. Springer, 2002. 5.1.2, 5.1.2

[84] D. Dua and C. Graff, "UCI machine learning repository," 2017. 5.2, 6, 7.3.2

[85] K. Jagadish — Kaggle, "Mall customers." `https://www.kaggle.com/datasets/kandij/mall-customers`, 2019. Accessed: 2023-04-28. 5.2

[86] R. E. Kass, L. Tierney, and J. B. Kadane, "Laplace's method in bayesian analysis," *Contemporary Mathematics*, vol. 115, pp. 89–99, 1991. 7

[87] Statisticat and LLC., *LaplacesDemon Tutorial*, 2021. R package version 16.1.6. 7

[88] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022. 7, 5.4, 4

[89] M. Fasiolo, *An introduction to mvnfast. R package version 0.2.8*. University of Bristol, 2014. 1

[90] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006. 2

[91] H. Wickham, J. Hester, W. Chang, K. Müller, and D. Cook, *memoise: 'Memoisation' of Functions*, 2021. R package version 2.0.1. 3

[92] W. Chang, *cachem: Cache R Objects with Automatic Pruning*, 2021. R package version 1.0.6. 5

[93] G. Csárdi, *crayon: Colored Terminal Output*, 2022. R package version 1.5.1. 6

[94] B. Schloerke, D. Cook, J. Larmarange, F. Briatte, M. Marbach, E. Thoen, A. Elberg, and J. Crowley, *GGally: Extension to 'ggplot2'*, 2021. R package version 2.1.2. 6

[95] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. 6

[96] A. Kassambara, *ggpubr: 'ggplot2' Based Publication Ready Plots*, 2022. R package version 0.5.0. 6

[97] H. Wickham and D. Seidel, *scales: Scale Functions for Visualization*, 2022. R package version 1.2.1. 6

[98] E. Campitelli, *ggnewscale: Multiple Fill and Colour Scales in 'ggplot2'*, 2022. R package version 0.4.8. 6

[99] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009. 6

[100] W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn, and W. B. Ford, "The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait," *Sea Fisheries Division, Technical Report*, vol. 48, p. p411, 1994. 6

[101] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *The American Journal of Cardiology*, vol. 64, no. 5, pp. 304–310, 1989. 6

[102] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001. 7

[103] R. Akolekar, A. Syngelaki, L. Poon, D. Wright, and K. H. Nicolaides, "Competing risks model in early screening for preeclampsia by biophysical and biochemical markers," *Fetal Diagnosis and Therapy*, vol. 33, no. 1, pp. 8–15, 2013. 7.1

[104] T. Viering and M. Loog, "The shape of learning curves: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 7.3, 7.3

[105] S. Singh, "Modeling performance of different classification methods: deviation from the power law," *Project Report, Department of Computer Science, Vanderbilt University, USA*, 2005. 7.3

[106] I. Vernon, J. Liu, M. Goldstein, J. Rowe, J. Topping, and K. Lindsey, "Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions," *BMC Systems Biology*, vol. 12, no. 1, pp. 1–29, 2018. 7.3, 7.3, 7.3

[107] R. G. Bower, M. Goldstein, and I. Vernon, "Galaxy formation: a bayesian uncertainty analysis," 2010. 7.3

[108] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010. 7.3.1, 7.3.1

[109] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh, "Regret for expected improvement over the best-observed value and stopping condition," in *Asian Conference on Machine Learning*, pp. 279–294, PMLR, 2017. 7.3.1

[110] P. G. Bissiri, C. C. Holmes, and S. G. Walker, "A general framework for updating belief distributions," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 78, no. 5, pp. 1103–1130, 2016. 8.1.2

[111] T. Broderick, R. Giordano, and R. Meager, "An automatic finite-sample robustness metric: When can dropping a little data make a big difference?," *arXiv preprint arXiv:2011.14999*, 2020. 8.1.5

[112] G. Voronoï, "New applications of continuous parameters to the theory of quadratic forms," *Z. Reine Angew. Math*, vol. 134, p. 198, 1908. A.1

[113] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. A.2

[114] E. Anderson, "The irises of the Gaspe Peninsula," *Bulletin American Iris Society*, vol. 39, pp. 2–15, 1935. A.2

[115] C. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," *Advances in Neural Information Processing Systems*, vol. 14, 2001. A.4

[116] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, pp. 209–230, 1973. A.4

[117] A. E. Gelfand and A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990. A.4

[118] A. Beskos, D. Crisan, and A. Jasra, "On the stability of sequential monte carlo methods in high dimensions," 2014. B.1

APPENDIX A

Further Information on Previous Clustering Methods

A.1 The Effect of K-means Clustering in Covariate Space

When convergence of the K-means algorithm (algorithm 26) has been reached to give index sets $\mathfrak{V}_1, \ldots, \mathfrak{V}_K$ (with associated centroids $\mathbf{c}_1, \ldots, \mathbf{c}_K$), the centroids induce a Voronoi diagram [112] which consists of Voronoi regions $\mathcal{R}_1, \ldots, \mathcal{R}_K$ of the covariate space, defined by the Euclidean distance metric and $\mathbf{c}_1, \ldots, \mathbf{c}_K$.

Definition A.1.1 (Voronoi Region)**.** *Given a distance metric $d(\cdot, \cdot)$ and centroids $\mathbf{c}_1, \ldots, \mathbf{c}_K$, a Voronoi region of the space $\mathcal{R}$ is given by*

$$\mathcal{R}_k = \{\mathbf{x} \in \mathcal{R} : d(\mathbf{x}, \mathbf{c}_k) \leq d(\mathbf{x}, \mathbf{c}_l) \ \forall \ l \neq k\}$$

It is important to note here that the covariate space is not partitioned by $\mathcal{R}_1, \ldots, \mathcal{R}_K$, as $\mathcal{R}_k \cap \mathcal{R}_l \neq \emptyset$ for neighbouring regions. However, regions defined in a similar way except with a strict inequality on the distance measures will form a partition when considered with the decision boundaries. Therefore, a Voronoi diagram gives a strong indication into the topology of the possible regions of the

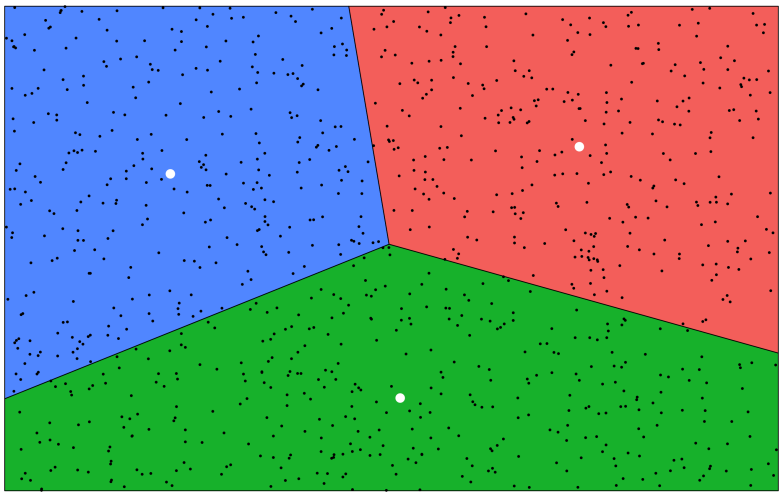


Figure A.1: Voronoi diagram from centroids (white) produced by K-means. Black points indicate observations.

covariate space formed by the presence of new data, as once the centroids have reached stability with the training data, new data can be assigned a cluster based on Euclidean distance to the nearest centroid.

A.2 Visualisation of the Hierarchical Clustering Algorithm

The choice of K is still a quantity that requires specification before algorithm 27 can be run. However, by selecting $K = 1$ for agglomerative clustering (or $K = n$ for divisive clustering) and saving the output at each iteration we can obtain the outputted clusters for every value of K. Plotting this as a dendrogram then allows for a clear visual inspection of which value of K is most suitable. An example of this for a sample of the iris dataset [113,114] (using the Euclidean distance metric and the complete linkage method) is shown in figure A.2. We can also utilise dendrograms to identify the differences in clustering output that occur for different linkage methods, as seen in figure A.3.

This additional level of visualisation (as well as more flexibility in the shape of clusters produced by the algorithm) gives in some aspects hierarchical clustering a advantage over K-means for both interpretability and potential predictive power.

Algorithm 26: Euclidean distance K-means

Input : *Number of Clusters* — K, *Covariate Matrix* — $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, *Centroid Matrix* — $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_K)^T$

Initialisation : Let $\tilde{\mathfrak{V}}_1 = \emptyset, \ldots, \tilde{\mathfrak{V}}_K = \emptyset$

Step 1 : Let $\mathfrak{V}_k = \emptyset$ for $k = 1, \ldots, K$.

Step 2 : **for** $i = 1, \ldots, n$ **do**

  for $k = 1, \ldots, K$ **do**

    Calculate $d_k = \|\mathbf{x}_i - \mathbf{c}_k\|_2 = \sqrt{\sum_{j=1}^{p}(x_{ij} - c_{kj})^2}$

  end

  Assign observation i to cluster $k^\star = \arg\min_{k \in \{1,\ldots,K\}} \{d_k\}$ by adding i to $\mathfrak{V}_{k^\star}$

end

Step 3 : **for** $k = 1, \ldots, K$ **do**

  Let

$$\mathbf{c}_k = \frac{1}{|\mathfrak{V}_k|} \sum_{i \in \mathfrak{V}_k} \mathbf{x}_i$$

end

Step 4 : **if** $\tilde{\mathfrak{V}}_k \neq \mathfrak{V}_k \ \forall \ k = 1, \ldots, K$ **then**

  Let $\tilde{\mathfrak{V}}_k = \mathfrak{V}_k$ for $k = 1, \ldots, K$. Go to step 1.

else

  Stop.

end

Result : $\mathbf{C}$, *Index Sets* — $\mathfrak{V}_1, \ldots, \mathfrak{V}_K$

However, we must note that the visual appeal of dendrograms softens as the number of observations increase; for large data problems dendrograms are not a suitable substitute for a plot of the covariate values along with their cluster assignment.

A.3 The Gap Statistic

The gap statistic is a popular method used to determine the number of clusters for unsupervised methods. Here the smallest K is selected such that $\text{Gap}(K) \geq$

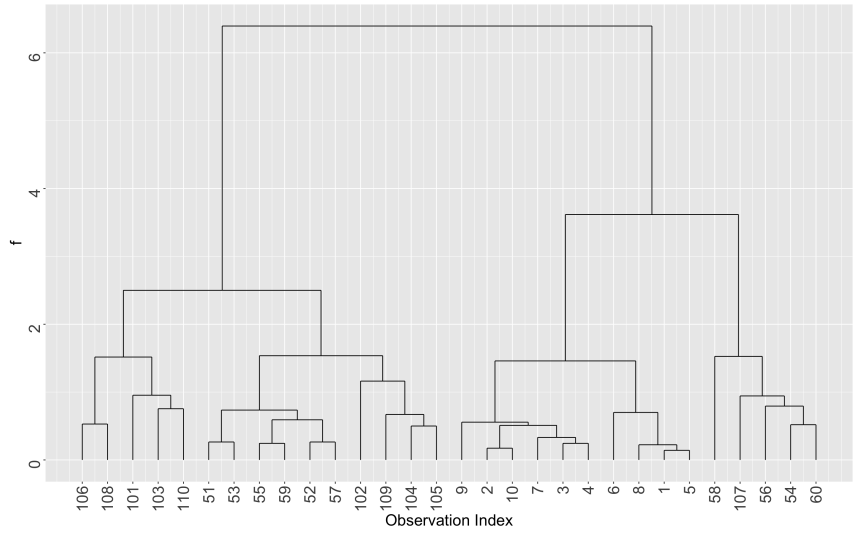


Figure A.2: Dendrogram for agglomerative complete linkage hierarchical clustering on a sample of 30 observations from the iris dataset.

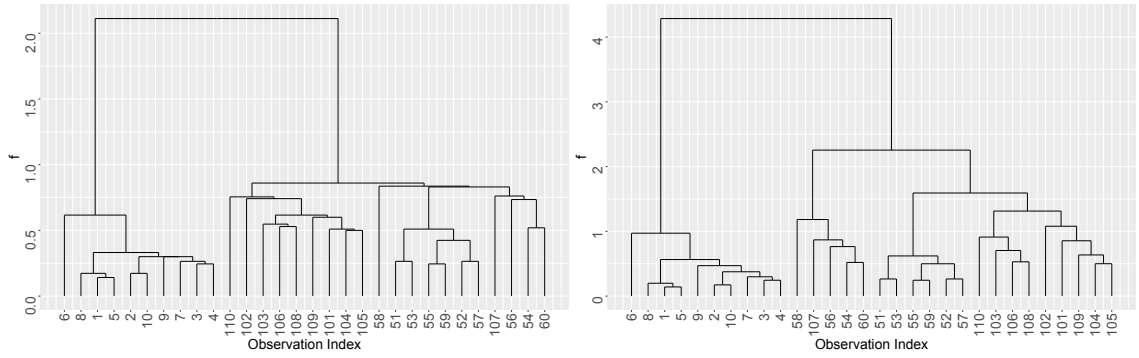


Figure A.3: Dendrograms for agglomerative hierarchical clustering using the single linkage (left) and average linkage (right) methods, on a sample of 30 observations from the iris dataset.

$\text{Gap}(K+1) - s_{K+1}$, where:

$$\omega_K(\mathfrak{V}'_1, \ldots, \mathfrak{V}'_K) = \sum_{k=1}^{K} \frac{1}{2|\mathfrak{V}'_k|} \sum_{i,j \in \mathfrak{V}'_k} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \tag{A.1}$$

$$\bar{\Omega} = \frac{1}{\Omega} \sum_{b=1}^{\Omega} \log(\omega_K(\mathfrak{V}^b_1, \ldots, \mathfrak{V}^b_K)) \tag{A.2}$$

$$\text{Gap}(K) = \bar{\Omega} - \log(\omega_K(\mathfrak{V}_1, \ldots, \mathfrak{V}_K)) \tag{A.3}$$

$$s_K = \sqrt{\frac{\Omega+1}{\Omega^2} \sum_{b=1}^{\Omega} (\log(\omega_K(\mathfrak{V}^b_1, \ldots, \mathfrak{V}^b_K)) - \bar{\Omega})^2} \tag{A.4}$$

Algorithm 27: Hierarchical clustering

Input : *Number of Clusters* — $K > 1$, *Distance Metric* — d, *Covariate Matrix* — $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, *Linkage Function* — f

Step 1 : **if** *Agglomerative* **then**

Let $\mathfrak{V} = \{\mathfrak{V}_1, \ldots, \mathfrak{V}_n\}$ where $\mathfrak{V}_1 = \{1\}, \ldots, \mathfrak{V}_n = \{n\}$

else

Let $\mathfrak{V} = \{\mathfrak{V}_1\}$ where $\mathfrak{V}_1 = \{1, \ldots, n\}$

end

Step 2 : **if** *Agglomerative* **then**

Let

$$(\mathfrak{V}_k, \mathfrak{V}_l) = \arg \min_{\mathfrak{V}_k \neq \mathfrak{V}_l} \{f(d, \mathbf{X}, \mathfrak{V}_k, \mathfrak{V}_l)\}$$

Update $\mathfrak{V}_k = \mathfrak{V}_k \cup \mathfrak{V}_l$ and remove $\mathfrak{V}_l$.

else

for $k = 1, \ldots, \mid \mathfrak{V} \mid$ **do**

Let $\{\tilde{\mathfrak{V}}_{k,1}^{(1)}, \tilde{\mathfrak{V}}_{k,2}^{(1)}\}, \ldots, \{\tilde{\mathfrak{V}}_{k,1}^{(2^{|\mathfrak{V}_k|-1}-1)}, \tilde{\mathfrak{V}}_{k,2}^{(2^{|\mathfrak{V}_k|-1}-1)}\}$ be all possible partitions of $\mathfrak{V}_k$. Let

$$t_k = \arg \max_{t \in \{1, \ldots, 2^{|\mathfrak{V}_k|-1}-1\}} \{f(d, \mathbf{X}, \tilde{\mathfrak{V}}_{k,1}^{(t)}, \tilde{\mathfrak{V}}_{k,2}^{(t)})\}$$

end

Let

$$k^\star = \arg \max_{k \in \{1, \ldots, |\mathfrak{V}|\}} \{f(d, \mathbf{X}, \tilde{\mathfrak{V}}_{k,1}^{(t_k)}, \tilde{\mathfrak{V}}_{k,2}^{(t_k)})\}$$

Let $\tilde{\mathfrak{V}}_{k^\star,1}^{(t_{k^\star})} = \mathfrak{V}_{|\mathfrak{V}|+1}$ and $\tilde{\mathfrak{V}}_{k^\star,2}^{(t_{k^\star})} = \mathfrak{V}_{|\mathfrak{V}|+2}$. Add $\mathfrak{V}_{|\mathfrak{V}|+1}$ and $\mathfrak{V}_{|\mathfrak{V}|+2}$ to $\mathfrak{V}$. Remove $\mathfrak{V}_{k^\star}$ from $\mathfrak{V}$. For $k > k^\star$ update the index of $\mathfrak{V}_k$ to $k - 1$.

end

Step 3 : **if** $|\mathfrak{V}| = K$ **then**

Stop.

else

Go to step 2.

end

Result : *Set of Index Sets* — $\mathfrak{V}$

with Ω specified by the user. The index sets $\mathfrak{V}_1^b, \ldots, \mathfrak{V}_K^b$ are obtained through unsupervised clustering on a simulated dataset $\mathbf{X}^b$ such that

$$x_{ij}^b \sim \mathcal{U}\left(\min_{a \in \{1, \ldots, n\}} \{x_{aj}\}, \max_{a \in \{1, \ldots, n\}} \{x_{aj}\}\right) \tag{A.5}$$

This in essence creates sets of samples from a uniformly distributed hypercube to mimic a scenario where one would typically not expect to discover a clustering structure. We note that this method requires the assumption of log-concave distributions

for the cluster densities, which may not be valid for certain datasets. This method can also create additional uncertainty in the form of estimation and can introduce measures of cluster quality that are not necessarily generally favorable (for example compactness of clusters).

A.4 Clustering Methods Which Treat K as Unknown

Here we give a brief introduction into two methods which apply a Bayesian treatment to the number of clusters K — Reversible Jump Markov Chain Monte Carlo (RJMCMC) and Dirichlet Process Priors (DPP).

RJMCMC acts as a trans-dimensional Metropolis–Hastings algorithm, where at each iteration the algorithm first selects which dimension to move to, generates a certain number of random values ($\mathbf{u}$) such that the dimensions of the current state and the target state match, applies a mapping from the current state and current random values to the proposal state and then accepts this proposal with probability:

$$\min\left\{1, \frac{\pi_{t+1}(\boldsymbol{\Theta}^{\{t+1\}}, k^{\{t+1\}}) j(k^{\{t\}} \mid k^{\{t+1\}}) q(\mathbf{u}^{\{t\}} \mid \mathbf{u}^{\{t+1\}})}{\pi_{t+1}(\boldsymbol{\Theta}^{\{t\}}, k^{\{t\}}) j(k^{\{t+1\}} | k^{\{t\}}) q(\mathbf{u}^{\{t+1\}} \mid \mathbf{u}^{\{t\}})} \det(J)\right\} \tag{A.6}$$

where t is the indicator of where we are in the chain, $j(\cdot)$ is the Probability Density Function (PDF) of the dimension sampler, $q(\cdot)$ is the PDF of the random number sampler and J is the Jacobian of the mapping from $(\boldsymbol{\Theta}^{\{t\}}, \mathbf{u}^{\{t\}})$ to $(\boldsymbol{\Theta}^{\{t+1\}}, \mathbf{u}^{\{t+1\}})$, i.e.

$$J = \left(\frac{\partial(\boldsymbol{\Theta}^{\{t+1\}}, \mathbf{u}^{\{t+1\}})}{\partial \pi(\boldsymbol{\Theta}^{\{t\}}, \mathbf{u}^{\{t\}})}\right) \tag{A.7}$$

The formation of the acceptance probability is such that the resulting Markov Chain is reversible and therefore will converge to desired joint posterior [55].

DPP models offer an entirely different modelling framework whilst still treating K as unknown. There are a vast number of potential applications of DPP models for clustering purposes, such as to determine the number of experts for mixture of experts models [115]. However, by far the most commonly used approach is

an unsupervised infinite mixture of models [56]. This model can then be used for supervised problems using sequential predictive modelling. So for unsupervised DPP models, the set up is as follows:

$$\pi(\mathbf{X} \mid \boldsymbol{\Theta}_1^\star, \ldots, \boldsymbol{\Theta}_n^\star) = \prod_{i=1}^{n} \pi(\mathbf{x}_i \mid \boldsymbol{\Theta}_i^\star) \tag{A.8}$$

$$\boldsymbol{\Theta}_i^\star \sim G \tag{A.9}$$

$$G \sim \mathrm{DP}(G_0, a) \tag{A.10}$$

where DP is a Dirichlet Process [116] with base distribution G_0 and concentration parameter a. A distribution for distributions, centered around G_0, is typically what $\mathrm{DP}(G_0, a)$ is thought of, with draws from $\mathrm{DP}(G_0, a)$ becoming more varied the larger a is. The goal here is to sample from $\pi(\boldsymbol{\Theta}_1^\star, \ldots, \boldsymbol{\Theta}_n^\star \mid \mathbf{X})$, and observations with the same parameters gives the clusters as a result. However, typically the problem can be simplified through consideration of the finite setting and introduction of a cluster assignment variable $\mathbf{z}$:

$$\pi(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K) = \prod_{i=1}^{n} \pi(\mathbf{x}_i \mid \boldsymbol{\Theta}_{z_i}) \tag{A.11}$$

$$\pi(z_i \mid \boldsymbol{\tau}) = \mathrm{Mult}(\boldsymbol{\tau}) \tag{A.12}$$

$$\boldsymbol{\Theta}_k \sim G_0 \tag{A.13}$$

$$\boldsymbol{\tau} \sim \mathrm{Dir}\left(\frac{a}{K}, \ldots, \frac{a}{K}\right) \tag{A.14}$$

From this the conditional distribution for z_i can be derived as

$$\pi(z_i = k \mid \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K) \propto \pi(\mathbf{x}_i \mid \boldsymbol{\Theta}_k) \frac{\sum_{j \neq i} \mathbb{1}(z_j = k) + \frac{a}{K}}{n - 1 + a} \tag{A.15}$$

Thenn we can extend the problem by not constraining the number of clusters to be a finite value. Here we let $K \longrightarrow \infty$, and for clarity we now give the clusters currently represented by $\mathbf{z}$ labels $\{1, \ldots, K'\}$, which gives

$$\pi(z_i = k \mid \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_{K'}) \propto \pi(\mathbf{x}_i \mid \boldsymbol{\Theta}_k) \frac{\sum_{j \neq i} \mathbb{1}(z_j = k)}{n - 1 + a} \tag{A.16}$$

But now in the infinite cluster setting we must consider the scenario in which z_i is equal to a value k which is not in $\{1, \ldots, K'\}$, which has the following probability

$$\pi(z_i \notin \{1, \ldots, K'\} \mid \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_{K'}) \propto \frac{a}{n-1+a} \int \pi(\mathbf{x}_i \mid \boldsymbol{\Theta}) dG_0(\boldsymbol{\Theta}) \quad \text{(A.17)}$$

If we can analytically solve this integral (which is possible if G_0 is a conjugate prior) then we have all the necessary components to sample a cluster for observation i, based on the current state of the other parameters[1]. With the cluster association variables sampled, we will have an updated set of labels $\{1, \ldots, K'\}$, leading to the conditional distributions for each of the parameters $\{1, \ldots, K'\}$ being

$$\pi(\boldsymbol{\Theta}_k | \mathbf{z}, \mathbf{X}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_{k-1}, \boldsymbol{\Theta}_{k+1}, \ldots, \boldsymbol{\Theta}_{K'}) = \pi(\boldsymbol{\Theta}_k | \mathbf{z}, \mathbf{X}_{\mathbb{1}(\mathbf{z}=k),\cdot}) \quad \text{(A.18)}$$

where $\mathbf{X}_{\mathbb{1}(\mathbf{z}=k),\cdot}$ are all observations i such that $z_i = k$. Again if G_0 is a conjugate prior then we shall be able to sample from these distributions directly. All together we have the tools to perform a version of Markov Chain Monte Carlo known as Gibbs sampling [117] to obtain samples from the full posterior. This particular method is one of many described by Neal [56], who gives an excellent insight into how to practically utilise DPP's.

[1]Note here that if observation i is in a cluster of its own, say k, then the probability of sampling cluster k is 0, and so the associated $\boldsymbol{\Theta}_k$ should be removed.

APPENDIX B

UNCOVER Parameter Specification & Dataset Information on Independent Variables

B.1 UNCOVER Parameter Specification

Whilst the amount of parameters required for the specification of an UNCOVER algorithm is vast, this is a deliberate choice to ensure flexibility in specification to either meet a specific stakeholder requirement or to improve computational efficiency. However, without a comprehensive list detailing recommendations for the various parameter specifications, new users of the UNCOVER framework face a daunting task[1]. Therefore, what follows in this section is a table which summarises the default parameters, together with a complete list of the parameters required for UNCOVER. This list offers further guidance and explanation to the table as to which values to select for said parameters. For further context on the purpose of these parameters see chapters 4 and 5.

[1]The `R` package `UNCOVER` [11] alleviates this by providing defaults for the majority of parameters for the function `UNCOVER`.

	Parameter	Default
Minimum Spanning Tree Construction	*Variable Subset* — $\mathfrak{P}$	$\{1, \ldots, p\}$
Iterated Batch Importance Sampling	*Number of SMC Samples* — N	1000
	ESS Threshold — ξ	$0.5 \times N$
Early Termination	*Stopping Criterion* — $\varkappa$	n
Deforestation Criteria	*Maximum Number of Clusters* — κ	$< \varkappa$
	Minimum Size for Clusters — $\bar{\mathfrak{K}}$	$> n \times \varkappa^{-1}$
	Maximal Regret Factor — ν	> 100
	Training Data Fraction — o	0.8
	Minimum Number of Minority Class Observations for Clusters — υ	$> \frac{n^{\dagger}}{\varkappa}$
Memoisation	*Cache Evaluation Threshold* — ρ	$\max\{1, n \times 2^{-\varkappa}\}$
Reverse Iterated Batch Importance Sampling	*RIBIS Observation Threshold* — $\bar{\rho}$	30
Asymptotic Approximations	*Asymptotic Approximation Threshold* — $\bar{n}$	$> \rho$

Table B.1: UNCOVER parameters, along with their deafults (or more generalised properties of the parameter recommendations if a natural default is not available). Parameters are also grouped into distinct aspects of the UNCOVER algorithm.

- Variable Subset — $\mathfrak{P}$: This parameter can be specified either through stakeholder requirements or variable selection methods with preliminary UNCOVER runs. In lieu of these options, one can use $\{1, \ldots, p\}$ (where p is the number of covariates) as a default.

- *Number of SMC samples* — N: As N has a direct effect on the accuracy of Bayesian evidence estimations, one recommends selecting N to be as large as computationally feasible. However, UNCOVER requires the generation of several Bayesian evidences and so in practical terms lower values of N might be necessary. As a rough guide, through experimentation with several datasets, 1000 samples appears to give robust outcomes[2].

- *ESS Threshold* — ξ: The effective sample size again has a direct impact on the accuracy of the Bayesian evidence estimate, as the more frequently the samples are rejuvenated through a move step the more likely the samples are to be

[2]Note that in the seminal paper for IBIS [9], Chopin recommended much larger values for N. In this paper, however, the IBIS scheme was not intended to be used multiple times as part of a larger algorithm.

representative of the target distribution, which in turn results in the Bayesian evidence estimate being more likely to be accurate. As a consequence, the desired specification of ξ would naturally be N such that we rejuvenate at every iteration of the IBIS scheme. This is not computationally efficient, however, and so a lower value of ξ is typically required. Through experimentation and suggested defaults in other pieces of work [118] this lower recommended value is $\frac{N}{2}$.

- *Stopping Criterion* — $\varkappa$: As $\varkappa$ only aids computational efficiency, the theoretical default is to use the number of observations n as this effectively removes the criterion from the algorithm. However, practically, one should aim to select $\varkappa$ as a value slightly above the number of suspected clusters. Explaining further, when we have obtained the true clusters (assuming the number of true clusters was equal to the number of suspected clusters) additional overfitting steps in UNCOVER are time consuming and unnecessary, as they will only be rectified in the deforestation stage. However, selecting $\varkappa$ slightly above the number of suspected clusters gives UNCOVER the flexibility to temporarily create more clusters than necessary if earlier mistakes were made due to the greedy nature of the algorithm.

- *Maximum Number of Clusters* — κ: Selection of κ should either be specified by the stakeholder or should be a conservative estimate on the number of suspected clusters. In general $\kappa < \varkappa$ to ensure the deforestation stage is impactful if the stopping criterion is met. Note κ only needs to be specified if the 'Number of Clusters' criterion is selected.

- *Minimum Size for Clusters* — $\daleth$: Selection of $\daleth$ should either be specified by the stakeholder or should be a judgement on the minimum number of observations a cluster would require to detect a regression signal. The latter point will depend on factors such as general response diversity and local response diversity in specific areas of the covariate space. In general $\daleth > \frac{n}{\varkappa}$ to ensure the deforestation stage is impactful if the stopping criterion is met. Note $\daleth$ only needs to be specified if the 'Size of Clusters' criterion is selected.

- *Maximal Regret Factor* — ν: As mentioned previously, maximal regret is intrinsically linked to Bayes Factors (rearranging equation (4.22) presents ν as an upper bound on a Bayes factor with the current model as the null model). This suggests using Jeffreys scale [53] to select a default, i.e. as we want the evidence for the current model to be decisive to not reintroduce an edge, we should set $\nu = 100$. However, this assumes the current model at the end of the planting stage is an acceptable output (hence being labeled the null model) when in fact without reintroducing edges the current model is likely to have succumb to overfitting. Therefore, we recommend a value for ν in general to be larger than 100 to encourage edge reintroduction. Note ν only needs to be specified if the 'Maximal Regret' criterion is selected.

- *Training Data Fraction* — o: The common choice for train:validation splits of the dataset is $80:20$, i.e $o = 0.8$. As discussed in section 4.6.1, for UNCOVER this is not always optimal as a small number of validation observations makes it less likely that small clusters outputted in the planting stage can be adequately assessed. As a result, in general we recommend 0.8 as a default but with the caveat that for large data problems smaller values of o are advised. Note o only needs to be specified if the 'Validation Data' criterion is selected.

- *Minimum Number of Minority Class Observations for Clusters* — υ: As seen in section 4.6.1, relatively small values of υ lead to promising results. In general, however, specification of υ should be made with consideration to the response diversity of the overall dataset. One has more freedom to select a larger value of υ with a balanced response dataset than with an imbalanced response dataset. Typically, $\upsilon > \frac{n^\dagger}{\varkappa}$ (where $n^\dagger$ is the number of observations in the total dataset which have an associated response in the minority class) to ensure the deforestation stage is impactful if the stopping criterion is met. Note υ only needs to be specified if the 'Diverse Response' criterion is selected.

- *Cache Evaluation Threshold* — ρ: Section 5.2.1 highlights the difficultly in specifying a ρ which is universally computationally efficient, as the computational efficiency depends on various factors such as the minimum spanning

tree structure, the cache size, the choice of prior et cetera. As a basic rule, however, one can specify that $\rho = \max\left\{1, \frac{n}{2^{\varkappa}}\right\}$, as this ensures that for every possible cluster split one of the two new clusters formed will require the cache to be checked. This value may not give the optimal choice for ρ for a given problem, but (for values of $\varkappa$ where $n \geq 2^{\varkappa}$) it allows the algorithm to move away from the extremes of never checking the cache or always checking the cache.

- *RIBIS Observation Threshold* — $\bar{\rho}$: Specification of $\bar{\rho}$ is problem specific and strongly related to how close the prior is to target distribution for any posterior encountered during the UNCOVER algorithm. Due to Bernstein von Mises theorem [41] asymptotically the transformation in RIBIS will be appropriate which suggests setting $\bar{\rho}$ to be large, but in practice this is computationally inefficient as the RIBIS algorithm is then rarely utilised. We have found through experimentation on the datasets presented in this thesis that relatively low values of $\bar{\rho}$ are robust, and so we give an ad-hoc recommendation of $\bar{\rho} = 30$.

- *Asymptotic Approximation Threshold* — $\bar{n}$: Like the RIBIS observation threshold, specification of $\bar{n}$ is largely dependent on the choice of prior (uninformative priors can result in posteriors that are well approximated by a transformation of the BIC for small values of $\bar{n}$ for example). Therefore, whilst we do not give a specific default for $\bar{n}$, we do recommend specifying $\bar{n}$ to be large. Additionally, $\bar{n} > \rho$ to ensure that it is possible to check the cache for the SMC sampler.

Whilst not necessarily falling under the term 'parameter', there is one final specification for UNCOVER that is crucial — the base prior $\pi(\cdot)$. The form of the prior is left to the discretion of the user, however, for computational efficiency it is recommended that for the prior chosen i.i.d. samples are easily obtainable (as prior samples are required to initialise the IBIS scheme). Additionally, if one intends on using features which require asymptotic approximations (i.e. $\bar{\rho}$ and $\bar{n}$), then care needs to be taken to ensure that the Bernstein von Mises theorem holds with the choice of prior. For example, for clusters encountered in UNCOVER whose poste-

riors require an asymptotic approximation of the Bayesian evidence, a multivariate uniform prior covering an area of low posterior density is unlikely to satisfy the conditions of the Bernstein von Mises theorem. By this reasoning we recommend the multivariate normal $\mathcal{N}_{p+1}(\cdot,\cdot)$ as a default, as the support is $\mathbb{R}^{p+1}$ and it takes the same distributional form as the posterior when the number of observations tends to infinity (provided the Bernstein von Mises theorem holds).

Finally, with the default form of the prior being a multivariate normal, one must also specify the prior mean and prior variance. Experimentation with a multivariate prior in section 4.6.1 shows that if one has prior knowledge (possibly gained from expert opinion) on the posterior means for each of the suspected clusters, then selection of a prior mean 'close' to all posterior means is advised. This clearly will not always be possible, and so as a secondary default we recommend selecting the prior mean $\boldsymbol{\mu} = (0, ...0)^T$, as this should perform well with a scaled dataset. Additionally, one should select a diffuse[3] prior variance Σ, in order to mitigate the negative effects on the output that can occur if $\boldsymbol{\mu}$ is ill-placed.

B.2 Dataset Information on Independent Variables

B.2.1 Customers Data

Attribute	Type	Summary
Sex	Categorical	*Count*: 'Female' (112), 'Male' (88)
Age	Numerical	*Mean*: 38.85, *SD*: 13.96901
Income (k — $)	Numerical	*Mean*: 60.56, *SD*: 26.26472

Table B.2: Mall customers dataset variables, along with their type and summary information (either mean and standard deviation or factor counts).

[3]How diffuse the variance should be is problem specific, however, experimentation has shown the prior variance $\Sigma = 16\mathcal{I}_{p+1}$ as an ad-hoc default performs well with prior mean $\boldsymbol{\mu} = (0, ...0)^T$.

B.2.2 Wine Quality Data

Attribute	Type	Summary
Fixed Acidity	Numerical	*Mean*: 7.21552275, *SD*: 1.319776662
Volatile Acidity	Numerical	*Mean*: 0.34417074, *SD*: 0.168264321
Citric Acid	Numerical	*Mean*: 0.31852200, *SD*: 0.147176538
Residual Sugar	Numerical	*Mean*: 5.04960511, *SD*: 4.500645455
Chlorides	Numerical	*Mean*: 0.05670045, *SD*: 0.036864803
Free Sulfur Dioxide	Numerical	*Mean*: 30.03046258, *SD*: 17.804364756
Total Sulfur Dioxide	Numerical	*Mean*: 114.10774727, *SD*: 56.783847640
Density	Numerical	*Mean*: 0.99453624, *SD*: 0.002965541
pH	Numerical	*Mean*: 3.22463896, *SD*: 0.160403301
Sulphates	Numerical	*Mean*: 0.53340165, *SD*: 0.149752704
Alcohol	Numerical	*Mean*: 10.54922214, *SD*: 1.185963672
Colour	Categorical	*Count*: 'Red' (1359), 'White' (3959)

Table B.3: Wine quality dataset variables, along with their type and summary information (either mean and standard deviation or factor counts).

B.2.3 Abalone Data

Covariate	Type	Summary
Sex	Categorical	*Count*: 'Female' (1307), 'Infant' (1342), 'Male' (1528)
Shell Length	Numerical	*Mean*: 0.5239921, *SD*: 0.12009291
Shell Diameter	Numerical	*Mean*: 0.4078813, *SD*: 0.09923987
Shell Height	Numerical	*Mean*: 0.1395164, *SD*: 0.04182706
Whole Weight	Numerical	*Mean*: 0.8287422, *SD*: 0.49038902

Table B.4: Abalone dataset variables, along with their type and summary information (either mean and standard deviation or factor counts).

B.2.4 Heart Disease Data

Attribute	Type	Summary
age	Numerical	*Mean*: 54.54208754, *SD*: 9.0497357
sex	Categorical	*Count*: 'Female' (96), 'Male' (201)
trestbps	Numerical	*Mean*: 131.69360269, *SD*: 17.7628064
chol	Numerical	*Mean*: 247.35016835, *SD*: 51.9975825
fbs	Categorical	*Count*: 'Above 120 mg/dl' (43), 'Equal/Below 120 mg/dl' (254)
restecg	Categorical	*Count*: 'Normal' (147), 'ST—T Wave Abnormality' (4), 'Probable/Definite Left Ventricular Hypertrophy' (146)
thalach	Numerical	*Mean*: 149.59932660, *SD*: 22.9415621
exang	Categorical	*Count*: 'Yes' (97), 'No' (200)
oldpeak	Numerical	*Mean*: 1.05555556, *SD*: 1.1661228
slope	Categorical	*Count*: 'Downsloping' (21), 'Flat' (137), 'Upsloping' (139)

Table B.5: Heart disease dataset variables, along with their type and summary information (either mean and standard deviation or factor counts).

B.2.5 Car Data

Attribute	Type	Summary
Buying Price	Categorical	*Count*: 'Low' (432), 'Medium' (432), 'High' (432), 'Very High' (432)
Maintenance Price	Categorical	*Count*: 'Low' (432), 'Medium' (432), 'High' (432), 'Very High' (432)
Number of Doors	Categorical	*Count*: 'Two' (432), 'Three' (432), 'Four' (432), 'Five or More' (432)
Capacity	Categorical	*Count*: 'Two' (576), 'Four' (576), 'More Than Four' (576)
Boot Size	Categorical	*Count*: 'Small' (576), 'Medium' (576), 'Big' (576)
Safety	Categorical	*Count*: 'Low' (576), 'Medium' (576), 'High' (576)

Table B.6: Car dataset variables, along with their type and summary information (factor counts).