THE UNIVERSITY OF GLASGOW

Automatic Architecture Selection for Hierarchical Mixture of Experts Models

by

Ivona Voroneckaja

Dissertation Submitted to the
University of Glasgow
for the degree of
Doctor of Philosophy

School of Mathematics & Statistics

March 14, 2023

Declaration of Authorship

I, Ivona Voroneckaja, declare that this thesis titled, 'Automatic Architecture Selection for Hierarchical Mixture of Experts Models' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

THE UNIVERSITY OF GLASGOW

Abstract

School of Mathematics & Statistics

Doctor of Philosophy

by Ivona Voroneckaja

Hierarchical mixture of experts (HME) is a powerful tree-structured modeling technique based on the divide and conquer principle. HME model trees consist of two types of nodes - gate nodes, which are responsible for splitting a large complex problem into several smaller subproblems, and expert nodes, which perform the corresponding subproblem-solving. Selecting the number of such nodes as well as the order in which they are arranged is, however, a non-trivial task. A commonly used approach involves fitting several architectures and using methods such as cross-validation to pick the best one. As well as being computationally intensive, this method first requires one to pick the set of architectures to consider. For complex models with a large number of architectural elements, this leads to an unmanageable number of potential options. Pre-setting model architecture also requires choosing initial parameter values, which becomes progressively more challenging as parameter dimensionality increases. The latter challenges could be addressed by growing trees during the model fitting process instead of selecting the architecture in advance. It is thus evident that HME models suffer from a lack of a flexible and adaptive way of performing automatic architecture selection.

The work presented in this thesis proposes automatic architecture selection methods for HME models, which allow for adding and removing tree nodes as well as adjusting the order in which they are arranged. As part of the development, three Bayesian parameter sampling strategies are proposed and systematically evaluated resulting in a recommended strategy. An adaptation of the Reversible Jump (RJ) algorithm is then used to grow and prune HME model trees. The main downfall of the RJ, which lies in low acceptance rates, is addressed by the addition of a novel reversible jump proposal algorithm. A new Gate Swaps (GS) algorithm is then proposed to tackle the problem of changing the order in which the existing tree nodes are arranged. Both algorithms are evaluated on two real-life problems with a particular focus on the Glasgow rental property prices data. It is shown that HME models fitted using the proposed RJ GS MCMC yield accurate predictions as well as provide an exceptionally high level of model interpretability, which is unusual amongst other machine learning methods.

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr Nema Dean and Dr Ludger Evers, who have given me immense help and support throughout the years. I consider myself to be incredibly lucky to have had you both with me every step of the way.

I would also like to thank the School for the Maclaurin Scholarship funding, which made this work possible.

A very special thank you to my husband Tristan for always reminding me what I am capable of, even when I do not believe it myself.

Dėkoju savo tėveliams Stanislavui ir Irinai, kad visada buvote šalia, nors mus ir skiria 2,734 km. Visada branginsiu Jūsų palaikymą ir tikėjimą manimi.

Ypatingai dėkoju savo seneliams Arvydui ir Galinai, kurie tiek daug investavo į mano tobulėjimą, daugelį metų vedė mane į popamokinius anglų kalbos kursus ir niekada manimi neabejojo.

Contents

List of Figures

List of Tables

To my younger self

Chapter 1

Introduction

Machine learning is a branch of artificial intelligence that uses data and algorithms to provide valuable insight and make high-quality predictions (Wang et al., 2022). It has been widely used to solve both regression and classification problems in real-world applications, such as speech recognition, visual classification, collaborative filtering, and automatic translation (Smola, 2008).

One of the widely used algorithm strategies in machine learning is called *divide and conquer*. The divide step of the strategy involves splitting a large, complex problem into several smaller subproblems, which are only defined on a subset of data. Given that the subsets of data are sufficiently simple, the subproblems can then be solved efficiently. In the conquer step of the strategy, the solutions to these subproblems are combined to produce a solution to the initial larger scale problem. Efficient steps of the algorithm yield the benefits of structural simplicity, computational efficiency, and parallel implementation (Smith, 1985). Structural simplicity is ensured by a careful choice of the programming language constructs that allow expressing divide and conquer algorithms concisely. Divide and conquer algorithms are *cache-oblivious* and make efficient use of memory caches by only accessing the memory of the subproblem in question instead of the slower main memory (Frigo et al., 1999). Lastly, parallel implementation means that the smaller subproblems can be solved at the same time, and hence, given their simplicity, further improve the computational efficiency. Despite the appeal of the divide and conquer algorithms, developing them is a nontrivial task. One must ensure that the problem is divided into subproblems that can indeed be solved efficiently as well as propose a suitable way to combine these solutions. Examples of divide and conquer models include decision trees (Loh, 2014), random forests (Breiman, 2001), mixture of experts (ME) models, and hierarchical mixture of experts (HME) models, which are the main focus of this thesis.

An HME model is a tree-structured model based on the divide and conquer principle where the problem space is divided between the so-called *experts*. A simple example of the HME model architecture with five experts is shown in Figure 1.1, where named boxes correspond to the elements of the tree, called *nodes*, which are then joined by the *edges* of the tree. For HME models, the nodes can be of two types - *gate* nodes and *expert* nodes. The green gate nodes perform the task of partitioning the space while the blue expert nodes are responsible for problem-solving. An ME model is a special case of the HME model with only one gate, for instance, the segment containing $G3$, $E2$ and $E3$ only from Figure 1.1 is an ME model.

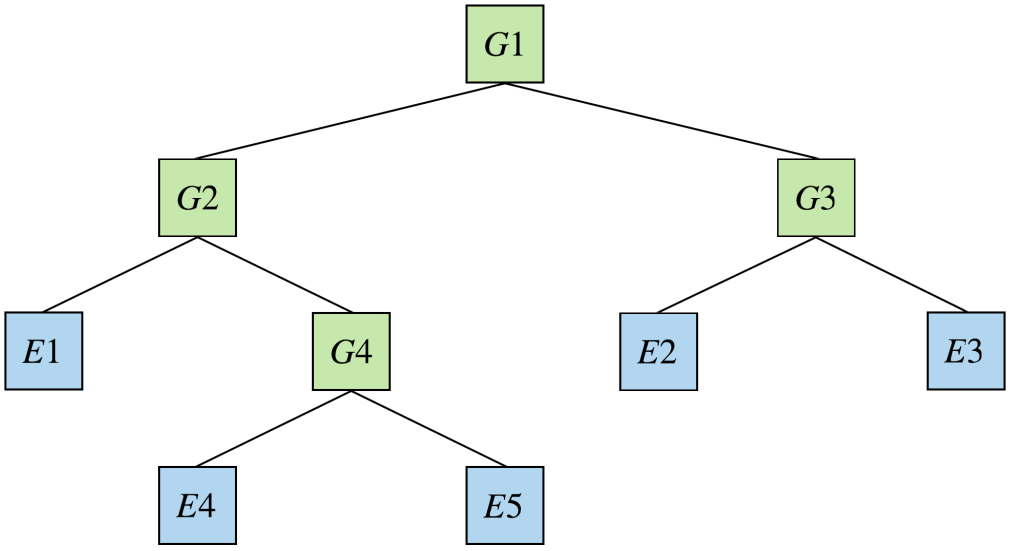


FIGURE 1.1: Illustration of an HME model with five experts.

The idea of ME models was first introduced by Jacobs et al. (1991) and extended to HME models by Jordan and Jacobs (1994) (detailed background on the ME models is presented in Chapter 2). HME models are widely used for solving both regression and classification problems based on soft probabilistic splits of the input space (Bishop and Svenskn, 2002). Each expert in the tree has its own associated expert density function. The most common choice of expert densities, discussed in Chapter 5, is Gaussian (linear regression) expert densities (Waterhouse et al., 1996). A non-normal mixture of experts, which can deal with possibly skewed, heavy-tailed data, is introduced by Chamroukhi (2015). Another popular choice for expert densities is generalised linear models (GLM) (McCullagh and Nelder, 1989).

The documented applications of the hierarchical mixture of experts models include time-series data (Huerta et al., 2003) and the well-known application to the speech recognition

data (Peng et al., 1996). The latest proposed extension to the hierarchical mixture of experts family is the hierarchical routing mixture of experts (HRME) (Zhao et al., 2019), which explores the idea of separating the output variables by jointly partitioning the input and output spaces. The latter framework is based on simple regression models assigned to each of the resulting partitions.

Originally, the parameters of HME have been estimated by frequentist inference (discussed in Chapter 3). The most common way of performing maximum likelihood estimation for this setting is using the expectation maximisation (EM) algorithm (Jordan and Jacobs, 1994). Within the thesis, we show that the parameters of the HME model cannot be found in closed-form by maximising the log-likelihood function. A well-known technique of introduction of latent assignment variables is implemented and leads to the complete data likelihood function definition. The latter in turn simplifies parameter inference by allowing to sample the expert parameters explicitly (Hurn et al., 2003; Diebolt and Robert, 1994). The complexity of the HME model architecture usually results in local maxima thus requiring multiple starting points when maximising the likelihood function (Huerta et al., 2003). A preferred way to estimate the model parameters is to use the Bayesian approach covered in Chapter 4 (Huerta et al., 2003).

Bayesian inference is most often achieved using Markov Chain Monte Carlo (MCMC) methods, spanning a range of algorithms for sampling from a probability density function. The most widely used algorithms include the Metropolis-Hastings algorithm (Hastings, 1970; Chib and Greenberg, 1995) and the Gibbs sampler (Gelfand, 2000; Geman and Geman, 1984). In some cases, a mixture of several samplers is required to sample the parameters of the distribution of interest. In Chapter 5, three Bayesian parameter sampling strategies are proposed and systematically compared in a special case of Gaussian experts. The first strategy involves sampling and retaining all parameters of the model, including the latent assignment variables. The second technique is the so-called *brute-force* sampling method, which does not take advantage of the parameter inference simplification resulting from the introduction of latent assignment variables. Finally, the third sampler proposes integrating out the Gaussian expert parameters and sampling only the remaining model parameters. Once the preferred parameter sampling strategy is determined, the problem of the model architecture selection is tackled next.

The main aim of this thesis is to develop a novel method for automatic architecture selection in HME models. Architecture selection for HME models is the process of choosing the total number of expert and gate nodes as well as how these nodes are arranged in the model tree. A literature review of the currently available HME architecture selection methods is presented in Chapter 6. In this thesis, an adaptation of the reversible jump (RJ) algorithm, which is a method for constructing reversible Markov chain samplers

first introduced by Green (1995), is proposed. The RJ MCMC is used to determine the number of experts and gates in an HME model. The general appeal of reversible jump lies in the natural generalisation of the existing Markov chain methods (Sisson, 2005). In fact, the RJ is a rather complex extension of the previously mentioned MH algorithm, which allows for exploring the sample space within a fixed dimension as well as making changes in dimensionality. The latter is particularly important for HME models, where both growing and pruning the tree means adding and removing gate and expert nodes, respectively, which in turn results in parameter dimensionality changes. The biggest challenge posed by the RJ algorithm is also addressed - the typically low acceptance rate, which is often caused by uninformed jumps (Al-Awadhi et al., 2004; Ehlers and P. Brooks, 2008; Farr et al., 2015; Brooks et al., 2003). A methodology for proposing intelligent reversible jumps is developed, presented, and evaluated against the naive RJ algorithm. While the proposed RJ methodology tackles the problem of choosing the number of nodes in the tree, it does not address the order in which these nodes are arranged.

The second method of architecture selection for HME models offers a way of adjusting the tree architecture by swapping the existing nodes, which does not change the total number of nodes in the model. The proposed method is named the *Gates Swap* (GS) algorithm and is presented in Chapter 7. The latter type of architecture selection has the potential to improve mixing, allow for better exploration of the model architecture space as well as escape unfortunate splitting decisions previously made. Lastly, combining the RJ and GS architecture selection algorithms together allows one to propose and consider models which would have been missed otherwise. The addition of the GS algorithm to the RJ MCMC is evaluated in Chapter 7.

The RJ MCMC and the RJ GS MCMC are used to fit HME models to two real-life applications in Chapters 6, 7, and 9. Both methods of architecture selection are also systematically evaluated against two competitors discussed in Chapter 8 - the Generalised Additive Model (Hastie and Tibshirani, 1990) and the Bayesian Additive Regression Trees (Chipman et al., 2010b).

The work presented within this thesis required creating the implementation of HME models. In order to take advantage of the previously discussed structural simplicity of the divide and conquer algorithms, the object-orientated programming language of C++ was selected for the task. As a result, a fully functional implementation of the hierarchical mixture of Gaussian experts with both the reversible jump and the gate swap algorithms has been designed and built.

Finally, the results of the undertaken work are summarised, caveats and liminations are discussed, and potential future research opportunities are outlined in Chapter 10.

Chapter 2

Definition of Hierarchical Mixture of Experts Model

This chapter aims to introduce and define hierarchical mixture of experts models. Firstly, mixture models are discussed in Section 2.1, which offers some background information as well as the definition of the model density function. Next, the definition of mixture models is extended to mixtures of experts models in Section 2.2. Similarly, the background and well-known applications of the mixtures of experts are discussed followed by the definition of the model density function. Lastly, hierarchical mixtures of experts models are introduced in Section 2.3 starting with the underlying architecture, which is illustrated using a simple example seen in Chapter 1. All model parameters and other elements of hierarchical mixtures of experts are defined alongside the corresponding proposed notation resulting in the definition of the model density function.

2.1 Mixture Model

2.1.1 Background on Mixture Models

Mixture models (mixtures) are weighted sums of several probability density functions and hence can be used to represent densities with multiple modes. Mixtures have gained a lot of popularity due to their flexibility (McLachlan and Peel, 2004). The first major analysis involving mixture models consisted of fitting a mixture of two normal probability density functions with different means and variances (Pearson, 1894). A Gaussian mixture model (GMM) remains to be the most widely used type of mixture model to this day. Amongst others, the applications of GMM include text-independent speaker identification (Reynolds and Rose, 1995), texture and color for image database retrieval

(Permuter et al., 2006), and market volatility modeling (Brigo and Mercurio, 2002). Mixture models are closely related to mixtures of experts and hence hierarchical mixtures of experts models. Thus we start by defining the mixture model density function.

2.1.2 Mixture Model Density Function

Let $\boldsymbol{\phi}$ denote a vector of all parameters in the mixture model. The mixture model density function for outcome y_i conditional on $\mathbf{x}_i$ is then defined as

$$f(y_i|\mathbf{x}_i, \boldsymbol{\phi}) = \sum_{C \in \mathcal{C}} \pi^{(C)} f^{(C)}(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(C)}), \tag{2.1}$$

where y_i is the response for the i-th observation in the model and $\mathbf{x}_i$ is a vector of corresponding covariates (for $i = 1, ..., n$). A mixture model consists of *components*. The set of all components in the model is denoted as $\mathcal{C}$. Each component $C \in \mathcal{C}$ has its own *component density*, $f^{(C)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(C)}\right)$, where $\boldsymbol{\theta}^{(C)}$ denotes a vector of distinct parameters occurring in the component C density. The component densities must be valid probability density functions, i.e. $f^{(C)}(\cdot) > 0$ and $\int_y f^{(C)}(y)dy = 1$ for $C \in \mathcal{C}$. The component densities appearing in (2.1) do not need to belong to the same parametric family, however in most applications, this will be the case (Titterington et al., 1985). All of the components in the mixture model have corresponding weights, which are called *mixing weights* and are denoted as $\pi^{(C)}$, where $\pi^{(C)} > 0$ and $\sum_{C \in \mathcal{C}} \pi^{(C)} = 1$.

In the next section, the definition of mixture models is extended to the definition of the mixture of experts model.

2.2 Mixture of Experts Model

2.2.1 Background on Mixture of Experts Model

Mixture of experts is one of the most popular divide and conquer methods, which is based on dividing the problem space into several problem solver experts supervised by a gating network (Masoudnia and Ebrahimpour, 2014). Mixture of experts models were originally developed as a machine learning technique (Jacobs et al., 1991), however, have been widely used in a number of areas since. Amongst others, examples include mixed models (Wang et al., 1996), latent class regression models (DeSarbo and Cron, 1988), concomitant-variable latent-class models (Dayton and Macready, 1998) as well as the methodology for switching regression models (Quandt, 1972). Mixture of experts models

have been shown to perform well on different types of data. Gormley and Murphy (2008) applies a mixture of experts to the rank data in elections studies and Gormley and Murphy (2010a) uses a mixture of experts in clustering ranked preference data. Further examples include modelling social network data (Gormley and Murphy, 2010b), time-series data, such as labor market entry and earnings dynamics (Frühwirth-Schnatter et al., 2012), and longitudinal data (Tang and Qu, 2016).

More background on mixture of experts models can be found in McLachlan and Peel (2000), Yuksel et al. (2012) and Frühwirth-Schnatter (2006), while Masoudnia and Ebrahimpour (2014) present a literature survey for mixture of experts models from a machine learning perspective. Gormley and Frühwirth-Schnatter (2018) provide more detail on the inclusion of covariates in the mixture of experts models. Identifiability of a mixture of experts is discussed in Jiang and Tanner (1999b).

We proceed by defining the density function for the mixture of experts model and highlighting the main difference when compared to the mixture model.

2.2.2 Mixture of Experts Model Density Function

A mixture of experts model (ME) can be viewed as an extension of the mixture model discussed in Section 2.1. In the ME model, the components are called *experts*. Let $\mathcal{E}$ denote the set of all experts in the model and let E be an expert from the set $\mathcal{E}$. As before, let y_i represent the response for the i-th observation in the model with a corresponding vector of covariates, $\mathbf{x}_i$, for $i = 1, ..., n$. In the mixture of experts model, the mixing weights $\pi_i^{(E)}$ are individual specific and depend on the associated covariates $\mathbf{x}_i$ through a logistic function

$$\pi_i^{(E)} = \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(E)}\right)}{\sum_{E \in \mathcal{E}} \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(E)}\right)}, \tag{2.2}$$

for $i = 1, ..., n$, where the mixing proportions are controlled by the *gating parameters* $\left(\boldsymbol{\gamma}^{(E)}\right)_{E \in \mathcal{E}}$ with $\boldsymbol{\gamma}^{(E)} = \mathbf{0}$ for the first $E \in \mathcal{E}$ in order to ensure identifiability (Yuksel et al., 2012). The mixing proportions allow for soft probabilistic boundaries between the experts in the model and must satisfy $\sum_{E \in \mathcal{E}} \pi_i^{(E)} = 1$ for all $i = 1, ..., n$. Given (2.2) we can now write down the mixture of experts density as follows

$$f(y_i|\mathbf{x}_i, \boldsymbol{\phi}) = \sum_{E \in \mathcal{E}} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right), \tag{2.3}$$

where $\boldsymbol{\phi}$ denotes a vector of all expert parameters in the mixture of experts model while $\boldsymbol{\theta}^{(E)}$ denotes a vector of distinct parameters occurring in the expert E density, $f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right)$.

The main difference between a mixture and a mixture of experts lies in the expression of the mixing proportion. In the mixture model, the mixing proportion is constant for all observations while in the mixture of experts, the mixing weight is different for each individual observation and depends on the covariates. In the following section, we introduce the hierarchical mixture of experts model and explain how an ME model is a special case of the HME model.

2.3 Hierarchical Mixture of Experts Model

This section first introduces the tree architecture of HME models by providing a toy example. The relationship between mixtures of experts and hierarchical mixtures of experts is then explained. Next, the key notation for all elements of an HME model is proposed. Finally, the HME density function is stated and defined.

2.3.1 HME Model Architecture

Hierarchical mixture of experts (HME) models are based on a tree architecture, an example of which, seen in Chapter 1, is shown again in Figure 2.1. The elements of the tree, called *nodes*, are joined by the so-called *edges* of the tree. For HME models, the nodes can be of two types - gate nodes and expert nodes. In any tree, nodes that do not have any descendants are also called *terminal* nodes or *leaves*. The HME tree has gate nodes (green) at the non-terminal nodes and expert nodes (blue) at the leaves (Fritsch et al., 1996). The first gate node of the tree is usually referred to as the *root* node. All nodes, descending from some node H, are called the *children* of H, while H is, in turn, called their *parent*. By the nature of this architecture design, gate nodes can be both parents and children while expert nodes can only be children. Two children nodes that share a parent are also referred to as *sibling* nodes. Any node that is located closer to the root node than some node H is called a *senior* node with respect to node H. Similarly, any node that is located further down the tree than some node H is called a *junior* node with respect to node H.

Every non-terminal node of the tree corresponds to a logistic gate, which computes a probability of an observation going down the tree in each direction resulting in soft probabilistic splits. Viewing HME models as a divide and conquer approach, gating

networks divide the problem into smaller subproblems while expert nodes perform the problem-solving duty, i.e. they produce an output of interest.

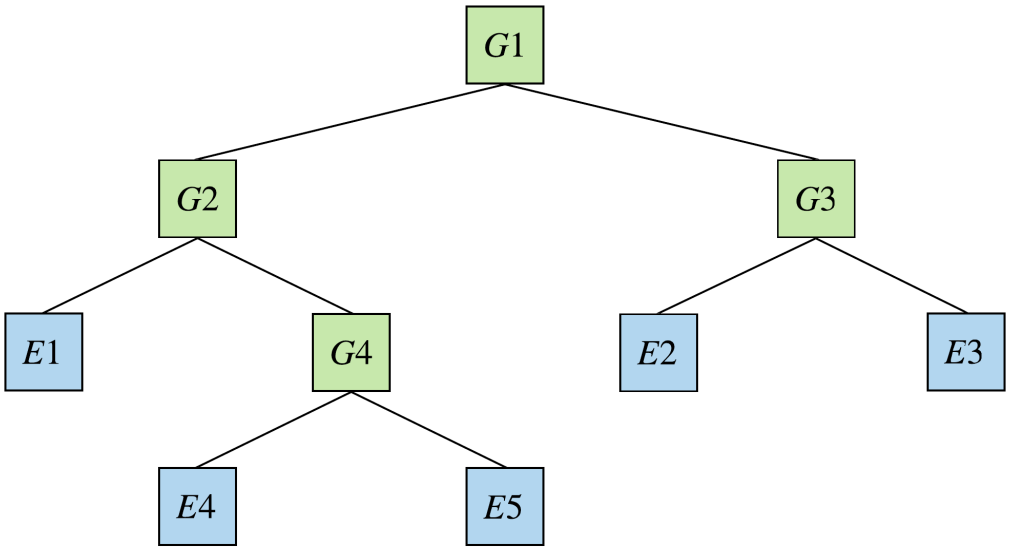


FIGURE 2.1: Illustration of an HME model with five experts equivalent to Figure 1.1.

In the case of the HME model depicted in Figure 2.1, an observation enters the model via the root gate $G1$, which then produces probabilities of it being passed on to the left gate $G2$ and the right gate $G3$. At the left gate $G2$ another calculation is made, i.e. obtaining probabilities to be passed on to the expert $E1$ or to continue the journey via the middle gate $G4$ and so on until the observation reaches a terminal node. For example, one of the possible paths an observation can take is $(G1, G2) \rightarrow (G2, G4) \rightarrow (G4, E4)$. The latter path would have an associated probability of the particular observation reaching expert $E4$, called the *path probability*. Similar to ME models, each expert in the tree has its own associated *expert density function*.

Having introduced the architecture of HME models, it is evident that an ME model is a special case of the HME model with a tree depth equal to one. Similarly, HME models can be thought of as mixture of experts models, where experts are mixtures of experts themselves. Hence all the literature and methods discussing mixture of experts models are relevant and can be extended to hierarchical mixture of experts models.

Following an introduction to the tree architecture, the next section presents the notation for node numbering and paths used within this thesis.

2.3.2 Node Numbering and Paths in HME Models

In order to identify each node, a unique index is assigned to both expert and gate nodes in the tree. As shown in Figure 2.1, gate and expert nodes are numbered separately. The node index is assigned from left to right generation-wise, i.e. all children of each gate are numbered first, followed by its grandchildren, great-grandchildren and so on.

To allow for writing down the definition of the HME model density, it is crucial to propose a suitable framework for the path notation. As seen in the notation used for ME models in Section 2.2.2, let E denote an expert from the set of all experts in an HME model, $\mathcal{E}$. Additionally, let G denote a gate from the set of all gates in an HME model, $\mathcal{G}$. Every terminal expert node can be defined by the unique path going from the root node to itself. Let P_E denote the path corresponding to a particular expert E and let $\mathcal{P}$ denote the set of all paths in an HME model. Further, let (G, H) denote a segment of the path joining nodes G and H. In Figure 2.1 the collection of all paths, $\mathcal{P}$, is:

$$\begin{aligned}\mathcal{P} =&\{(P_{E1}, P_{E2}, P_{E3}, P_{E4}, P_{E5})\}\\ =&\{(G1, G2) \to (G2, E1),\\ &(G1, G3) \to (G3, E2),\\ &(G1, G3) \to (G3, E3),\\ &(G1, G2) \to (G2, G4) \to (G4, E4),\\ &(G1, G2) \to (G2, G4) \to (G4, E5)\}.\end{aligned}$$

Let $\mathcal{P}^{(G)}$ denote all paths from $\mathcal{P}$ passing through the gate G. For example, consider $\mathcal{P}^{(G3)}$ from Figure 2.1. In this case, all paths passing through the gate $G3$ are:

$$\begin{aligned}\mathcal{P}^{(G3)} = \{&(G1, G3) \to (G3, E2),\\ &(G1, G3) \to (G3, E3)\}.\end{aligned}$$

Further, let $\mathcal{P}_{>}^{(G)}$ denote all the paths descending from the gate G to the terminal nodes. Following the example,

$$\begin{aligned}\mathcal{P}_{>}^{(G3)} = \{&(G3, E2),\\ &(G3, E3)\}.\end{aligned}$$

It can be seen that $\mathcal{P}_{>}^{(G3)}$ is the *lower end* of the full paths $\mathcal{P}^{(G3)}$.

Lastly, let $r^{(H)}$ denote the number of descendants of node H. In the case of Figure 2.1, the tree is binary and thus $r^{(G)} = 2$ for all $G \in \mathcal{G}$. Since experts cannot have children by design, $r^{(E)} = 0$ for all $E \in \mathcal{E}$.

Using the notation proposed in this section, the HME model density function is outlined next.

2.3.3 HME Model Density Function

The density of the response y_i, given a vector of covariates, $\mathbf{x}_i$, from hierarchical mixture of experts model can be written down as follows

$$f(y_i|\mathbf{x}_i, \boldsymbol{\phi}) = \sum_{E \in \mathcal{E}} \pi_i^{(E)} f^{(E)} \left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)} \right), \tag{2.4}$$

for $i = 1, ..., n$, where $\boldsymbol{\phi}$ denotes a vector of all parameters in HME model while $\boldsymbol{\theta}^{(E)}$ denotes a vector of distinct parameters occurring in the expert E density, $f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right)$. The path probabilities $\pi_i^{(E)}$ can be written as

$$\pi_i^{(E)} = \prod_{(G,H) \in P_E} \pi_i^{(G,H)}, \tag{2.5}$$

where mixing proportions at each gate G are equal to

$$\pi_i^{(G,H)} = \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G,H)}\right)}{\sum_{H'} \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G,H')}\right)}, \tag{2.6}$$

for all H, children of gate G, with $\boldsymbol{\gamma}^{(G,H')} = \mathbf{0}$ for the first child H' in order to ensure identifiability and $\sum_H \pi_i^{(G,H)} = 1$.

The path probabilities $\pi_i^{(E)}$ are obtained as a product of mixing proportions from the unique path P_E, consisting of edges leading to E. Each gate node G in the model has its own gating parameters, $\boldsymbol{\gamma}^{(G,H)}$, controlling the mixing proportions at gate G. The evaluated quantities $\upsilon_i^{(E)} = \frac{\pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right)}{\sum_{E \in \mathcal{E}} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right)}$, for $i = 1, ...n$, across $E \in \mathcal{E}$ are also known as *responsibilities*.

Let us consider the five expert HME example once again. Figure 2.2 shows the mixing proportions, $\pi_i^{(G,H)}$, for all segments of paths joining nodes G and H.

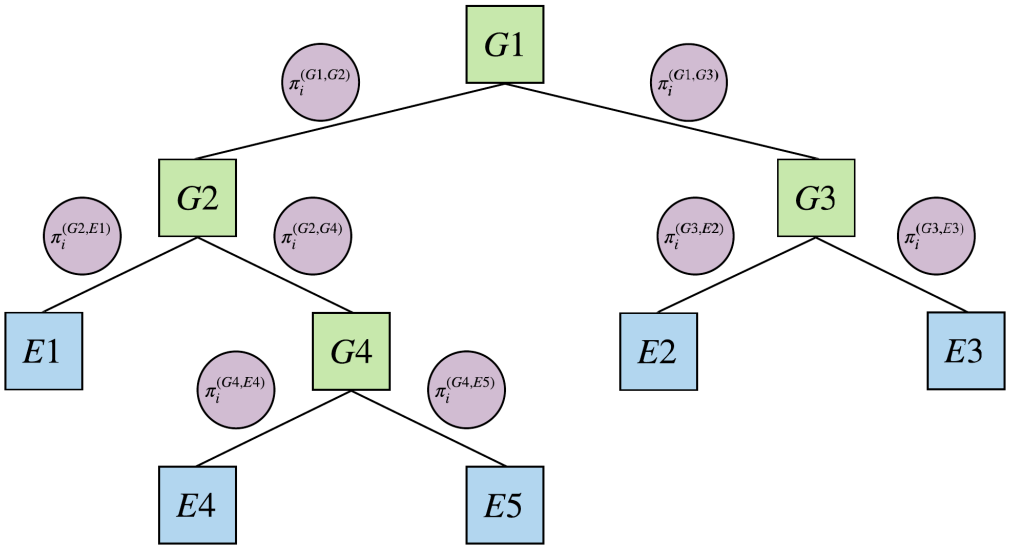


FIGURE 2.2: Illustration of an HME model with five experts and mixing proportions.

Expert $E2$ from Figure 2.2 is used to illustrate the process of obtaining the corresponding path probability $\pi_i^{(E2)}$. Using the previously introduced notation, the path leading to expert $E2$ can be written as

$$P_{E2} = (G1, G3) \rightarrow (G3, E2).$$

Following equation (2.5), the associated path probability is

$$\begin{aligned}
\pi_i^{(E2)} &= \pi_i^{(G1,G3)} \cdot \pi_i^{(G3,E2)} \\
&= \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G1,G3)}\right)}{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G1,G2)}\right) + \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G1,G3)}\right)} \cdot \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G3,E2)}\right)}{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G3,E2)}\right) + \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G3,E3)}\right)} \\
&= \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G1,G3)}\right)}{1 + \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G1,G3)}\right)} \cdot \frac{1}{1 + \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G3,E3)}\right)},
\end{aligned}$$

as $\boldsymbol{\gamma}^{(G1,G2)} = \boldsymbol{\gamma}^{(G3,E2)} = \mathbf{0}$, because $G2$ is the first child of $G1$ and $E2$ is the first child of $G3$.

This section concludes the chapter, where the HME model density function is defined and suitable notation is proposed. The next two chapters cover inference for gating and expert parameters in HME model. Two schools of thought are discussed in Chapters 3

and 4 offering some statistical background as well as outlining methods for parameter inference.

Chapter 3

Frequentist Inference for HME Models

This chapter introduces frequentist parameter inference methods for HME models. Firstly, some background on the frequentist approach is presented in Section 3.1. The HME likelihood function is then defined and discussed in Section 3.2. Following the discussion of the limitations posed by the use of HME likelihood function, latent assignment variables, which lead to the complete data likelihood function, are introduced and illustrated in Section 3.3. Section 3.4 discusses parameter inference using the expectation maximisation algorithm (Dempster et al., 1977). Finally, Section 3.5 summarises the suitability of the frequentist approach for inference in HME models.

3.1 Introduction to Frequentist Statistics

In any statistical modeling problem, it is of interest to estimate the values of unknown parameters in the model. Frequentist statistics is one of the two main schools of thought in the field of statistics, which draws conclusions about the outcome of an event by using the frequency or proportion of that outcome occurring across a large number of repetitions of the event. To illustrate the frequentist statistics approach, consider a vector of observed data $\mathbf{y} = (y_1, ..., y_n)$, which is believed to have a probability density function $f(\mathbf{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$ are the unknown parameters of interest. In contrast to Bayesian approach, discussed in detail in Chapter 4, frequentist approach treats unknown parameters $\boldsymbol{\theta}$ as fixed values with point estimates, $\hat{\boldsymbol{\theta}}$, obtained as a function of the sample of data, called *an estimator*. A common estimator used in frequentist statistics is the so-called *maximum likelihood estimator* (MLE), which aims to answer the question of

which parameter values are most likely given the observed data. In other words, what values of $\boldsymbol{\theta}$ yield the maximum value for the likelihood function defined as

$$L(\boldsymbol{\theta}|\mathbf{y}) = \prod_{i=1}^{n} L(\boldsymbol{\theta}|y_i) = \prod_{i=1}^{n} f(y_i|\boldsymbol{\theta}),$$

where y_i are assumed to be identically distributed and independent (iid) for $i = 1, ..., n$. The MLE can then be written as

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathbf{y}).$$

In practice, it is often more convenient to find the MLE which maximises a log-likelihood function defined as

$$l(\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^{n} l(\boldsymbol{\theta}|y_i) = \sum_{i=1}^{n} \log\left(f(y_i|\boldsymbol{\theta})\right),$$

which does not change the value of $\hat{\boldsymbol{\theta}}_{MLE}$ since the log is a monotonic function. The MLE estimator is consistent and asymptotically efficient, which means that the MLE has the smallest variance of all well-behaved estimators (see (Wasserman, 2010) for more details on estimator properties).

In frequentist statistics, the uncertainty around parameter estimates is specified by *confidence intervals*. For a particular probability of a, the confidence intervals are defined such that if the data were repeatedly sampled, resulting in a confidence interval each time, then a of these intervals would contain the *true* value of the unknown fixed parameters. It is important to highlight, that unlike the intervals produced in Bayesian setting, the confidence intervals do not mean that the unobserved true value of the parameter falls into the interval with a probability of a.

We proceed by defining HME likelihood and log-likelihood functions and discussing the arising parameter estimation problems and procedures.

3.2 HME Likelihood Function

Using the HME density function defined as per (2.4), the HME likelihood function can be written as

$$L(\boldsymbol{\phi}|\mathbf{y}) = \prod_{i=1}^{n} f(y_i|\mathbf{x}_i, \boldsymbol{\phi}) = \prod_{i=1}^{n} \left[\sum_{E \in \mathcal{E}} \pi_i^{(E)} f^{(E)} \left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)} \right) \right], \tag{3.1}$$

for the i-th observation y_i with the corresponding vector of covariates, $\mathbf{x}_i$, for $i = 1, ..., n$, where $\boldsymbol{\phi}$ denotes a vector of all parameters in the HME model while $\boldsymbol{\theta}^{(E)}$ denotes a vector of distinct parameters occurring in the expert E density, $f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right)$. The log-likelihood function is then

$$\begin{aligned} l(\boldsymbol{\phi}|\mathbf{y}) = \sum_{i=1}^{n} \log\left(f(y_i|\mathbf{x}_i, \boldsymbol{\phi})\right) &= \sum_{i=1}^{n} \log\left(\sum_{E \in \mathcal{E}} \pi_i^{(E)} f^{(E)} \left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)} \right) \right) \\ &= \sum_{i=1}^{n} \log\left(\sum_{E \in \mathcal{E}} \left(\prod_{(G,H) \in P_E} \pi_i^{(G,H)} \right) f^{(E)} \left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)} \right) \right). \end{aligned} \tag{3.2}$$

It is evident that the maximum likelihood estimates of the parameters cannot be found in closed-form. Thus an iterative procedure must be used. One choice would be to directly optimise the maximum likelihood function numerically. An alternative choice, which simplifies the resulting optimisation problem considerably is the expectation maximisation algorithm (Dempster et al., 1977), discussed in Section 3.4.

3.3 HME Complete Data Likelihood Function

The latent variables are variables that are not directly observed but inferred through other directly observed variables in the model. The earliest example of such variables being used is in the field of psychology, where Spearman (1904) investigates objectively measuring and determining general intelligence. In HME models, the latent variables assign observations to unobserved groups defined by the experts in the model. The latent assignment variables, also called *allocation* or *indicator* variables within this thesis, are defined as follows

$$z_i^{(G,H)} = \begin{cases} 1 & \text{if } (G,H) \in P_{E(i)} \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

and

$$z_i^{(H)} = \begin{cases} 1 & \text{if } z_i^{(G,H)} = 1 \text{ for some } (G,H) \in P_{E(i)} \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where $E(i)$ denotes the expert grouping of the i-th observation. To describe such unobserved grouping, we write *observation i is assigned/allocated to expert $E(i)$*. The variables $z_i^{(G,H)}$ and $z_i^{(H)}$, introduced in (3.3) and (3.4) respectively, simply denote whether the i-th point would need to travel through the corresponding segment (G,H) or the node H of the tree to reach expert $E(i)$.

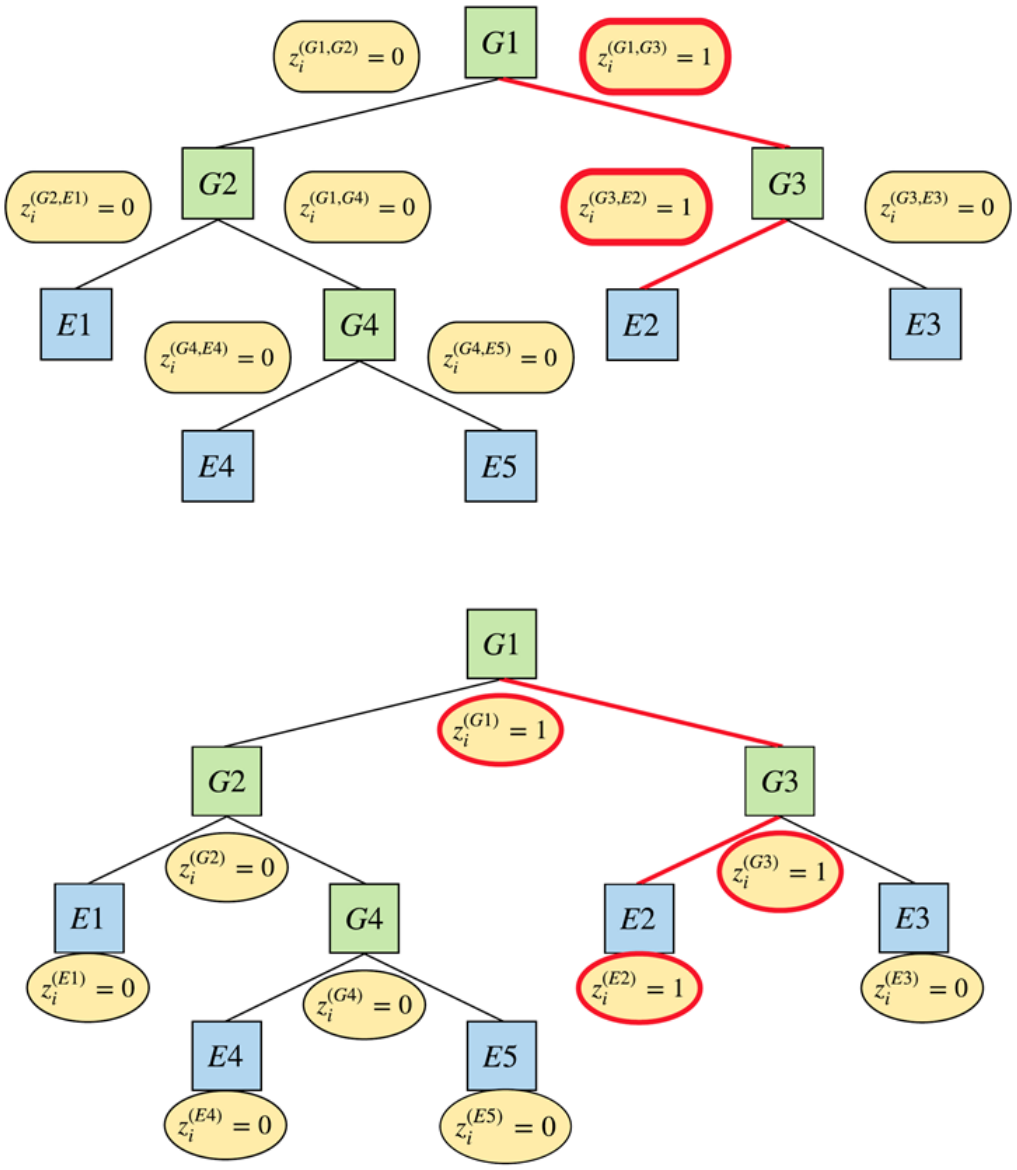


FIGURE 3.1: Illustration of the indicator variables $z_i^{(G,H)}$ and $z_i^{(H)}$ for the i-th point, which is assigned to the expert $E2$, i.e. $E(i) = E2$.

For example, let us consider the i-th observation, that has been assigned to expert $E2$ (Figure 3.1). Recall that the corresponding path can be written as

$$P_{E(i)} = P_{E2} = (G1, G3) \rightarrow (G3, E2),$$

which corresponds to the path highlighted by a red bold line. It is evident that the i-th point considered in this example would have to travel through segments $(G1, G3)$ and $(G3, E2)$ of the tree to reach $E2$. Thus we have

$$z_i^{(G,H)} = \begin{cases} 1 & \text{for } (G1, G3) \text{ and } (G3, E2) \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, the i-th point would only pass through nodes $G1$, $G3$, and $E2$, which means that

$$z_i^{(H)} = \begin{cases} 1 & \text{for } G1, G3 \text{ and } E2 \\ 0 & \text{otherwise.} \end{cases}$$

The introduction of the latent assignment variables allows for the grouping of all observations into $|\mathcal{E}|$ unobserved groups. Hence, the HME model probability density function seen in (2.4) can be re-written as follows

$$f(y_i, \mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\phi}) = \begin{cases} \pi_i^{(E(i))} f^{(E(i))}\left(y_i | \mathbf{x}_i, \boldsymbol{\theta}^{(E(i))}\right) & \text{for } E(i) \in \mathcal{E} \text{ such that } z_i^{(E(i))} = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{3.5}$$

for the i-th observation with the corresponding vector of covariates, $\mathbf{x}_i$, for $i = 1, ..., n$, where $\mathbf{z}_i = \left(z_i^{(E)}\right)_{E \in \mathcal{E}}^T$. The likelihood function, now called the *complete data likelihood function* and denoted $L^c(\cdot)$, can then be written as:

$$
\begin{aligned}
L^c(\boldsymbol{\phi}|\mathbf{y},\mathbf{z}) &= \prod_{i=1}^{n} f(y_i,\mathbf{z}_i|\mathbf{x}_i,\boldsymbol{\phi}) \\
&= \prod_{i=1}^{n}\prod_{E\in\mathcal{E}} \left[\pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right)\right]^{z_i^{(E)}},
\end{aligned} \tag{3.6}
$$

for the i-th observation with the corresponding vector of covariates, $\mathbf{x}_i$, for $i = 1, ..., n$, where $\boldsymbol{\phi}$ denotes a vector of all parameters in HME model while $\boldsymbol{\theta}^{(E)}$ denotes a vector of distinct parameters occurring in the expert E density, $f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right)$. The complete data log-likelihood function is then:

$$
\begin{aligned}
l^c(\boldsymbol{\phi}|\mathbf{y},\mathbf{z}) &= \sum_{i=1}^{n}\sum_{E\in\mathcal{E}} z_i^{(E)} \log\left[\pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right)\right] \\
&= \sum_{i=1}^{n}\sum_{E\in\mathcal{E}} z_i^{(E)} \log\left[\left(\prod_{(G,H)\in P_E} \pi_i^{(G,H)}\right) f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E(i))}\right)\right].
\end{aligned} \tag{3.7}
$$

In the above equation, $z_i^{(E)}$ is equal to 1 only for one $E \in \mathcal{E}$ hence simplifying the log-likelihood evaluation process. The resultant advantage of using the complete data log-likelihood is that the expert parameters can be sampled explicitly.

Lastly, having latent assignment variables present in the model lets us write down the density function for the gating parameters. Let $\boldsymbol{\gamma}^{(G)} = \left(\boldsymbol{\gamma}^{(G,H)}\right)_H^T$ denote a collection of all gating parameters for gate G and let $\mathbf{z}_i^{(G)} = \left(z_i^{(G,H)}\right)_H$ and $\mathbf{z}^{(G)} = \left(\mathbf{z}_1^{(G)}, ..., \mathbf{z}_n^{(G)}\right)^T$. The conditional probability density function of $\boldsymbol{\gamma}^{(G)}$ is then

$$
f\left(\boldsymbol{\gamma}^{(G)}|\mathbf{z}^{(G)}\right) \propto f\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) f\left(\boldsymbol{\gamma}^{(G)}\right), \tag{3.8}
$$

where

$$
f\left(\mathbf{z}_i^{(G)}|\boldsymbol{\gamma}^{(G)}\right) = \prod_{H}\left(\pi_i^{(G,H)}\right)^{z_i^{(G,H)}}, \tag{3.9}
$$

which is simply a probability density function of the multinomial distribution, where $\sum_H \pi_i^{(G,H)} = 1$ and $z_i^{(G,H)} = 1$ for only one child H of G. The corresponding indicator likelihood, which is equivalent to the joint density of the allocation variables $\mathbf{z}^{(G)}$ given the parameters $\boldsymbol{\gamma}^{(G)}$, and log-likelihood functions are then

$$L\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) = f\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) = \prod_{i=1}^{n}\prod_{H}\left(\pi_i^{(G,H)}\right)^{z_i^{(G,H)}}, \tag{3.10}$$

and

$$l\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) = \sum_{i=1}^{n}\sum_{H}\log\left(\left(\pi_i^{(G,H)}\right)^{z_i^{(G,H)}}\right). \tag{3.11}$$

Having written down the log-likelihood functions for the parameters of HME models, the process of the maximum likelihood estimation is discussed next.

3.4 The Expectation Maximisation Algorithm

As seen in Section 3.2, for complex models, such as HME, the likelihood function can get quite complicated resulting in difficult optimisation problems. In some situations, like in the case of HME models, introducing a latent variable can significantly simply the resultant optimisation problem. In such case, the expectation maximisation algorithm (EM) can be used. The EM algorithm was first introduced by Dempster et al. (1977) and has since been used in machine learning and data mining applications. The idea behind the EM algorithm consists of iterating two steps, called the expectation (*E-step*) and the maximisation (*M-step*). Jordan and Jacobs (1994) outlines the two steps of the EM algorithm for the HME models as follows.

Given the initial parameter values $\boldsymbol{\phi}^{(0)}$ repeat until convergence:

1. E-Step. Obtain the deterministic function Q:

$$Q(\boldsymbol{\phi}, \boldsymbol{\phi}^{(t)}) = \mathbb{E}\left[l^c(\boldsymbol{\phi}|\mathbf{y}, \mathbf{z})\right],$$

   where $\boldsymbol{\phi}^{(t)}$ is the value of the expert parameters at the t-th iteration, the expectation is taken with respect to $\boldsymbol{\phi}^{(t)}$, and $l^c(\cdot)$ is the complete data likelihood as per (3.7).

2. M-Step. Maximise function Q with respect to $\boldsymbol{\phi}$ to find the new parameter estimates $\boldsymbol{\phi}^{(t+1)}$:

$$\boldsymbol{\phi}^{(t+1)} = \arg\max_{\boldsymbol{\phi}} Q\left(\boldsymbol{\phi}, \boldsymbol{\phi}^{(t)}\right).$$

The EM algorithm is a powerful method, however, it does have its faults. The EM algorithm has been known to suffer from slow convergence, especially in Gaussian mixtures

(Park and Ozeki, 2009). It is also dependent on the choice of stopping criterion as well as the initial parameter values used (Karlis and Xekalaki, 2003). When automatically growing or pruning HME models, the number of parameters is constantly changing making the arbitrary choice of the initial parameter values even more challenging. Lastly, the complexity of the HME model architecture usually results in many local maxima thus requiring multiple starting points when maximising the complete likelihood function therefore further decreasing the efficiency of the parameter estimation process (Huerta et al., 2003).

3.5 Final Remarks

This chapter introduces the statistical background for frequentist parameter estimation based on maximum likelihood. While it is important to understand and review the frequentist inference methods for HME, certain disadvantages of the approach make the opposing Bayesian school of thought a more suitable candidate for HME model parameter inference. It has been previously mentioned that the iterative maximum likelihood estimation methods tend to be slow to converge and depend on initial values as well as the stopping criterion. This issue can be especially difficult in complex models such as HME, because there is a large number of model parameters present involved in defining the expert and gating distributions (Bishop and Svenskn, 2002). The problem is then further complicated by the introduction of automatic architecture selection, tackled in Chapter 6, which results in an everchanging parameter dimensionality. In the latter case, the frequentist approach does not offer a natural method for selecting the resulting tree architectures. Thus in the next chapter, the opposing Bayesian inference approach is presented and applied to all methods within this thesis.

Chapter 4

Bayesian Inference for HME Models

This chapter spans several topics in Bayesian statistics starting with an introduction in Section 4.1, which covers the main idea behind the approach and highlights the differences when compared to the frequentist approach seen in Chapter 3. Following the introduction of prior and posterior distributions, the former are discussed in more detail in Section 4.2. Next, sampling and updating of the model parameters are covered in Section 4.3. In particular, the Metropolis-Hastings algorithm (Hastings, 1970; Chib and Greenberg, 1995) and the Gibbs sampler (Gelfand, 2000; Geman and Geman, 1984) sampling algorithms are presented and discussed in the context of HME model parameters. The steps for updating gating parameters and allocation variables in HME models are outlined in Section 4.3.3, where a general framework for updating expert parameters is discussed. The latter is then refined in Chapter 5 for a special case of normal experts, which is the main focus of this thesis. Finally, the concepts of convergence and mixing for HME models are discussed in Sections 4.3.4 and 4.3.5.

4.1 Introducton to Bayesian Statistics

Bayesian statistics is founded on the basis of Bayes' theorem, developed by Thomas Bayes, which describes the probability of an event that is conditioned on some prior knowledge (Bayes and Price, 1763). According to Bayes theorem, the probability of event A occurring given that the event B has occurred is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

where $P(A)$ and $P(B)$ are, respectively, the probabilities of events A and B occurring while $P(B|A)$ is the probability of event B occurring, given event A has occurred.

In Bayesian Statistics, the estimation of unknown model parameters is based on Bayes' theorem, which implies that estimates should change given relevant information. Unlike the frequentist approach, which aims to provide point estimates for unknown parameters, the goal of Bayesian statistics is to obtain their probability density functions. The corresponding parameter density functions then lead to point estimation of unknown parameters as well as the high probability regions around them. The latter means that unknown parameters are treated as random variables as opposed to fixed values as seen in frequentist statistics.

In order to present the adaptation of Bayes theorem used in Bayesian statistics, consider a vector of observed data $\mathbf{y} = (y_1, ..., y_n)$, which is believed to have a probability density function $f(\mathbf{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$ is a vector of the associated unknown parameters one wants to estimate. The prior beliefs about unknown parameters can be expressed via a joint probability density function called *the prior distribution* and denoted $f(\boldsymbol{\theta})$ (discussed in detail for HME in Section 4.2). The prior distribution does not take into account any evidence from the observed data $\mathbf{y}$ and thus the density function for $\boldsymbol{\theta}$ given $\mathbf{y}$ is updated as follows

$$f(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\mathbf{y})},$$

where $f(\mathbf{y}|\boldsymbol{\theta})$ is the data likelihood, $f(\boldsymbol{\theta}|\mathbf{y})$ is the joint parameter density function conditioned on the observed data $\mathbf{y}$, called *the posterior distribution*, and $f(\mathbf{y})$ is the marginal distribution of the observed data, $\mathbf{y}$.

If the posterior distribution, $f(\boldsymbol{\theta}|\mathbf{y})$, is in the same probability distribution family as the prior distribution, $f(\boldsymbol{\theta})$, both are then called *conjugate distributions* and the prior distribution is called a *conjugate prior*. Conjugate priors are a popular choice because they allow for a derivation of a closed-form expression for the posterior distribution, which can then be used to sample from the posterior parameters. In the case of non-conjugate priors, the computation required is usually more complex and hence requires a different approach. In most cases, a numerical simulation is performed by drawing a sample of parameter values from an approximation of the posterior distribution.

It is important to note that $f(\mathbf{y})$ does not depend on $\boldsymbol{\theta}$ and thus the posterior distribution can be expressed up to a constant of proportionality as

$$f(\boldsymbol{\theta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta}),$$

which does not require estimating $f(\mathbf{y})$. One can say that the posterior distribution of unknown model parameters is proportional to the product of data likelihood and the joint prior density function of those parameters.

The point estimates of the parameters $\boldsymbol{\theta}$, denoted as $\hat{\boldsymbol{\theta}}$, can then be obtained from the posterior distribution by calculating the posterior mean, median or mode. In contrast to the frequentist approach, the *credible intervals* used in Bayesian statistics provide a probabilistic interpretation of the uncertainty around the parameter estimates. That is, the credible interval is an interval in which an unobserved parameter value falls with a particular probability.

The debate on frequentist versus Bayesian approaches dates back a few centuries and is well reflected by a quote from Kendall (1949), which says "Few branches of scientific method have been subject to so much difference of opinion as the theory of probability". The choice between frequentist and Bayesian inference can also be viewed as a choice between relative frequencies and degrees of belief (Vallverdu, 2011). In frequentist statistics, a probability is interpreted as a relative frequency of an outcome of an event occurring over a large number of independent repetitions of the event under roughly the same conditions (Bickel and Lehmann, 2012). On the other hand, under Bayesian framework, a probability is considered to be the plausibility, representing a prior belief, of an outcome of the event occurring, which can be updated in the light of the evidence. In this thesis, a choice in favour of the degrees of belief is made. The parameter estimation approaches developed within this thesis are thus outlined using Bayesian approach. Prior distributions required for Bayesian parameter estimation in the context of HME models are discussed next.

4.2 Parameter Priors for HME Models

As discussed in the previous section, Bayesian approach for inference requires specifying prior distributions for the parameters in the model. A common choice for prior distributions of continuous parameters is the multivariate normal distribution, which benefits from the specification of prior covariances between the parameters as well as individual prior parameter means and variances (Waterhouse et al., 1996). Hence, Gaussian priors for the gating and expert parameters in HME models are outlined next.

The prior distribution for gating parameters can be written as

$$\boldsymbol{\gamma}^{(G,H)} \sim \mathsf{MVN}\left(\boldsymbol{\mu}_{\boldsymbol{\gamma}}^{(G,H)}, \Sigma_{\boldsymbol{\gamma}}^{(G,H)}\right).$$

From the density of the gating parameters shown in (3.8), it can be seen that the density is not Gaussian and thus the proposed prior is not a conjugate prior. Methods for sampling parameters in such cases are discussed in Section 4.3. Similarly to gating parameters, the prior distribution for expert parameters can be written as

$$\boldsymbol{\theta}^{(E)} \sim \mathsf{MVN}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}^{(E)}, \Sigma_{\boldsymbol{\theta}}^{(E)}\right).$$

In the special case of normal experts, discussed in Chapter 5, the proposed prior distribution becomes a conjugate prior thus greatly simplifying expert parameter inference.

Even though Gaussian priors are selected for the applications considered in this thesis, more complex choices of prior distributions have been investigated in the relevant literature. Examples of such include conditioning the prior variance for the gating parameter on the inverse of the mean parameter of its descending expert's distribution (Bishop and Svenskn, 2002). In addition to Gaussian priors, Gamma priors are discussed by Waterhouse et al. (1996). Lastly, Diebolt and Robert (1994) investigates using non-informative approximations associated with improper priors. Next, the standard sampling techniques for Bayesian parameter inference are outlined.

4.3 Markov Chain Monte Carlo Methods

Markov Chain Monte Carlo (MCMC) methods are a group of sampling techniques that are often used for estimating posterior distributions in Bayesian inference, where the closed-form expression cannot be obtained (van Ravenzwaaij et al. (2018), Metropolis et al. (1953)). An alternative method for Bayesian inference, called variational inference, is discussed for HME models in Bishop and Svenskn (2002). In addition to approximating the target posterior distribution, the aim of the approach is to derive a lower bound for the likelihood of the observed data, which is often used to perform model selection. Given the requirement to pre-select a group of models to be considered, variational inference, however, is not suitable for problems of changing dimensionality (discussed in Chapter 6), and thus MCMC methods are used throughout this thesis.

MCMC methods rely on drawing samples where the current value is dependent on the value that was generated before it. Such draws then form a chain, called a *Markov chain*. Beneficially for HME models, the dependability on the previous value allows MCMC algorithms to narrow in on the distribution that is being sampled even in cases of high parameter dimensionality (Brownlee, 2019). The resulting Markov chain should

then effectively sample from the desired target posterior after it reaches equilibrium (assessing when the equilibrium is reached is discussed in Section 4.3.4).

The two most commonly used MCMC algorithms are the Metropolis-Hastings (MH) algorithm (Hastings, 1970) and the Gibbs sampler (Geman and Geman, 1984). For HME models, we use and evaluate a combination of both algorithms to estimate the model parameters. The simplest method of generating a Markov chain, the MH algorithm, is discussed next.

4.3.1 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm requires an additional distribution, called the *proposal* distribution, and generates a proposed new value of the model parameter using the previously specified proposal distribution. The likelihood function is then evaluated at both the current and the proposed value of the parameter. The obtained values are compared in order to decide if the proposed new value should be accepted. More formally, the steps of the MH algorithm are as follows. Let us partition the parameter vector $\boldsymbol{\theta}$ into b blocks, i.e. $\boldsymbol{\theta} = (\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_b)$.

1. Set initial starting values for the parameter vector, i.e., $\boldsymbol{\theta}^{(0)} = \left(\boldsymbol{\eta}_1^{(0)}, ..., \boldsymbol{\eta}_b^{(0)}\right)$.
2. For $t = 1, ..., T$ and for $k = 1, ..., b$:
   (a) Generate a proposed value $\boldsymbol{\eta}_k^*$ from the proposal distribution $g\left(\boldsymbol{\eta}_k^* | \boldsymbol{\eta}_k^{(t-1)}\right)$.
   (b) Accept $\boldsymbol{\eta}_k^*$ with probability

$$\alpha = \min\left\{1, \frac{f\left(\boldsymbol{\eta}_k^* | \mathbf{y}\right) g\left(\boldsymbol{\eta}_k^{(t-1)} | \boldsymbol{\eta}_k^*\right)}{f\left(\boldsymbol{\eta}_k^{(t-1)} | \mathbf{y}\right) g\left(\boldsymbol{\eta}_k^* | \boldsymbol{\eta}_k^{(t-1)}\right)}\right\}.$$

   (c) If the proposal is accepted, set $\boldsymbol{\eta}_k^{(t)} = \boldsymbol{\eta}_k^*$. Otherwise, set $\boldsymbol{\eta}_k^{(t)} = \boldsymbol{\eta}_k^{(t-1)}$.

A special case of the algorithm above, called the Metropolis algorithm, arises when the proposal distribution is symmetric, i.e., $g\left(\boldsymbol{\eta}_k^{(t-1)} | \boldsymbol{\eta}_k^*\right) = g\left(\boldsymbol{\eta}_k^* | \boldsymbol{\eta}_k^{(t-1)}\right)$ (Metropolis et al., 1953). The latter simplifies the acceptance probability α to the form

$$\alpha = \min\left\{1, \frac{f\left(\boldsymbol{\eta}_k^* | \mathbf{y}\right)}{f\left(\boldsymbol{\eta}_k^{(t-1)} | \mathbf{y}\right)}\right\}.$$

It is a common practice to record whether each proposal for a block k of parameters has been accepted, resulting in a binary vector $\mathbf{c}_k = (c_{k1}, ..., c_{kT})$, where each

$$c_{kt} = \begin{cases} 1 & \text{if } t\text{-th proposal is accepted} \\ 0 & \text{otherwise,} \end{cases}$$

for $t = 1, ..., T$. The acceptance rate is then equivalent to the mean of $\mathbf{c}_k$, i.e. $\bar{\mathbf{c}}_k = \frac{\sum_{m=1}^{T} c_{km}}{T}$. A high acceptance rate could indicate that the new proposed values are very close to the current ones. Thus, one may consider increasing the variance of the proposal distribution to encourage wider exploration of the space. On the other hand, a low acceprance rate means that there are many rejections and thus computation time is being wasted. In such case, decreasing the variance of the proposal distribution might help, however, in some cases, the suitability of the proposal distribution might need to be reassessed.

Next, the method of Gibbs sampling, which is used when the full conditional posterior distribution of the parameter block is known, is discussed.

4.3.2 The Gibbs Sampling Algorithm

The Gibbs algorithm is used to sample from the posterior distribution of the parameter block, where the full conditional posterior distribution is known (Geman and Geman, 1984). Such cases tend to happen when a conjugate prior is used and hence the full conditional posterior distribution can be easily obtained. As before, let us partition the parameter vector $\boldsymbol{\theta}$ into b blocks, i.e. $\boldsymbol{\theta} = (\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_b)$. The Gibbs sampling algorithm can then be written as follows

1. Set initial starting values for the parameter vector, i.e., $\boldsymbol{\theta}^{(0)} = \left(\boldsymbol{\eta}_1^{(0)}, ..., \boldsymbol{\eta}_b^{(0)}\right)$.
2. For $t = 1, ..., T$ and for $k = 1, ..., b$ draw a sample $\boldsymbol{\eta}_k^{(t)}$ from the conditional distribution $f\left(\boldsymbol{\eta}_k^{(t)} | \boldsymbol{\eta}_1^{(t)}, ..., \boldsymbol{\eta}_{k-1}^{(t)}, \boldsymbol{\eta}_{k+1}^{(t-1)}, ..., \boldsymbol{\eta}_b^{(t-1)}, \mathbf{y}\right)$.

In the above, each block of the model parameters is updated in turn with the acceptance probability of 1. For HME model parameters, a mixture of both Metropolis-Hastings and Gibbs sampling algorithms is used and discussed next.

4.3.3 Updating HME Model Parameters

This section covers the general updates of the HME model parameters for a set architecture.

4.3.3.1 Allocation Variables Updates

In this section, we propose a way of updating the latent allocation variables as if trying to answer the question of how likely an observation is to have come from a particular expert density in the model. Recall that the latent allocation variables $\mathbf{z}^{(G)} = \left(z_1^{(G)}, ..., z_n^{(G)}\right)^T$ indicate if the i-th point has passed through the gate G. Let $\mathbf{z}_{z_i=1}^{(G)}$ denote a subset of size n_G containing points that have reached gate G. In addition to gating and expert parameter updates, this subset can then be updated as

1. Determine all terminal expert nodes descending from the gate G and denote the set $\mathcal{E}'$.
2. For each point $j = 1, ..., n_G$ in the set $\mathbf{z}_{z_i=1}^{(G)}$:
   (a) Calculate
   $$\alpha_j^{(E)} = \left(\prod_{(G,H)\in P_>^{(G)}} \pi_j^{(G,H)} \right) f^{(E)}\left(y_j|\mathbf{x}_j, \boldsymbol{\theta}^{(E)}\right),$$
   for all $E \in \mathcal{E}'$.
   (b) Assign the j-th point to expert E, i.e., set $z_j^{(E)} = 1$ and $z_j^{(E'')} = 0$ for $E'' \neq E$ with probability $\alpha_j^{(E)}$, where $\sum_{E\in\mathcal{E}'} \alpha_j^{(E)} = 1$.

The above algorithm calculates the probabilities of an observation *belonging* to each of the experts at their current state. These probabilities are then used to reassign observations. Next, the updates of gating and expert parameters are considered.

4.3.3.2 Gating and Expert Parameters Updates

In HME models, gating parameters at gate G, i.e. $\boldsymbol{\gamma}^{(G)} = \left(\boldsymbol{\gamma}^{(G,H)}\right)_H^T$ and expert parameters $\boldsymbol{\theta}^{(E)}$ can be updated using the Metropolis-Hastings algorithm outlined in Section 4.3.1. As noted before, the MH algorithm requires one to specify the proposal distribution. In this section, a standard proposal distribution is presented followed by a proposed alternative, which is tailored to suit the complexity of HME models.

Standard MH Proposal Approach

For a generic continuous parameter vector $\boldsymbol{\eta}$, the following multivariate normal distribution is often used as a proposal distribution:

$$\boldsymbol{\eta}^* \sim \mathsf{MVN}\left(\boldsymbol{\eta}, \Sigma_{\boldsymbol{\eta}^*}\right),$$

where $\boldsymbol{\eta}^*$ is the new proposed parameter value, $\boldsymbol{\eta}$ is the current parameter value, and $\Sigma_{\boldsymbol{\eta}^*}$ is the variance-covariance matrix of the proposal distribution.

For this approach, one is required to select the variance of the proposal distribution. The larger the variance, the wider exploration of the space. Thus, there is a balance to be had between generating proposals that are too close to the previous parameter value and proposals that venture too far out thus wasting computation time. This approach works well for ME cases, such as those tackled in Chapter 5, however, it fails to address the following issues specific to HME.

Selecting the variance of proposal distribution is a non-trivial task for models, where multiple model parameters are updated using the same proposal. For example, an HME model with multiple gate nodes will have multiple gating parameters that require updating. The proposal distribution for each of those gating parameters should thus take full advantage of the information provided by the observed data, which is relevant to the specific node. The latter requirement is not met by an arbitrary choice for the variance of the proposal distribution, which is applied across all gating parameter updates.

Also, adding and removing nodes within the tree model architecture (discussed in Chapter 6) results in an everchanging parameter dimensionality. This means that, immediately after the parameter dimensionality changes, the previous parameter values do not always provide the most reliable centering point for the proposal distribution. For example, an addition of a new gate node, G', will require setting intial gating parameters $\boldsymbol{\gamma}^{(G')}$, which will then serve as a mean vector for the proposal distribution of the new parameter value, i.e., $\boldsymbol{\gamma}^{(G')^*} \sim MVN(\boldsymbol{\gamma}^{(G')}, \Sigma_{\boldsymbol{\gamma}^*})$. In the situation where the value of $\boldsymbol{\gamma}^{(G')}$ is not well chosen the proposed update $\boldsymbol{\gamma}^{(G')^*}$ is less likely to be accepted thus causing the chain to move slower. A new approach for constructing expert and gating parameter proposals, which addresses these shortcomings, is proposed next.

Modified MH Proposal Approach for HME Models

Consider the following proposal distribution

$$\boldsymbol{\eta}^* \sim \mathsf{MVN}\left(\hat{\boldsymbol{\eta}}, \Sigma\right),$$

where $\hat{\boldsymbol{\eta}}$ is the Iteratively Weighted Least Squares (IWLS) estimate of the parameter of interest (see Appendix A for details on the IWLS algorithm), and Σ is the corresponding variance-covariance matrix obtained during the estimation process (shown in Appendix A.1).

By centering the proposal distribution around the parameter estimate, one is targeting the high density areas of the parameter space, which is likely to improve the proposals and hence acceptance rates. This approach is particularly useful when encountering the previously mentioned issues caused by changing parameter dimensionality. Such cases are discussed in Chapter 6, where the tailored proposal distribution is used to update the gating parameters of HME models (details outlined in Appendix B).

Another advantage of performing IWLS estimation is the ability to extract the proposal variance-covariance matrix during the estimation process. The latter is performed with the help of a matrix decomposition. Unlike the standard approach, the variance-covariance matrix for the proposal distribution thus follows the direction of the target distribution and guides the width of the proposal. For HME models, this means that the proposal distribution will adapt to each node and reflect the relevant information provided by the observed data.

Difference in Expert and Gate Nodes Updates

It is known that the density of gating parameters in HME model is not Gaussian, unlike their set prior distribution. Thus the Metropolis-Hastings algorithm, as outlined in this section, is used to update the gating parameters in the model. In this thesis, evaluations performed on simple ME cases use the standard proposal distribution. On the other hand, more complex HME models with changing parameter dimensionality take advantage of the tailored proposal distribution for the gating parameters.

In this section, a general framework for updating any expert parameters using the MH algorithm is presented. In practice, the method used for updating expert parameters depends on the corresponding expert density functions. For example, a conjugate prior for expert parameters can be used when considering Gaussian experts. In such case, the full conditional posterior distribution is known and hence the Gibbs sampling algorithm can be used instead. The special case of Gaussian experts and their parameter sampling techniques are covered in Chapter 5.

Next, the concept of MCMC chain convergence is discussed in more detail.

4.3.4 Convergence for HME Models

It has been previously mentioned that the chain resulting from the MCMC draws, after it reaches equilibrium, should be effectively sampling from the desired target posterior distribution. An MCMC chain is said to have reached its equilibrium, or *converged*, if the chain has reached a stationary distribution (Toft et al., 2007). In simple terms, a stationary distribution is a probability distribution that remains unchanged in the chain

as the number of iterations increases. In this section, both visual and formal assessment of convergence is discussed as well as a method for interrogating the convergence of the overall model is proposed.

4.3.4.1 Visual Evaluation

It is important to note that convergence to the stationary distribution does not occur instantly. A set of samples, generated before convergence has been achieved, is thus discarded and referred to as the *burn-in* period. One can visually assess the convergence of a single model parameter by looking at the so-called *trace plots*, which show the history of parameter values across the MCMC iterations. In the trace plots, signs of non-convergence include the chain staying in the same state for too long (flat areas) and evidence of too many consecutive steps in one direction. The desired trace plot of a chain that has achieved convergence is often said to resemble a *hairy caterpillar* such as shown in Figure 4.1. This approach works well in cases where a small number of low dimensional model parameters is sampled, however, investigating trace plots becomes rather tedious as the number and/or the dimensionality of model parameters increase.

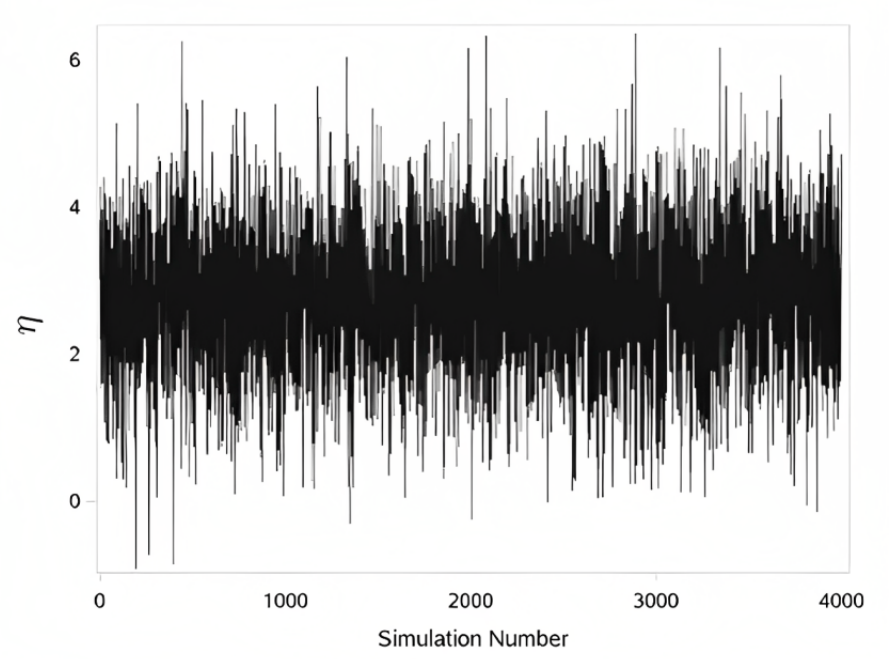


FIGURE 4.1: An example of a desired trace plot for some parameter η provided by SAS Help Center (2019).

4.3.4.2 Alternative Method of Convergence Assessment for HME Models

It is clear that, although effective, evaluating individual parameter convergence visually becomes rather time-consuming as the number of parameters sampled increases. Since HME models are based on a tree architecture, they often have a large number of model parameters, which increase in dimensionality as the number of explanatory variables increases. The latter makes assessing the individual parameter convergence rather challenging. The method proposed in this section improves the ease of the convergence assessment process for pre-set tree architectures, however more crucially, it offers a framework for assessing convergence in problems of changing parameter dimensionality, where tracking the individual parameter values is not possible in the first place.

The idea behind the proposed method involves assessing an overall HME model convergence by interrogating the convergence of predictions, obtained by using the posterior model parameters at each MCMC iteration. More formally, prediction for the i-th point at iteration t , $\hat{y}_i^{(t)}$, can be calculated as

$$\begin{aligned}\hat{y}_i^{(t)} &= \sum_{E\in\mathcal{E}} \pi_i^{(E)(t)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)(t)}\right) \\ &= \sum_{E\in\mathcal{E}} \prod_{(G,H)\in P_E} \pi_i^{(G,H)(t)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)(t)}\right),\end{aligned} \tag{4.1}$$

where mixing proportions at each gate G are equal to

$$\pi_i^{(G,H)(t)} = \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G,H)(t)}\right)}{\sum_{H'} \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G,H')(t)}\right)},$$

for $(G,H) \in P_E$ and or $E \in \mathcal{E}$. In (4.1) the predictions depend on the expert parameter values, $\boldsymbol{\theta}^{(E)(t)}$, and the path probabilities, $\pi_i^{(E)(t)}$, which are obtained using the gating parameter values, $\boldsymbol{\gamma}^{(G)(t)} = \left(\boldsymbol{\gamma}^{(G,H)(t)}\right)_H^T$, at iteration t for $i = 1, ..., n$ with $(G,H) \in P_E$ and $E \in \mathcal{E}$. It is thus proposed to assess the convergence of $\hat{\mathbf{y}}^{(t)} = \left(\hat{y}_1^{(t)}, ..., \hat{y}_n^{(t)}\right)^T$ across $t = 1, ..., T$ instead of examining the convergence of the gating and expert parameters individually.

Having obtained $\hat{\mathbf{y}} = \left(\hat{\mathbf{y}}^{(1)}, ..., \hat{\mathbf{y}}^{(T)}\right)$ one can visually assess the convergence of predictions. In two-dimensional cases, this can be done by plotting the fitted lines for all iterations and observing when and if they start looking consistent. To demonstrate, consider Figure 4.2, which has been created for illustration purposes only.

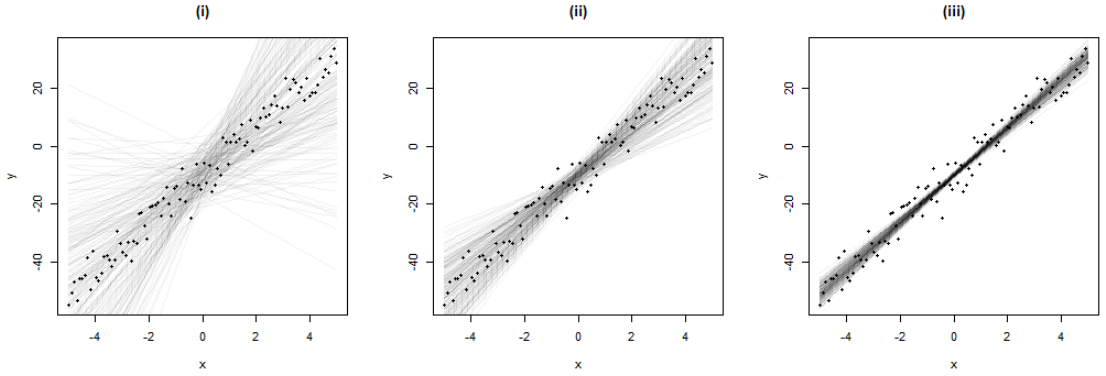


FIGURE 4.2: Figure used for illustrating the proposed method for an overall HME model convergence assessment. Depicted data and fitted lines have been simulated for illustration purposes only.

Assume that some MCMC chain has been run for 600 iterations. Let plots (i), (ii) and (iii) show predictions resulting from the first 200, $201 - 400$, and $401 - 600$ MCMC iterations, respectively. There appears to be high variability present across the first set of predictions suggesting that convergence has not yet been achieved. The second set of predictions exhibits a notable decrease in variability, however, the fitted lines do not appear as consistent as those seen for the last set. In this case, one may suggest that the convergence in predictions is achieved from the 400th iteration onwards.

For problems of higher dimensionality, the prediction planes can be plotted and interrogated in the same manner. Visual assessment, however, becomes more challenging for problems with many explanatory variables, and thus a formal assessment is required.

4.3.4.3 Formal Gelman-Rubin Convergence Assessment

A formal assessment of the convergence can be undertaken using the Gelman-Rubin convergence diagnostic (Gelman and Rubin, 1992; Brooks and Gelman, 1998). Let M denote the number of chains and assume that all of them are of length N. Let $\hat{\mu}_m$ and $\hat{\sigma}^2_m$ be the posterior mean and variance of the m-th chain. The statistic used to undertake the test is called the potential scale reduction factor (PSRF) and is defined to be the ratio of the pooled variance, $\hat{V}$, to the within-chain variance, $\hat{W}$, obtained as follows

$$PSRF = \frac{\hat{V}}{\hat{W}}$$
$$\hat{V} = \frac{N-1}{N}\hat{W} + \frac{M+1}{MN}\hat{B}$$
$$\hat{W} = \frac{1}{M}\sum_{m=1}^{M}\hat{\sigma}_m^2,$$

where B is between-chains variance calculated as

$$\hat{B} = \frac{N}{M-1}\sum_{m=1}^{M}(\hat{\mu}_m - \hat{\mu})^2$$

with $\hat{\mu}$ denoting the overall posterior mean of all chains.

The Gelman-Rubin method assesses and compares the estimates of the between-chains and within-chain variances of the produced predictions, where large differences between these variances indicate non-convergence. Brooks and Gelman (1998) have suggested that one can be fairly confident that convergence has been reached given the PSRF value is less than 1.2. It is important to note that the posterior mean used in the PSRF calculation does not capture the spread of the distribution. In some cases, it would thus be more beneficial to consider multivariate PSRF such as parameter posterior distribution quantiles.

If the convergence hasn't been achieved, a high between-chains variance might be pointing to poor mixing, which is discussed next.

4.3.5 Mixing for HME Models

Recall that the draws obtained from the posterior distribution are not independent, because the new value in the chain depends on the previous one. This results in correlation between samples that can cause the chain to move slowly. To evaluate the extent to which the samples produced by the MCMC are correlated, a metric of *effective sample size* denoted N_{ESS} is used. For a positively correlated sample, produced by an MCMC algorithm, the effective sample size is defined as:

$$N_{ESS} = \frac{n}{1 + 2\sum_{\tau=1}^{\infty}\rho(\tau)},$$

where n is the number of samples and $\rho = \rho(\tau)$ denotes the corellation at lag τ (Cook, 2017; Ripley, 1987). The effective sample size quantifies the loss of information due to

positive correlation in the sample. Cook (2017) proposes an easy way to understand the effective sample size. It is suggested to think of N_{ESS} as an *exchange rate* between the dependent and independent samples. For example, considered a sampler that has produced $1,000$ MCMC samples after discarding burn-in. It could be that those certain MCMC samples are *worth* 100 independent samples or 900 independent samples. The former would be the case if the MCMC samples were highly correlated and the latter would occur if the MCMC samples were weakly correlated.

A high correlation may lead to spending too much time in one region of parameter space, yielding an unreliable picture of the whole posterior distribution. In more severe cases, the chain might struggle to escape these regions and hence miss some parts of the posterior distribution entirely. Both of these situations are referred to as bad mixing (Verity, 2019).

HME models are prone to experiencing poor mixing within the MCMC chain due to their structural makeup. Having multiple models fitted across the problem space as well as having several local minima/maxima thus increases the chances of the chain getting stuck in those regions. Mixing can then be further compromised by the level of correlation present between the model parameters. Thus the proposal distributions for updating the HME model parameters should encourage wider exploration of the space that would ensure all modes of the posterior distribution are visited by the chain.

The topics of convergence and mixing are revisited for a special case of Gaussian experts in the next chapter.

Chapter 5

Normal Experts HME Sampling Methods

This chapter and the remainder of this thesis focus on a special case of HME models with Gaussian experts, which are defined in Section 5.1. Gaussian experts are an extension of the popular regression model and thus are suitable for most continuous outcome variables. In Bayesian framework, experts with normal densities are a popular choice due to expert parameters having a conjugate prior distribution available (discussed in Section 5.2). The latter leads to the full conditional posterior distribution, which then allows sampling using the Gibbs sampler. In addition, the marginal posterior distribution of the response variable can be obtained in closed-form, which can then be used in a variation of the Gibbs sampler, called the collapsed Gibbs sampler (presented in Section 5.3). Overall, when considering normal experts, a number of sampling techniques for the expert parameters become available. Section 5.4 proposes and systematically compares three sampling strategies for HME models - explicitly sampling and retaining all parameters, brute force posterior sampler, and the collapsed Gibbs sampler. All three techniques are compared in the context of the effective sample size (Section 5.4.6), run-time (Section 5.4.7), and exploration of the space (Section 5.4.8). The results are then critically assessed and discussed in Section 5.4.9 before recommending a preferred sampling strategy. Section 5.5 evaluates the convergence and mixing of the MCMC chains on a non-trivial example using the recommended sampling strategy, which highlights the challenges faced by HME models.

5.1 Definition of Normal Expert

An expert in an HME model is called a normal expert if its density function is of the form

$$f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right)=f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\beta}_E,\sigma_E^2\right)=\phi_{\mu_{iE},\sigma_E^2}(y_i), \tag{5.1}$$

for $E\in\mathcal{E}$. Here $\phi(\cdot)$ is the normal density with the mean $\mu_{iE}=\mathbf{x}_i^T\boldsymbol{\beta}_E$ and variance parameter σ_E^2.

In the case of normal experts, a conjugate normal-inverse gamma (NIG) prior for the expert parameters can be used and is discussed next.

5.2 Normal-Inverse-Gamma Prior

The following normal-inverse-gamma (NIG) priors are assigned to the parameters in (5.1):

$$\begin{aligned}\boldsymbol{\beta}_E,\sigma_E^2&\sim\mathsf{NIG}\left(\boldsymbol{\beta}_E^{(p)},V_E^{(p)},a_E^{(p)},b_E^{(p)}\right)\\ \boldsymbol{\beta}_E|\sigma_E^2&\sim\mathsf{MVN}\left(\boldsymbol{\beta}_E^{(p)},\sigma_E^2V_E^{(p)}\right)\\ \sigma_E^2&\sim\mathsf{IG}\left(a_E^{(p)},b_E^{(p)}\right),\end{aligned}$$

for $E\in\mathcal{E}$. The posterior NIG distribution can then be written as

$$\boldsymbol{\beta}_E,\sigma_E^2|X^{(E)},\mathbf{y}^{(E)}\sim\mathsf{NIG}\left(\boldsymbol{\beta}_E^{(post)},V_E^{(post)},a_E^{(post)},b_E^{(post)}\right),$$

where $X^{(E)}$ and $\mathbf{y}^{(E)}$ denote subsets of the design matrix and the response vector, respectively, containing points assigned to expert E and

$$\begin{aligned}\boldsymbol{\beta}_E^{(post)}&=\left(V_E^{(p)^{-1}}+X^{(E)^T}X^{(E)}\right)^{-1}\left(V_E^{(p)^{-1}}\boldsymbol{\beta}_E^{(p)}+X^{(E)^T}\mathbf{y}^{(E)}\right)\\ V_E^{(post)}&=\left(V_E^{(p)^{-1}}+X^{(E)^T}X^{(E)}\right)^{-1}\\ a_E^{(post)}&=a_E^{(p)}+n_E/2\\ b_E^{(post)}&=b_E^{(p)}+\frac{1}{2}\left(\boldsymbol{\beta}_E^{(p)^T}V_E^{(p)^{-1}}\boldsymbol{\beta}_E^{(p)}+\mathbf{y}^{(E)^T}\mathbf{y}^{(E)}-\boldsymbol{\beta}_E^{(post)^T}V_E^{(post)^{-1}}\boldsymbol{\beta}_E^{(post)}\right),\end{aligned} \tag{5.2}$$

where n_E denotes the number of observations assigned to expert E.

For normal experts, the expert parameters $\boldsymbol{\beta}_E$ are equivalent to the coefficients of linear regression. Thus, elements of $\boldsymbol{\beta}_E$ correspond to intercept and slope parameters in a standard linear model. The corresponding prior distributions are often centered at zero for standardised data. One could think of a prior distribution with a mean of zero as a conservative choice because it assumes an intercept only linear model with no effect of covariates. It is then the observed data that pushes the estimate away from zero and not some initial predisposition (Lunn, 2013). The selection of variance parameters for the prior distributions of the slopes can be guided by the steepness of potential regression lines. Similarly, the range of the response variable can help in determining the variance of the prior distribution for the intercept parameter.

The main argument for choosing the inverse-gamma (IG) prior for the expert variance parameter is simplicity and computational efficiency. Provided the variances are to the right of the IG distribution mode, such prior expresses a slight preference for smaller variances. An overview, presented by Gelman (2006), revealed the most commonly used values for the hyperparameters of the inverse gamma distribution to be 1, 0.01, or 0.001. Throughout the work undertaken as part of this thesis, it has been noted that smaller expert variance parameters encourage tighter allocation to experts. To illustrate, consider a single data point and probabilities defined by how likely this observation is to have come from each of the expert densities in the model. Large expert variances would encourage these probabilities to be more evenly distributed across all experts. Thus, the slight preference for smaller variances imposed by the IG prior helps avoid situations in which observations would have high probabilities of being allocated to all experts solely due to high prior within-expert variability.

Given the conjugate prior, sampling methods such as the Gibbs sampler can be used, i.e. the parameters can be drawn directly from the posterior distribution. Furthermore, Banerjee (2008) demonstrates that the marginal posterior distribution of the response variable in the conjugate case can be written as

$$f^{(E)}(\mathbf{y}) = \frac{\exp\left(a_E^{(p)} \log\left(b_E^{(p)}\right)\right) \cdot \Gamma\left(a_E^{(post)}\right) \sqrt{\left|V_E^{(p)}\right|}}{(2\pi)^{\frac{n_E}{2}} \Gamma\left(a_E^{(p)}\right) \sqrt{\left|V_E^{(post)}\right|}} \cdot \exp\left(-{a_E}^{(post)} \log\left(b_E^{(post)}\right)\right), \tag{5.3}$$

which means that if sampling expert parameters is not of interest, a variant of the Gibbs sampler, called the collapsed Gibbs sampler, focusing on expert allocations for HME sampling can be implemented.

5.3 Collapsed Gibbs Sampler for HME Sampling

The idea behind a collapsed Gibbs sampler is to integrate out as many (noise) variables as possible before sampling from the conditional distribution of the variables of interest (Liu, 1994). It has been shown in the previous section that when the NIG prior is used for Gaussian experts, the marginal posterior distribution of the response can be obtained. Using this result, the steps of the collapsed Gibbs sampler for updating the allocation variables are outlined.

As before, let $\mathbf{z}_{z_i=1}^{(G)}$ denote a subset of size n_G containing points that have reached gate G. In addition to gating parameter updates, this subset can then be updated as

1. Determine all terminal expert nodes descending from the gate G and denote the set $\mathcal{E}'$.
2. For each point $j = 1, ..., n_G$ in the set $\mathbf{z}_{z_i=1}^{(G)}$:
   (a) Calculate
   $$\alpha_j^{(E)} = \left(\prod_{(G,H)\in P_>^{(G)}} \pi_j^{(G,H)} \right) f^{(E)}\left(\mathbf{y}_{j+}^{(E)}\right), \tag{5.4}$$
   for all $E \in \mathcal{E}'$, where $f^{(E)}(\cdot)$ denotes the marginal posterior distribution of the response variable as per (5.3) and $\mathbf{y}_{j+}^{(E)}$ denotes all points that are already in expert E and the j-th point.
   (b) Assign the j-th point to expert E, i.e., set $z_j^{(E)} = 1$ and $z_j^{(E'')} = 0$ for $E, E'' \in \mathcal{E}'$ and for $E'' \neq E$, with probability $\alpha_j^{(E)}$, where $\sum_{E\in\mathcal{E}'} \alpha_j^{(E)} = 1$.

Eliminating the sampling of expert parameters decreases the total number of parameters to be sampled and hence could potentially make the sampling process more efficient.

Complete MCMC update steps for all parameters in HME models with Gaussian experts are outlined in the following section, where the collapsed Gibbs sampler and two alternative sampling strategies are evaluated in terms of their produced effective sample size, run-time, and exploration of space.

5.4 Systematic Evaluation of HME Model Parameter Sampling Strategies

This section proposes and evaluates three expert parameter sampling techniques in the case of a mixture model with two Gaussian experts. Firstly, the associated model density

function is defined and the data sets, simulated for the purpose of this evaluation, are presented in Section 5.4.1. Next, the three sampling techniques are introduced in Section 5.4.2, and, more formally, the steps of the proposed sampling algorithms are outlined in Sections 5.4.3, 5.4.4 and 5.4.5. The samplers of interest are then evaluated on two simulated data sets with respect to metrics of effective sample size and acceptance rates (Section 5.4.6), run-time (Section 5.4.7), and exploration of the space (Section 5.4.8). Lastly, the results of the systematic comparison are discussed and critically assessed in Section 5.4.9 before deciding on the recommended sampling strategy.

5.4.1 Two Normal Expert ME Case

The density of the mixture of two normal experts E^* and E^{**} can be written as:

$$\begin{aligned} f\left(y_i|\mathbf{x}_i,\boldsymbol{\phi}\right) &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right) \\ &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\beta}_E,\sigma_E^2\right) \\ &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} \phi_{\mu_{iE},\sigma_E^2}(y_i) \\ &= \pi_i^{(E^*)} \phi_{\mu_{iE^*},\sigma_{E^*}^2}(y_i) + \pi_i^{(E^{**})} \phi_{\mu_{iE^{**}},\sigma_{E^{**}}^2}(y_i) \end{aligned} \tag{5.5}$$

where $\boldsymbol{\phi}$ denotes a vector of all parameters in the HME model while $\boldsymbol{\theta}^{(E)} = \left(\boldsymbol{\beta}_E, \sigma_E^2\right)$ denotes a vector of distinct parameters occurring in the expert E density for $E \in (E^*, E^{**})$, and $\phi_{\mu,\sigma^2}(\cdot)$ denotes the normal density with the mean μ and variance σ^2, and $\mu_{iE} = \mathbf{x}_i^T\boldsymbol{\beta}_E$.

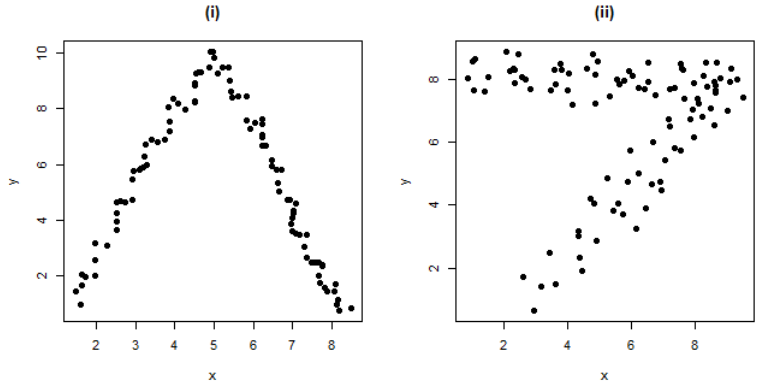


FIGURE 5.1: Simulated data sets used for the evaluation of three ME parameter sampling strategies.

The evaluation undertaken in this section is performed on the simulated data sets shown in Figure 5.1. The first data set (i) is a straightforward mixture of two experts that are well separated while the second example (ii) investigates a slightly more complex case where the separation between two experts cannot be defined by some explanatory variable value. In both scenarios, there is one explanatory variable present, i.e., $\mathbf{x}_i = (1, x_i)$ and $\boldsymbol{\beta}_E = (\beta_{0E}, \beta_{1E})$, where β_{0E} and β_{1E} correspond to the intercept and slope parameters of simple linear regression, respectively. In such case, the density (5.5) can be written as

$$\begin{aligned} f\left(y_i|\mathbf{x}_i, \boldsymbol{\phi}\right) &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right) \\ &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\beta}_E, \sigma_E^2\right) \\ &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \beta_{0E}, \beta_{1E}, \sigma_E^2\right) \\ &= \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} \phi_{\mu_{iE},\sigma_E^2}(y_i) \\ &= \pi_i^{(E^*)} \phi_{\mu_{iE^*},\sigma_{E^*}^2}(y_i) + \pi_i^{(E^{**})} \phi_{\mu_{iE^{**}},\sigma_{E^{**}}^2}(y_i), \end{aligned} \tag{5.6}$$

where $\mu_{iE} = \beta_{0E} + \beta_{1E} x_i$ for $E \in (E^*, E^{**})$. Recall that for ME models the mixing proportions are defined as

$$\pi_i^{(E)} = \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(E)}\right)}{\sum_{E\in\mathcal{E}} \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(E)}\right)}$$

for $E \in (E^*, E^{**})$. For a binary split, there are two gating parameters present in the model, i.e., $\boldsymbol{\gamma}^{(E^*)}$ and $\boldsymbol{\gamma}^{(E^{**})}$. As always, the gating parameters corresponding to the first expert are set to zero in order to ensure identifiability, i.e., $\boldsymbol{\gamma}^{(E^*)} = \mathbf{0}$, leaving $\boldsymbol{\gamma}^{(E^{**})}$ to be estimated. Let us simplify the notation by writing $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(E^{**})}$. The mixing proportions for the two experts are then

$$\pi_i^{(E^*)} = \frac{1}{1 + \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}\right)} \ , \ \pi_i^{(E^{**})} = \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}\right)}{1 + \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}\right)},$$

where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$ and $\pi_i^{(E^*)} + \pi_i^{(E^{**})} = 1$ for $i = 1, ...n$.

For a mixture of two Gaussian experts with one explanatory variable, a sampling methodology should cover updating the two-dimensional gating parameter $\boldsymbol{\gamma}$, allocation variables $\mathbf{z}$, and expert parameters, which include intercepts β_{0E}, slopes β_{1E}, and variance parameters σ_E^2 for two experts E^* and E^{**}. In this section, three sampling approaches, which

span a mixture of Metropolis-Hastings, Gibbs sampler, and collapsed Gibbs sampler updates are presented. To ensure comparability, the same parameter priors are set for all samplers. To allow for wider exploration of expert parameter space, weakly informative priors are chosen for the intercept and slope parameters while a narrower expert variance parameter is set to reflect the low variability present in the simulated data:

$$\boldsymbol{\beta}_E, \sigma_E^2 \sim \mathsf{NIG}\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, 0.001, 0.001\right).$$

Similarly, a wide prior centered at zero is chosen for the gating parameter in order to explore various degrees of abruptness in separation between the two experts

$$\boldsymbol{\gamma} \sim MVN\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}\right). \tag{5.7}$$

All sampling strategies considered update the gating parameter $\boldsymbol{\gamma}$ using the Metropolis-Hastings algorithm with the following proposal distribution

$$\boldsymbol{\gamma}^* \sim MVN\left(\boldsymbol{\gamma}, \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}\right),$$

where $\boldsymbol{\gamma}^*$ and $\boldsymbol{\gamma}$ denote the new proposed and the previous values for the gating parameter, as outlined in Section 4.3.3.2. The moderate variance of the proposal distribution is chosen to encourage a careful exploration of the parameter space for a relatively simple problem.

Having stated the model density function and the associated parameter priors, the proposed sampling techniques are introduced next.

5.4.2 Sampling Strategies

Table 5.1 summarises the three proposed strategies for sampling the parameters of mixture of two experts model.

TABLE 5.1: Summary of the three samplers considered. The explicitly sampled parameters are denoted by ✓and not explicitly sampled parameters are denoted by ✗.

	$\boldsymbol{\gamma}$	$\boldsymbol{\beta}_E$	σ_E^2	$\mathbf{z}_i$
Sampler I	✓	✓	✓	✓
Sampler II	✓	✓	✓	✗
Sampler III	✓	✗	✗	✓

The first sampler corresponds to a standard Bayesian sampling technique, where all model parameters are sampled and retained (Hurn et al., 2003). To evaluate ME density of the response y_i, given a vector of covariates, $\mathbf{x}_i$, one is required to obtain the mixing proportions as well as individual expert densities. As shown in Figure 5.2, normal expert parameters, $\boldsymbol{\beta}_E$ and σ_E^2, feed into the evaluation of the density in parallel to the gating parameters, $\boldsymbol{\gamma}$, that are used to calculate the mixing proportions $\pi_i^{(E)}$ and obtain the corresponding latent allocation variables $\mathbf{z}_i$. The latter standard approach is compared with two alternative ones.

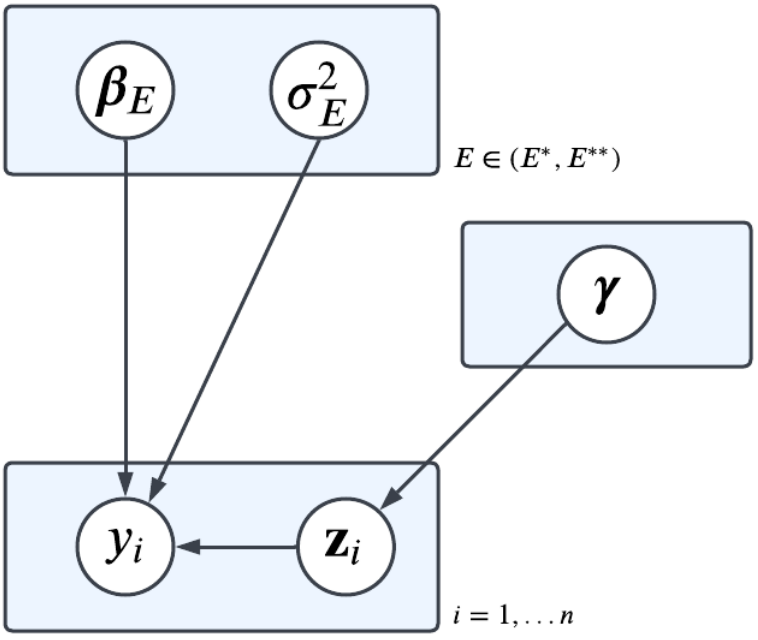


FIGURE 5.2: Illustration of Sampler I.

Sampler II is the brute force posterior sampler, which does not make use of the latent allocation variables. As discussed previously, in such cases the maximum a posteriori estimates of the expert parameters cannot be found in closed-form. This section proposes

a step-by-step method, which includes the Metropolis-Hastings algorithm, to sample the expert parameters under the given circumstances.

Finally, Sampler III is the collapsed Gibbs sampler discussed in Section 5.3. The expert parameters are integrated out hence reducing the number of parameters sampled. During the update, the marginal posterior distribution of the response variable is evaluated instead of the log-likelihood function.

All three samplers are run for $5,000$ iterations with the first $1,000$ draws discarded for burn-in, leaving samples with $4,000$ iterations. Trace plots have been used to confirm that convergence has been achieved for model parameters across all samplers considered. All sampling techniques, discussed in this chapter, produced consistently similar results in terms of the posterior means and credible intervals of the model parameters (see Appendix D).

Next, the steps of the three sampling strategies considered are outlined more formally.

5.4.3 Sampler I

The first approach considered involves sampling and retaining all of the parameters (see Table 5.1). Given some starting values $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{\beta}_E^{(0)}$, $\sigma_E^{2(0)}$ and $\mathbf{z}_i^{(0)}$ and proposal distribution parameter values, the following steps are iterated for $t = 1, ..., T$:

1. Update $\boldsymbol{\gamma}^{(t)}$ using Metropolis-Hastings algorithm as follows:

   (a) Generate a proposed value $\boldsymbol{\gamma}^*$ from the proposal distribution

   $$\boldsymbol{\gamma}^* \sim MVN\left(\boldsymbol{\gamma}^{(t-1)}, \Sigma_{\gamma^*}\right),$$

   (b) Accept $\boldsymbol{\gamma}^*$ with probability

   $$\begin{aligned}\alpha &= \min\left\{1, \frac{f\left(\boldsymbol{\gamma}^*|\mathbf{z}^{(t-1)}\right) g\left(\boldsymbol{\gamma}^{(t-1)}|\boldsymbol{\gamma}^*\right)}{f\left(\boldsymbol{\gamma}^{(t-1)}|\mathbf{z}^{(t-1)}\right) g\left(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}^{(t-1)}\right)}\right\} \\ &= \min\left\{1, \frac{f\left(\mathbf{z}^{(t-1)}|\boldsymbol{\gamma}^*\right) f\left(\boldsymbol{\gamma}^*\right) g\left(\boldsymbol{\gamma}^{(t-1)}|\boldsymbol{\gamma}^*\right)}{f\left(\mathbf{z}^{(t-1)}|\boldsymbol{\gamma}^{(t-1)}\right) f\left(\boldsymbol{\gamma}^{(t-1)}\right) g\left(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}^{(t-1)}\right)}\right\}\end{aligned}$$

   where $f(\mathbf{z}|\boldsymbol{\gamma})$ and $f(\boldsymbol{\gamma})$ denote the gate likelihood function and prior distribution as per (3.10) and (5.7), respectively, while $g(\cdot)$ corresponds to the density of the proposal distribution outlined in step 1a.

   (c) If the proposal is accepted, set $\boldsymbol{\gamma}^{(t)} = \boldsymbol{\gamma}^*$. Otherwise, set $\boldsymbol{\gamma}^{(t)} = \boldsymbol{\gamma}^{(t-1)}$.

2. Calculate $\boldsymbol{\pi}_i^{(t)} = \left(\pi_i^{(E^*)(t)}, \pi_i^{(E^{**})(t)}\right)$ as

$$\pi_i^{(E^*)(t)} = \frac{1}{1+\exp\left(\mathbf{x}_i^T\boldsymbol{\gamma}^{(t)}\right)} \text{ and } \pi_i^{(E^{**})(t)} = \frac{\exp\left(\mathbf{x}_i^T\boldsymbol{\gamma}^{(t)}\right)}{1+\exp\left(\mathbf{x}_i^T\boldsymbol{\gamma}^{(t)}\right)}$$

for $i = 1, ..., n$, where $\mathbf{x}_i = (1, x_i)$.

3. Update $\boldsymbol{\beta}_E^{(t)}$ and $\sigma_E^{2(t)}$ for $E \in (E^*, E^{**})$ by drawing the posterior parameters from the NIG posterior distribution

$$\boldsymbol{\beta}_E^{(t)}, \sigma_E^{2(t)} | X^{(E)(t)}, \mathbf{y}^{(E)(t)} \sim \mathsf{NIG}\left(\boldsymbol{\beta}_E^{(post)(t)}, V_E^{(post)(t)}, a_E^{(post)(t)}, b_E^{(post)(t)}\right),$$

where the posterior parameters $\boldsymbol{\beta}_E^{(post)(t)}, V_E^{(post)(t)}, a_E^{(post)(t)}, b_E^{(post)(t)}$ at iteration t are obtained as per (5.2), and $X^{(E)(t)}$ and $\mathbf{y}^{(E)(t)}$ denote the design matrix and the response vector, respectively, containing points assigned to expert E at iteration t.

4. Update $\mathbf{z}_i^{(t)}$ for $i = 1, ..., n$ as follows

   (a) Calculate

$$\alpha_i^{(E)} = \pi_i^{(E)(t)} f^{(E)}\left(y_i | \mathbf{x}_i, \boldsymbol{\theta}^{(E)(t)}\right),$$

   for $E \in (E^*, E^{**})$, where $\sum_{E\in(E^*,E^{**})} \alpha_i^{(E)} = 1$.

   (b) Assign the i-th point to expert E, i.e., set $z_i^{(E)(t)} = 1$ and $z_i^{(E')(t)} = 0$ for $E' \neq E$ with probability $\alpha_i^{(E)}$.

5.4.4 Sampler II

The second approach considered involves sampling and retaining expert and gate parameters only (see Table 5.1). This sampler does not take advantage of the addition of allocation variables and thus cannot use the full conditional posterior distribution for expert parameter sampling. In this section, an alternative method for sampling the expert parameters under such circumstances is proposed.

Given some starting values $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{\beta}_E^{(0)}$ and $\sigma_E^{2(0)}$ and proposal distribution parameter values, the following steps are iterated for $t = 1, ..., T$:

1. Update $\boldsymbol{\gamma}^{(t)}$ in the same way as seen for Sampler I, however using the data likelihood function, $\prod_{i=1,...,n} L\left(y_i | \mathbf{x}_i, \boldsymbol{\phi}\right)$, given the other parameters in lieu of the conditional distribution of $\boldsymbol{\gamma}$ given $\mathbf{z}$, $f\left(\mathbf{z}|\boldsymbol{\gamma}\right)$.

2. Calculate $\boldsymbol{\pi}_i^{(t)} = \left(\pi_i^{(E^*)(t)}, \pi_i^{(E^{**})(t)}\right)$ for $i = 1, ..., n$ in the same way as seen for Sampler I.

3. Update $\boldsymbol{\beta}_E^{(t)}$ for $E \in (E^*, E^{**})$ as follows

   (a) Draw
   $$\begin{pmatrix} \delta \\ \epsilon \end{pmatrix} \sim \left(MVN \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_\delta^2 & 0 \\ 0 & \sigma_\epsilon^2 \end{pmatrix} \right).$$

   (b) Obtain the proposed value for $\boldsymbol{\beta}_E^{(t)}$ for $E \in (E^*, E^{**})$ as follows:
   $$\begin{aligned} \boldsymbol{\beta}_E^* = \begin{pmatrix} \beta_{0E}^* \\ \beta_{1E}^* \end{pmatrix} &= \begin{pmatrix} \beta_{0E}^{(t-1)} + \bar{x}_E \left(\beta_{1E}^{(t-1)} - \beta_{1E}^* \right) + \delta \\ \beta_{1E}^{(t-1)} + \epsilon \end{pmatrix} \\ &= \begin{pmatrix} \beta_{0E}^{(t-1)} - \bar{x}_E \cdot \epsilon + \delta \\ \beta_{1E}^{(t-1)} + \epsilon \end{pmatrix}, \end{aligned}$$

   where $\bar{x}_E = \sum_i \pi_i^{(E)(t)} x_i$. For small σ_δ^2, we have $\beta_{0E}^{(t-1)} + \beta_{1E}^{(t-1)} \bar{x}_E \approx \beta_{0E}^* + \beta_{1E}^* \bar{x}_E$, which ensures that the random perturbation does not move the regression line by a large amount if the values of the covariates are not centered (which for the data being allocated to an expert cannot be guaranteed).

   (c) Set $\boldsymbol{\beta}_{E^*}^{(t)} = \boldsymbol{\beta}_{E^*}^*$ with probability
   $$\begin{aligned} \alpha = \min \Bigg(1, & \frac{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^*, \sigma_{E^*}^{2(t-1)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t-1)}, \sigma_{E^{**}}^{2(t-1)}}(y_i) \right]}{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t-1)}, \sigma_{E^*}^{2(t-1)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t-1)}, \sigma_{E^{**}}^{2(t-1)}}(y_i) \right]} \\ & \times \frac{f\left(\boldsymbol{\beta}_{E^*}^* | \sigma_{E^*}^{2(t-1)} \right)}{f\left(\boldsymbol{\beta}_{E^*}^{(t-1)} | \sigma_{E^*}^{2(t-1)} \right)} \Bigg), \end{aligned}$$

   where $\phi_{\mu,\sigma^2}(\cdot)$ is the Normal density with mean μ and variance σ^2. Having updated $\boldsymbol{\beta}_{E^*}^{(t)}$, set $\boldsymbol{\beta}_{E^{**}}^{(t)} = \boldsymbol{\beta}_{E^{**}}^*$ with probability
   $$\begin{aligned} \alpha = \min \Bigg(1, & \frac{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t)}, \sigma_{E^*}^{2(t-1)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^*, \sigma_{E^{**}}^{2(t-1)}}(y_i) \right]}{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t)}, \sigma_{E^*}^{2(t-1)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t-1)}, \sigma_{E^{**}}^{2(t-1)}}(y_i) \right]} \\ & \times \frac{f\left(\boldsymbol{\beta}_{E^{**}}^* | \sigma_{E^{**}}^{2(t-1)} \right)}{f\left(\boldsymbol{\beta}_{E^{**}}^{(t-1)} | \sigma_{E^{**}}^{2(t-1)} \right)} \Bigg), \end{aligned}$$

   where $\phi_{\mu,\sigma^2}(\cdot)$ is the Normal density with mean μ and variance σ^2. [1]

4. Update $\sigma_E^{2(t)}$ for $E \in (E^*, E^{**})$ as follows

   (a) Draw $u \sim N(0, \sigma_u^2)$.

   (b) Propose a new value of $\sigma_E^{2*} = \sigma_E^{2(t-1)} \times \exp(u)$ for $E \in (E^*, E^{**})$.

[1] For simplicity, this step has been stated for a special case of two experts E^* and E^{**}.

(c) Set $\sigma_{E^*}^{2(t)} = \sigma_{E^*}^{2*}$ with probability

$$\alpha = \min\left(1, \frac{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t)}, \sigma_{E^*}^{2*}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t)}, \sigma_{E^{**}}^{2(t-1)}}(y_i)\right]}{\prod_i \left[\pi_i^{E^*(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t)}, \sigma_{(E^*)}^{2(t-1)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t)}, \sigma_{E^{**}}^{2(t-1)}}(y_i)\right]} \times \frac{f\left(\sigma_{E^*}^{2*}\right)}{f\left(\sigma_{E^*}^{2(t-1)}\right)} \times \frac{q\left(\sigma_{E^*}^{2(t-1)} | \sigma_{E^*}^{2*}\right)}{q\left(\sigma_{E^*}^{2*} | \sigma_{E^*}^{2(t-1)}\right)}\right),$$

where $\phi_{\mu,\sigma^2}(\cdot)$ is the Normal density with mean μ and variance σ^2. Having updated $\sigma_{E^*}^{2(t)}$, set $\sigma_{E^{**}}^{2(t)} = \sigma_{E^{**}}^{2*}$ with probability

$$\alpha = \min\left(1, \frac{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t)}, \sigma_{E^*}^{2(t)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t)}, \sigma_{E^{**}}^{2*}}(y_i)\right]}{\prod_i \left[\pi_i^{(E^*)(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^*}^{(t)}, \sigma_{E^*}^{2(t)}}(y_i) + \pi_i^{(E^{**})(t)} \phi_{\mathbf{x}_i^T \boldsymbol{\beta}_{E^{**}}^{(t)}, \sigma_{E^{**}}^{2(t-1)}}(y_i)\right]} \times \frac{f\left(\sigma_{E^{**}}^{2*}\right)}{f\left(\sigma_{E^{**}}^{2(t-1)}\right)} \times \frac{q\left(\sigma_{E^{**}}^{2(t-1)} | \sigma_{E^{**}}^{2*}\right)}{q\left(\sigma_{E^{**}}^{2*} | \sigma_{E^{**}}^{2(t-1)}\right)}\right),$$

where $\phi_{\mu,\sigma^2}(\cdot)$ is the Normal density with mean μ and variance σ^2. [2]

Given that the simulated data presents two rather simple cases, all proposal parameters selected for this evaluation are $\sigma_\delta^2 = \sigma_\epsilon^2 = \sigma_u^2 = 0.1$, which, after experimenting with various values, proved to be sufficient when exploring the particular expert parameter spaces.

5.4.5 Sampler III

The last method of sampling from a mixture of experts model considered involves integrating out the expert parameters (see Table 5.1). Given some starting values $\boldsymbol{\gamma}^{(0)}$, $\mathbf{z}_i^{(0)}$, and proposal distribution parameter values, the following steps are iterated for $t = 1, ..., T$:

1. Update $\boldsymbol{\gamma}^{(t)}$ in the same way as seen for Sampler I.
2. Calculate $\boldsymbol{\pi}_i^{(t)} = \left(\pi_i^{(E^*)(t)}, \pi_i^{(E^{**})(t)}\right)$ for $i = 1, ..., n$ in the same way as seen for Sampler I.
3. Update $\mathbf{z}_i^{(t)}$ for $i = 1, ..., n$ using the collapsed Gibbs sampler as follows

[2] For simplicity, this step has been stated for a special case of two experts E^* and E^{**}.

(a) Calculate

$$\alpha_i^{(E)} = \pi_i^{(E)(t)} f^{(E)}\left(\mathbf{y}_{i+}^{(E)}\right), \tag{5.8}$$

for all $E \in (E^*, E^{**})$, where $\sum_{E\in(E^*,E^{**})} \alpha_i^{(E)} = 1$, and $f^{(E)}(\cdot)$ denotes the marginal posterior distribution of the response variable as per (5.3) and $\mathbf{y}_{i+}^{(E)}$ denotes all points that are already in expert E and the i-th point.

(b) Assign the i-th point to expert E, i.e., set $z_i^{(E)(t)} = 1$ and $z_i^{(E')(t)} = 0$ for $E' \neq E$ with probability $\alpha_i^{(E)}$.

5.4.6 Effective Sample Size and Acceptance Comparison for HME Samplers

In this section, the two data sets shown in Figure 5.1 are discussed in the context of the effective sample size produced by the three samplers. The results for the first data set shown in Figure 5.1 (i) are presented in Table 5.2 and the corresponding acceptance rates are given in Table 5.3.

TABLE 5.2: Effective sample size (ESS) for the first data set shown in Figure 5.1 (i).

N_{ESS}	$\boldsymbol{\gamma}_0$	$\boldsymbol{\gamma}_1$	β_{0E^*}	β_{1E^*}	$\beta_{0E^{**}}$	$\beta_{1E^{**}}$	$\sigma^2_{E^*}$	$\sigma^2_{E^{**}}$
Sampler I	661.95	655.40	3970.41	3181.09	3116.80	3229.43	3821.37	4000.00
Sampler II	595.17	588.65	566.14	544.76	441.29	448.49	699.73	791.36
Sampler III	635.49	639.86	–	–	–	–	–	–

TABLE 5.3: Acceptance rates for the first data set shown in Figure 5.1 (i).

Acceptance Rate	$\boldsymbol{\gamma}$	$\boldsymbol{\beta}_{E^*}$	$\boldsymbol{\beta}_{E^{**}}$	$\sigma^2_{E^*}$	$\sigma^2_{E^{**}}$
Sampler I	0.30	(1)	(1)	(1)	(1)
Sampler II	0.32	0.40	0.38	0.59	0.56
Sampler III	0.33	–	–	–	–

It can be seen that the effective sample size for the gating parameter is similar across the three samplers. As discussed in Section 4.3.5, this can be thought of as $4,000$ MCMC samples producing 661.95, 595.17, and 635.49 independent samples in the case of sampler I, II and III respectively. The acceptance rates for the gating parameter are in the range of $30 - 33\%$ for all three samplers (see Table 5.3). Further, it is clear that Sampler I outperforms Sampler II in terms of the effective sample size for the expert parameters. The latter is not surprising, because Sampler II uses the Metropolis-Hastings updating

step, which is known to perform substantially worse in this metric as correlation increases (Turner et al., 2013). The acceptance rates for the expert parameters are only meaningful for Sampler II, where the Metropolis-Hastings update is used. The acceptance rates seem satisfactory and are in the range of $38\% - 59\%$. Overall, it is evident that Sampler I and Sampler III outperform Sampler II in the case of the first data set.

Next, the analogous results for the second data set (Figure 5.1 (ii)) are investigated. The effective sample size results are shown in Table 5.4 and the corresponding acceptance rates are given in Table 5.5.

TABLE 5.4: Effective sample size (ESS) for the second data set shown in Figure 5.1 (ii).

N_{ESS}	γ_0	γ_1	β_{0E^*}	β_{1E^*}	$\beta_{0E^{**}}$	$\beta_{1E^{**}}$	$\sigma^2_{E^*}$	$\sigma^2_{E^{**}}$
Sampler I	158.47	105.82	2897.17	1787.74	2670.55	2310.25	2937.03	2712.67
Sampler II	281.19	234.37	460.21	428.58	684.36	571.74	330.00	502.00
Sampler III	197.23	139.67	–	–	–	–	–	–

TABLE 5.5: Acceptance rates for the second data set shown in Figure 5.1 (ii).

Acceptance Rate	$\boldsymbol{\gamma}$	$\boldsymbol{\beta}_{E^*}$	$\boldsymbol{\beta}_{E^{**}}$	$\sigma^2_{E^*}$	$\sigma^2_{E^{**}}$
Sampler I	0.12	(1)	(1)	(1)	(1)
Sampler II	0.16	0.60	0.31	0.71	0.67
Sampler III	0.12	–	–	–	–

It is evident that the highest effective sample size of 281.19 for the gating parameter is achieved by Sampler II - the opposite result of the one seen for the first data set. However, the magnitude of the difference between the highest and the lowest result is relatively low (121.72). Further, it is evident that Sampler I outperforms Sampler II in terms of the effective sample size for the expert parameters, which is in agreement with the results seen for the first data set. The acceptance rates for the gating parameter are lower than those seen for the first data set and range between $12 - 16\%$, with Sampler II yielding the highest acceptance rate. Three out of four acceptance rates for the remaining expert parameters are higher than the ones seen for the first data set. Overall, it is clear that all three samplers perform worse on the second data set. This is not surprising as the second data set was designed to be more challenging when it comes to *cutting the space* with respect to x.

5.4.7 Run Time Comparison for HME Samplers

When comparing the three sampling techniques, it is important to consider the cost of running them. We proceed by looking at the run time for the three samplers, which is summarised in Table 5.6.

TABLE 5.6: Run time in seconds for the three sampling techniques applied to the two data sets shown in Figure 5.1 measured for all $5,000$ iterations.

Time (s)	Data I	Data II
Sampler I	8.54	8.59
Sampler II	6.23	6.16
Sampler III	88.98	88.07

It is evident that Sampler III is notably slower than Sampler I and Sampler II. This means that integrating the expert parameters out (using collapsed Gibbs sampler) is more costly than sampling and retaining them as proposed by the Sampler I. Further, it can be seen that Sampler II is marginally quicker than Sampler I across both data sets. However, in the previous section, it was discovered that Sampler II performs the worst in terms of the effective sample size. An increase of 2.31 and 2.43 seconds in run time yields an average effective sample size increase from 584.45 to $2,829.56$ and from 436.56 to $1,947.46$ for the first and second data sets respectively. Next, the exploration of the space for the three techniques is assessed.

5.4.8 Exploration of the Space Comparison for HME Samplers

Another important metric to consider when comparing the three samplers is how well the parameter space is explored. A simple way to quantify the exploration of the space is to investigate the posterior variance of the parameter estimates produced by the MCMC (Table 5.7). The higher the variance, the more space that has been explored by the sampler.

TABLE 5.7: Posterior variances of the MCMC parameter samples across the two data sets shown in Figure 5.1 and three HME sampling techniques.

Variance	γ_0	γ_1	β_{0E^*}	β_{1E^*}	$\beta_{0E^{**}}$	$\beta_{1E^{**}}$	$\sigma^2_{E^*}$	$\sigma^2_{E^{**}}$
	Data 1							
Sampler I	3.40	0.13	0.24	0.02	1.10	0.02	0.002	0.004
Sampler II	3.34	0.13	0.04	0.004	0.21	0.005	0.002	0.002
Sampler III	3.41	0.13	–	–	–	–	–	–
	Data 2							
Sampler I	0.61	0.03	0.30	0.007	0.10	0.003	0.02	0.002
Sampler II	0.56	0.02	0.19	0.05	0.02	0.0007	0.02	0.002
Sampler III	0.72	0.03	–	–	–	–	–	–

It is evident that all samplers perform similarly with Sampler I performing better across all parameters for the first data set. It is important to remember that the data sets considered here are rather simple and the modes of the posterior distributions can be found without requiring much exploring. Once a preferred sampling technique is determined, the topic of space exploration is revisited for a more complex example in Section 5.5.

5.4.9 Discussion

The investigation of the three sampling techniques revealed that all three samplers yield a similar effective sample size result for the gating parameter, however Sampler I performs notably better than Sampler II for expert parameters.

The run time experiment has highlighted that running Sampler III takes approximately 10 times longer than Sampler I and approximately 14 times longer than Sampler II. A major advantage of Sampler III is not sampling and retaining the expert parameters, which might become increasingly beneficial as the number of experts in the tree grows resulting in storage issues. On the other hand, Sampler III does not allow one to view the expert parameter estimates produced by the MCMC chain. Therefore, if it is of interest to investigate the expert densities in detail for each iteration, the required parameters would need to be drawn from their posterior distributions thus increasing the run time even further.

Given the notable reduction in the number of parameters sampled, there is, however, scope and merit in trying to improve the run time of Sampler III. For example, computing the marginal posterior from scratch requires $O(p^3)$ operations, where p is the number of expert parameters. However, the change to the already computed inverse is just the

addition or removal of a single observation, which corresponds to a rank one update. By exploiting the Sherman-Morrison formula (Sherman and Morrison, 1950), which is a special case of the Woodbury formula (Woodbury, 1950), the inverse can be calculated in just $O(p^2)$ operations. For large p this will lead to a considerable speed increase. Future research could include optimising or parallelising some calculations as well as further exploring caching the results and using matrix decompositions.

Lastly, it has been shown that all samplers yield a similar result for the posterior variances of parameter estimates produced by the MCMC. The latter suggests that all samplers explore the parameter space to a similar degree in the considered application.

Given the above results, Sampler I is chosen as a recommended sampling technique going forward.

The work undertaken in this section is directly applicable to hierarchical mixture of experts models, where all experts are Gaussian and all splits are binary. Having determined the recommended strategy for sampling such model parameters, the topics of space exploration, also known as mixing, and convergence are discussed for a more complicated application.

5.5 Motorcycle Accident Data Application

5.5.1 Motorcycle Accident Data Introduction

Given that the simulated data sets used so far are relatively straightforward, another example, shown in Figure 5.3, is considered. The study, conducted by Schmidt et al. (1981), presents a series of measurements in a simulated motorcycle accident, used to test crash helmets.

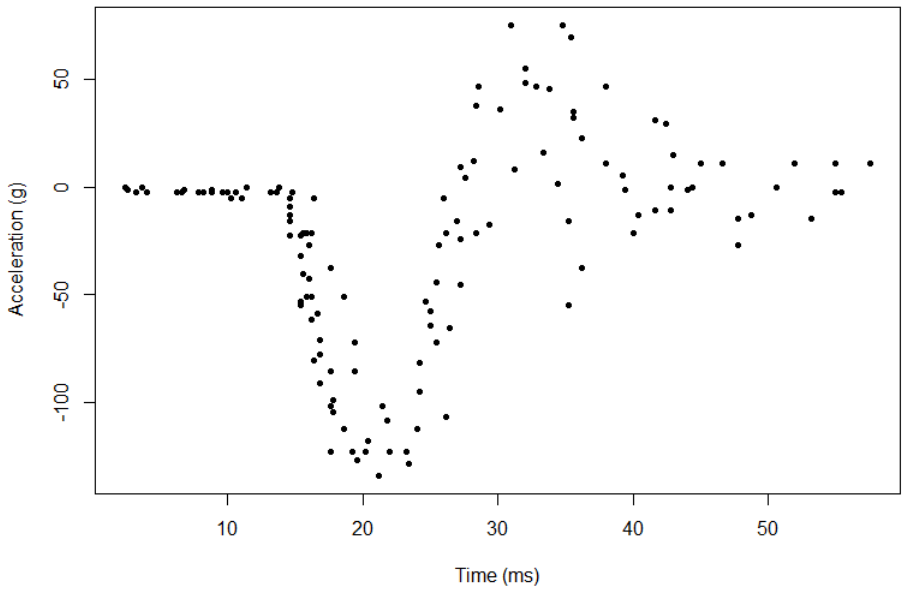


FIGURE 5.3: Simulated motorcycle accident data.

The explanatory variable, $\mathbf{x} = (x_1, ..., x_{133})^T$, is the time (in ms) after a simulated impact on motorcycles, and the response variable, $\mathbf{y} = (y_1, ..., y_{133})^T$, is the head acceleration (in g) of a test object (Chen et al., 2009). The dataset is widely available as `mcycle` in the `R` package `MASS` and was initially used by Silverman (1985) to showcase some aspects of the spline smoothing for nonparametric regression fitting.

It is known that the time points are not regularly spaced, and there are multiple observations at some time points (Silverman, 1985). From Figure 5.3, it is clear that the variance of the data is not constant, i.e., data exhibits heteroscedasticity. In fact, the variance of observations increases as time increases. There are also several change points present in the data. The transitions between different patterns in the response range from smooth, as seen at around 40 ms, to abrupt, as seen in the area around 15 ms. The latter characteristics make the motorcycle data set challenging for standard models to fit. On the other hand, the greater complexity of HME models can adapt to counter the challenges posed by the data without sacrificing the interpretability.

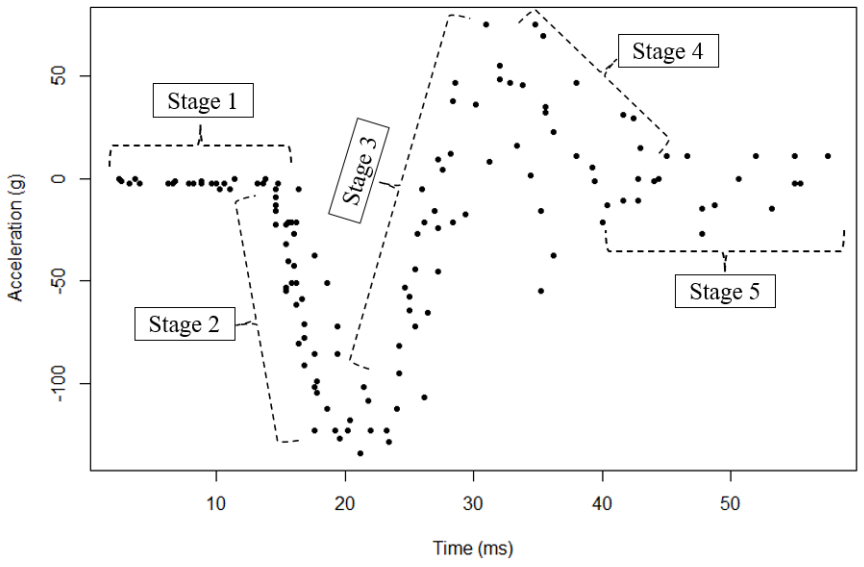


FIGURE 5.4: Simulated motorcycle accident data split into 5 stages.

Looking at the relationship between time and acceleration in more detail (Figure 5.4), it can be seen that the acceleration stays at just under zero for the first 15 ms after the impact (stage 1), which is followed by a steep deceleration with approximately -120 g reached at around 21 ms post-impact (stage 2). After the point of change, the acceleration exhibits a steep increase up to around 70 g until roughly 35 ms after impact (stage 3). Once again, a change in the relationship between time and acceleration is observed with the response decreasing until approximately 40 ms (stage 4) before leveling off and staying at around zero acceleration (stage 5). It is interesting to note that for stages 1 and 5, one could fit an intercept-only linear model, however, the variability present in the response at stage 1 is notably smaller than that observed at stage 5. Similarly, for stages 2 and 4, there appears to be a negative association present between the time and acceleration with a notably higher variance in response present for stage 4. Lastly, it is noted that the separation between stages 4 and 5 is rather smooth and could potentially be represented by one linear model. It is thus anticipated that an HME with four to five experts would be sufficient in this scenario.

At this stage, the motorcycle accident data is used to illustrate several challenges faced by HME models. These shortcomings are then addressed and improved upon in the subsequent chapters.

5.5.2 HME Model Fitting for Motorcycle Accident Data

In this section, an HME model is fitted to the motorcycle data set using the previously recommended sampling strategy (Sampler I). Firstly, it is noted that the response values range between -135 g and 75 g. Such wide range and varying magnitude implies high variability across the potential expert parameters and thus makes the selection of the prior distributions rather challenging. Generally, it is a common practice in machine learning to perform data standardisation, which is the process of rescaling all variables so that they have a mean of zero and variance equal to one. This allows for a justified default centering at zero for the prior parameter distributions. In the cases with more than one explanatory variable, standardising them also ensures a common scale without distorting the differences in the range of their values (see Zheng and Casari (2018) for more details on feature engineering for machine learning). For the motorcycle data set, both the response and the explanatory variable are therefore standardised.

The exploratory analysis of the data revealed that there are likely to be four to five experts present in the HME tree. For simplicity, a tree architecture with four experts is pre-set. Given the previously noted potentially varying levels of abruptness in the separation between the experts, the following prior distribution is selected for the gating parameters

$$\boldsymbol{\gamma}^{(G,H)} \sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 50 & 0 \\ 0 & 100 \end{bmatrix}\right),$$

for all $(G, H) \in P_E$. This prior also reflects the anticipated higher variability in the slope parameter of the logistic regression. Taking into account the forseeable variability present across the potential individual expert fitted regression lines, wide weakly informative priors are chosen for the intercept and slope expert parameters. A moderate prior is set for the expert variance parameter in order to encourage a tighter allocation to experts. The NIG prior is thus chosen as

$$\boldsymbol{\beta}_E, \sigma_E^2 \sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, 1, 0.1\right)$$

for all $E \in \mathcal{E}$. The above prior distributions are used for all HME models fitted to the motorcycle accident data throughout this thesis.

For this illustration, consider Sampler I with three deliberately unhelpful starting points obtained by repeating the following three times:

1. Pre-set an architecture with 4 experts.
2. Randomly allocate observations to the experts in the tree.

3. Draw the gating parameters $\boldsymbol{\gamma}^{(G)} \sim \mathcal{N}\begin{pmatrix} 0 & 100^2 \\ 100^2 & 0 \end{pmatrix}$ for all $G \in \mathcal{G}$.

4. Draw the Gaussian expert parameters $\boldsymbol{\beta}_E \sim \mathcal{N}\begin{pmatrix} 0 & 25 \\ 25 & 0 \end{pmatrix}$ and $\log \sigma_E^2 \sim \mathcal{N}(0, 25)$ for all $E \in \mathcal{E}$.

Each of the resulting three states is then used to initialise an MCMC chain, which is run for $5,000$ iterations with the first 500 predictions discarded as burn-in, leaving $4,500$ iterations. The three chains are then used to assess if the target posterior distribution has been fully explored. A formal assessment of convergence is undertaken using the Gelman-Rubin convergence diagnostic. The potential scale reduction factor (PSRF) value of 5.02 is achieved hence strongly suggesting that convergence has not been reached.

To better understand this result, the predictions obtained from the three chains, shown in Figure 5.5, are investigated next. In the figure, each colour represents one of the three starting points. It is clear that neither of the three sets of starting parameter values led to an appropriate average fit. In fact, it is evident that the MCMC runs have not explored all modes of the distribution hence indicating poor mixing.

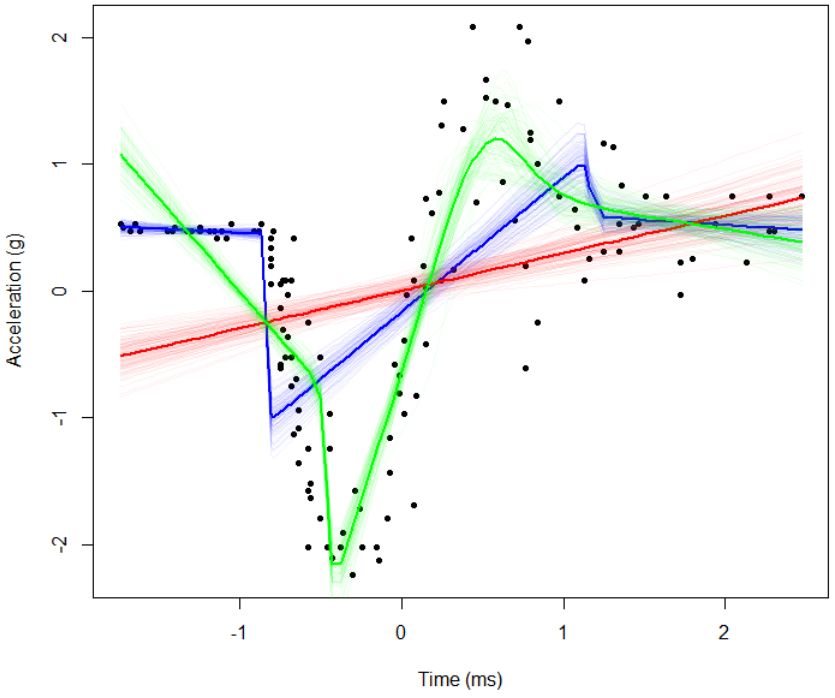


FIGURE 5.5: MCMC predictions for motorcycle accident data with three unhelpful starting points reflected by colour. The thick lines represent the average predictions while the thin lines represent every 10-th prediction.

It is, however, interesting to note that the individual prediction lines appear to be consistent for each of the three colours. In fact, investigating the trace plots of predictions would lead one to believe that convergence has been reached. For instance, consider the randomly selected 49-th observation, which is highlighted in red in Figure 5.6 (i). Plots (ii), (iii) and (iv) of Figure 5.6 show the trace plots of the predictions for the 49-th point across the MCMC iterations after accounting for burn-in. It is evident that all three trace plots indeed resemble *hairy caterpillars* and, if looked at individually, would suggest that convergence has been achieved. The thick lines in the trace plots represent the means of predictions and their colours distinguish the three initial MCMC starting points consistent with Figure 5.5. It is evident that all three chains yield vastly different mean predictions with the blue starting point resulting in -0.778, the red one in -0.171, and the green one in -0.628.

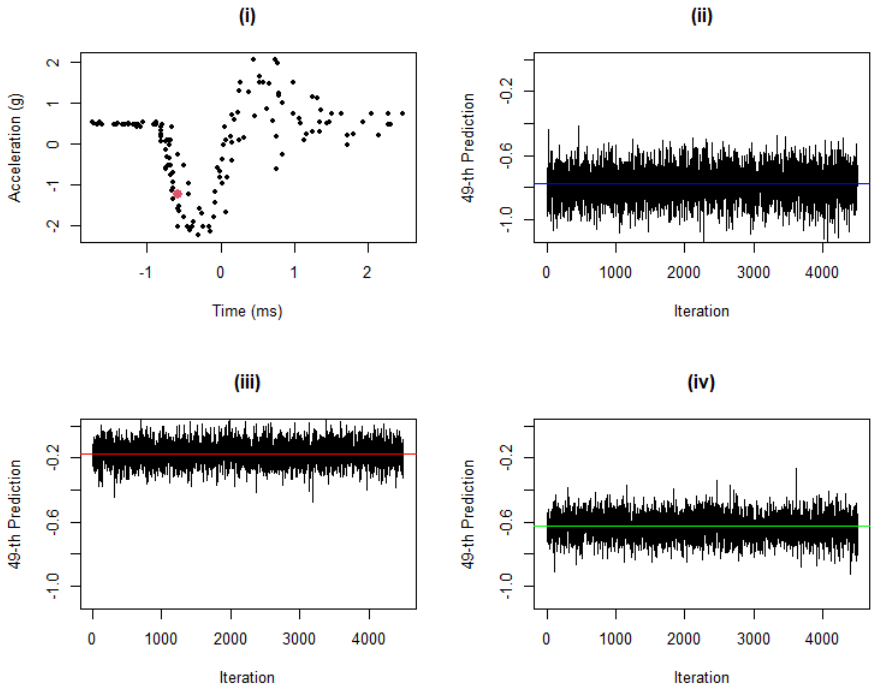


FIGURE 5.6: (i) Standardised motorcycle accident data with the 49-th point highlighted in red; (ii), (iii) and (iv) depict the trace plots of the 49-th point predictions across MCMC iterations after accounting for burn-in with the thick lines corresponding to the mean of predictions and the colour of the lines corresponding to the initial start of the chain consistent with Figure 5.5.

Using this deliberately challenging illustration, it has thus been shown that the HME models can suffer from both poor mixing and poor convergence, which in turn lead to an unreliable and inconsistent model fit. This thesis aims to provide a potential solution that works even in challenging circumstances such as discussed in this section. A flexible method that selects the tree architecture automatically could improve mixing

and make it more likely that MCMC chains converge irrespective of the initial parameter values. Such method should allow for escaping previously made unfortunate decisions and starting the model fitting process over if needed. The development of such a method would thus enable wider use of HME models. The next two chapters propose and evaluate such automatic architecture selection methods for HME models.

Chapter 6

Automatic Architecture Selection

6.1 Introduction to Automatic Architecture Selection for HME Models

Architecture selection for hierarchical mixture of experts (HME) models is the process of choosing the total number of expert and gate nodes as well as how these nodes are arranged in the model tree. Pre-setting model architecture poses some well-known challenges. Selecting a simple model with a small number of nodes can result in underfitting while choosing a complex model with too many nodes can lead to overfitting. As shown in the previous chapter, fixing the model architecture also requires setting the initial model parameter values, which can result in poor mixing and issues with convergence. The latter challenges could be tackled by growing trees during a model fitting process in contrast to selecting the architecture in advance.

The area of architecture selection for the hierarchical mixture of experts models has been investigated by Fritsch et al. (1996). One approach proposed therein consisted of determining the expert with the lowest likelihood and replacing it with a new gate and new experts. The gate weights are then allocated randomly while the expert parameters are inherited from the original expert and modified with an addition of noise. The method relies on random drawing of the gating parameters, which in turn means that the proposed state is not fully informed by the data available. Inheriting expert parameters may also cause issues in cases where the newly formed experts represent vastly different relationships between covariates and response. Further relevant research was undertaken in the area of Bayesian hierarchical clustering trees, which share some similarities with the hierarchical mixture of experts models in terms of the model architecture. The difference between the two lies in the leaves of the trees. In the usual clustering trees setting, the goal is to group the observations that are more similar to one another, while

the leaves of hierarchical mixture of experts models perform the task of fitting statistical models to the partitioned data. Thus some methods of creating splits and merges might be of interest when considering hierarchical mixtures of experts. In the case of Bayesian clustering trees, Heller and Ghahramani (2005) applies Bayesian hypothesis testing to decide which merges were advantageous as well as to choose the recommended depth of a binary tree. The problem is then further extended beyond a binary tree by Blundell et al. (2010). Generating trees with unbounded width and depth is discussed in Adams et al. (2010), where a tree-structured stick-breaking process is presented. The aim of this thesis is to contribute to the conducted research by proposing a Bayesian automatic architecture selection method for the HME models, which allows for both growing and pruning the trees as well as adjusting the existing architecture of the tree.

Consider two proposed ways of modifying the architecture of an HME model. The first approach, Type A, involves splitting and merging terminal nodes of the tree hence changing the total number of nodes in the model. This type of architecture selection can be thought of as operating in the leaves of the tree with the primary purpose of growing and pruning the tree. The second way of adjusting the tree architecture, Type B, involves swapping the existing nodes. This approach does not change the total number of nodes in the model. The success of the two types of architecture selection methods is somewhat dependent on each other. While Type A is essential to creating a tree and can be used on its own, it is not well-equipped to efficiently change the decisions made during the tree-growing process, particularly those high up in the tree. The latter downfall of Type A is perfectly matched by the sole purpose of Type B, which is reconsidering the order of the nodes in the tree. On the other hand, Type B can only operate if there are nodes available to swap, which are in turn created by Type A. Using the two types of architecture selection has the potential to increase the exploration of the space and hence improve the previously recorded poor mixing and convergence issues. Such flexible methods for automatic architecture selection would allow to propose and consider models which would have been missed otherwise as well as escape unfortunate previously made decisions, including an unhelpful initial state.

This chapter proposes and evaluates Type A architecture selection methods, which are designed to overcome the main challenges posed by the nature of the HME models. A novel approach for Type B of architecture selection is then presented and discussed in Chapter 7.

Type A of architecture selection is implemented using an adaptation of the *Reversible Jump* (RJ) algorithm. Some background on the algorithm is presented in Section 6.2 followed by an illustration of splits and merges in the HME model tree in Section 6.3. Next, the general framework for the RJ algorithm is outlined in Section 6.4 before discussing

the specifics of adapting the algorithm to the HME models in Section 6.5. The algorithm steps are then further specified for a binary tree in Section 6.5.1 before discussing the model size prior and methods of choosing which nodes to split or merge in Sections 6.5.3 and 6.5.4. The latter sections lead to outlining the step-by-step reversible jump algorithm for the HME models with normal experts in Section 6.6. The main challenge posed by the RJ algorithm is discussed and tackled in Section 6.5.2 and the proposed method is then evaluated in Section 6.7. All of the above is then applied to a real-life data set in Section 6.8, where the proposed methods are interrogated further. Finally, a tool for an interactive visualisation of the model fitting process is showcased in Section 6.8.5. The effects of the frequency and number of reversible jumps are then examined in Section 6.8.6 before making final remarks in Section 6.9.

6.2 Introduction to Reversible Jump

When proposing a split or a merge in an HME model, one is simply solving a model selection problem within the MCMC chain. The difficulty arises when comparing models with a different number of parameters. There is a need for a flexible and constructive way to jump between the plausible models. The latter can be achieved by the reversible jump (RJ) algorithm for the construction of reversible Markov chain samplers first introduced by Green (1995). The general appeal of the reversible jump lies in the natural generalisation of existing Markov chain methods (Sisson, 2005). The algorithm, which allows for exploring the sample space within a fixed dimension as well as making changes in dimensionality, is an extension of the widely used Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953). Another benefit of the reversible jump is the improved mixing caused by the new state proposals exploring larger spaces than the MH algorithm (Brooks et al., 2003).

The biggest challenge posed by the reversible jump algorithm is the typically low acceptance rate, which is often caused by uninformed jumps (Al-Awadhi et al., 2004; Ehlers and P. Brooks, 2008; Farr et al., 2015; Brooks et al., 2003). This issue prevents the use of the algorithm to its full potential. Developing informed proposals for hierarchical mixture of experts models includes several important decisions along the way. For example, a forward jump, or a split, requires proposing the location of the split, gating parameters, expert parameters, and allocation variables. An informed method for proposing all of the above would have the potential of increasing the acceptance rates of the reversible jump and thus improving mixing.

Research conducted on the topic of developing proposals for the reversible jump includes using an additional Markov chain to adjust the proposed moves (Al-Awadhi et al., 2004).

An adaptive proposal constructor, which optimises performance at each iteration is introduced in Ehlers and P. Brooks (2008). Farr et al. (2015) discusses an interpolation technique, which improves the proposals by implementing inter-model jumps. Further illustrations of proposal development procedures can be found in Brooks et al. (2003).

Generally, the Reversible Jump Markov Chain Monte Carlo (RJ MCMC) is a widely used method across a range of applications including inferring the elastic and petrophysical properties from pre-stack seismic data (Aleardi and Salusti, 2020), which also applies the delayed rejection updating scheme to speed up the convergence of the algorithm (Green and Mira, 2001). Keith et al. (2004) introduce a generalised Markov sampler, which is a mixture of several samplers, including the reversible jump. Jasra et al. (2007) present an extension of the population-based MCMC to the transdimensional case combining several methods, including the RJ algorithm. Waagepetersen and Sorensen (2001) provide a tutorial on the derivation of the reversible jump algorithm with application in genetics. The reversible jump, amongst others, is further discussed in Green and Mira (2001), Green (2003), Green and Hastie (2009) and Hastie and Green (2012).

Provided there is a carefully chosen proposal state, the reversible jump algorithm has the potential to improve the fitting and mixing of HME models via the introduction of automatic architecture selection. This chapter first illustrates the idea using a simple five expert hierarchical mixture model architecture.

6.3 Illustration of Reversible Jump

The reversible jump framework can be used when proposing splits and merges in hierarchical mixture of experts model. The proposed method operates in the leaves of the tree. That is, in the simplest binary tree case, the forward jump splits one expert into two, while the backward jump merges two experts into one. To ensure the reversibility of the jumps, only experts that have the same parent can be merged. The forward jump requires one to decide on where to *cut* the space and to introduce a new gate with two child experts to replace the original expert. The observations in the original expert then have to be reallocated to the newly formed experts. The backward jump in turn reverses the effects of the forward jump. A new expert, which absorbs the two original sibling experts, is introduced as a replacement for the parent gate. All observations are then assigned to the newly formed expert. It is evident that the number of model parameters changes with every jump, i.e., a forward jump, or a split, increases, while a backward jump, or a merge, decreases the number of parameters in the model. The reversible jump framework is well suited for undertaking such transdimensional jumps. Consider the illustration of the HME model with five experts as shown in Figure 6.1.

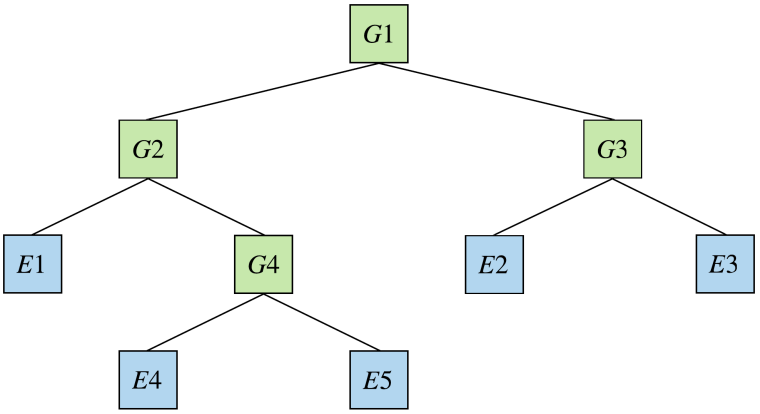


FIGURE 6.1: Illustration of an HME model with five experts equivalent to Figure 1.1.

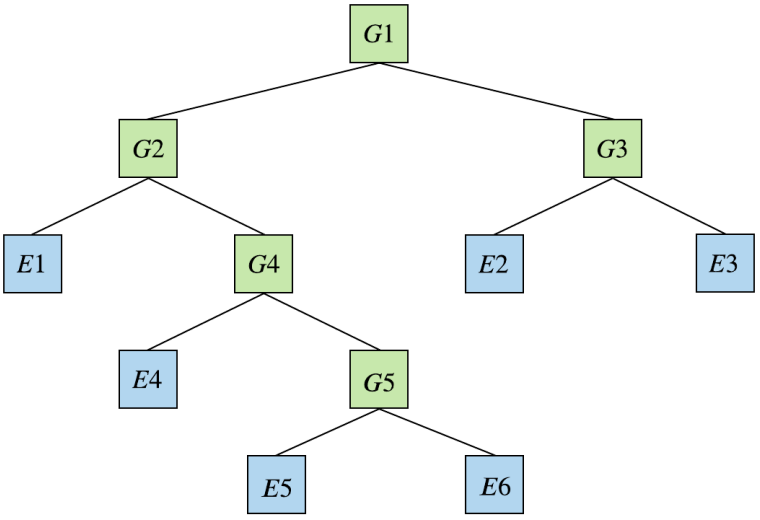


FIGURE 6.2: Illustration of an HME model with six experts. Split of $E5$ from Figure 6.1 into $E5$ and $E6$.

Let's say that one is interested in proposing a split of expert $E5$ into two new experts (forward jump). The resulting architecture is shown in Figure 6.2. It can be seen that a new gate $G5$ is introduced and it performs the task of partitioning the space further. Two new experts $E5$ and $E6$ are now formed and are children of $G5$. A backward jump would undo the effects of the forward jump, i.e. merge experts $E5$ and $E6$ from Figure 6.2 into one expert $E5$ from Figure 6.1. In the backward jump, the gate $G5$ is discarded and replaced by the newly formed expert. Such jumps can be performed as additional steps in the MCMC chain thus performing the automatic tree growth. Next, the steps of the reversible jump are outlined more formally.

6.4 General Framework and Algorithm for Reversible Jump

The general idea of the reversible jump allows for jumps between several plausible models with potentially different parameter dimensionalities. Assume we have a finite and countable set of models. Let each model have a density function $f_M(\boldsymbol{\theta})$ with the associated parameter vector $\boldsymbol{\theta}$ belonging to parameter space Θ. Further denote the transformation between the set of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ by T. For example, if we are performing a transformation from M parameters to M^* parameters, we write $\boldsymbol{\theta}^* = T_{M\to M^*}(\boldsymbol{\theta})$. The reversible jump algorithm starts with an initial model $M^{[0]}$ and initial parameter values for that model $\boldsymbol{\theta}^{[0]}$. Iterate for $t = 1, 2, ..., T$:

1. Update parameters of the current model using a sampler of your choice.
2. Select model M^* with probability $P(M^{[t-1]} \to M^*) = p_{M^{[t-1]}\to M^*}$.
3. Generate $\Delta^{[t-1]} \sim f_{M^{[t-1]}\to M^*}$.
4. Use T to transform between the parameters of the current and the proposed models, i.e. set $(\boldsymbol{\theta}^*, \Delta^*) = T_{M^{[t-1]}\to M^*}\left(\boldsymbol{\theta}^{[t-1]}, \Delta^{[t-1]}\right)$.
5. Compute the acceptance probability

$$\alpha := \min\left(1, \frac{f^{post}_{M^*}(\boldsymbol{\theta}^*) \times p_{M^*\to M^{[t-1]}} \times f_{M^*\to M^{[t-1]}}(\Delta^*)}{f^{post}_{M^{[t-1]}}\left(\boldsymbol{\theta}^{[t-1]}\right) \times p_{M^{[t-1]}\to M^*} \times f_{M^{[t-1]}\to M^*}\left(\Delta^{[t-1]}\right)} \times \left|\frac{\partial T_{M^{[t-1]}\to M^*}\left(\boldsymbol{\theta}^{[t-1]}, \Delta^{[t-1]}\right)}{\partial\left(\boldsymbol{\theta}^{[t-1]}, \Delta^{[t-1]}\right)}\right|\right), \tag{6.1}$$

   where $f^{post}_M(\boldsymbol{\theta})$ is the posterior distribution for the model M with parameter vector $\boldsymbol{\theta}$.
6. With probability α set $M^{[t]} = M^*$ and $\boldsymbol{\theta}^{[t]} = \boldsymbol{\theta}^*$, otherwise set $M^{[t]} = M^{[t-1]}$ and $\boldsymbol{\theta}^{[t]} = \boldsymbol{\theta}^{[t-1]}$.

The above steps outline a general case of the reversible jump. Next, the algorithm is discussed in the context of hierarchical mixture of experts models.

6.5 Reversible Jump for HME Models

6.5.1 Competing Models in a Binary HME Tree

Assume one is interested in splitting expert E' into two new experts E^* and E^{**}. From this point onwards, only the observations that have reached expert E' are considered.

Further denote the new gate parent of experts E^* and E^{**} by G^* with associated gating parameters $\boldsymbol{\gamma}^{(G^*,E^*)} = \mathbf{0}$ and $\boldsymbol{\gamma}^{(G^*,E^{**})}$. For ease of notation, for the remainder of the binary tree case discussion, denote $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(G^*,E^{**})}$. In this scenario, two competing models are considered at the depth of E'. The first model M_1 with no split and the density function

$$f_{M_1}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{M_1}\right) = f^{(E')}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E')}\right), \tag{6.2}$$

and the second model M_2 with the split and density function

$$f_{M_2}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{M_2}\right) = \sum_{E\in(E^*,E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right), \tag{6.3}$$

where $\pi_i^{(E)}$ are the mixing proportions obtained using the gating parameters of the parent gate G^*. In equations (6.2) and (6.3) $\boldsymbol{\theta}^{M_1}$ and $\boldsymbol{\theta}^{M_2}$ denote the parameters of the models M_1 and M_2 respectively. In this case, $\boldsymbol{\theta}^{M_1} = \left(\boldsymbol{\theta}^{(E')}\right)^T$ and $\boldsymbol{\theta}^{M_2} = \left(\boldsymbol{\theta}^{(E^*)}, \boldsymbol{\theta}^{(E^{**})}, \boldsymbol{\gamma}\right)^T$. As before, let $E(i)$ denote the expert to which the i-th observation is assigned, the complete data density for M_2 can then be written as [1]

$$f_{M_2}^{(c)}\left(y_i, \mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}^{M_2}\right) = \prod_{i\in(E^*,E^{**})} \pi_i^{(E(i))} f^{(E(i))}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E(i))}\right).$$

Let $f_M^{post}(\boldsymbol{\theta})$ denote the posterior distribution function for model M with associated model parameters $\boldsymbol{\theta}$. Following notation introduced in Section 6.4, denote the transformation between the set of parameters $\boldsymbol{\theta}^{M_1}$ and $\boldsymbol{\theta}^{M_2}$ by T. For example, if we are performing a transformation from M_1 parameters to M_2 parameters, we write $\boldsymbol{\theta}^{M_2} = T_{M_1\to M_2}(\boldsymbol{\theta}^{M_1})$. The reversible jump algorithm can then be implemented as per Section 6.4. For a special case of normal experts, which is the focus of this thesis, the steps of forward and backward reversible jumps are derived in Section 6.6.2. Next, an algorithm for developing efficient split proposals is presented.

6.5.2 Developing Efficient Proposals for the Forward Jump

The acceptance rate of the forward jumps in the RJ algorithm for HME models depends on the suitability of proposals for the new state. The RJ algorithm is well-known for low acceptance rates arising from randomly generated jumps (Al-Awadhi et al., 2004; Ehlers

[1] The notation $i \in (E^*, E^{**})$ is used as a shorthand for the i for which $z_i^{(E^*)} = 1$ or $z_i^{(E^{**})} = 1$.

and P. Brooks, 2008; Farr et al., 2015; Brooks et al., 2003). As discussed previously, a forward jump, or a split, requires one to pick a location of the split as well as propose the gating parameters for the newly formed gate. In the case of a backward jump, there is no addition of a new gate node and hence there is no need for developing a gating parameter proposal. In this section, a method of making intelligent forward jump gating parameter proposals for HME models is proposed.

The proposed method is illustrated in a two-dimensional setting and shown in Figure 6.3. Firstly, a point x^* is selected at random from the points allocated to the expert one wants to split (circled by a red dashed line). Meanwhile, a gating slope parameter, denoted γ_1, is drawn from chosen proposal distribution. The chosen point x^* and the gating slope parameter γ_1 are then used to infer the initial gating intercept parameter γ_0^*. The mixing proportions obtained using $\boldsymbol{\gamma} = (\gamma_0^*, \gamma_1)^T$ are shown as a dashed line in the bottom plot. Finally, some noise is added to the inferred intercept parameter resulting in γ_0. The final proposed mixing proportions obtained using $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)^T$ are then shown as a solid line in the bottom plot.

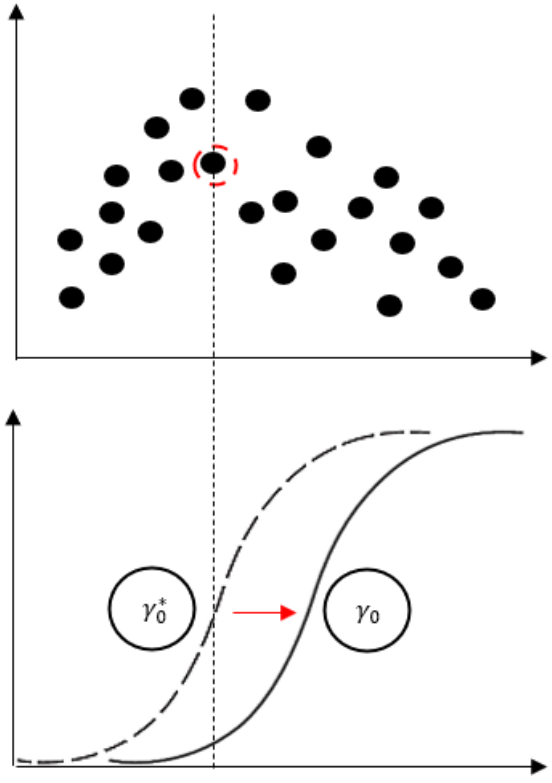


FIGURE 6.3: Illustration of the forward jump proposal algorithm for the reversible jump MCMC.

Forward Jump Proposal Generation Algorithm for the New Gating Parameters

More formally, identify the points that have reached the expert one wants to split, i.e. $i_1, ..., i_{n^*}$. Denote the total number of points in the set by n^*. The steps of the proposed algorithm are then as follows

1. Draw $\mathbf{x}^*$ from all $\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_{n^*}}$ at random. It then follows that $q(\mathbf{x}^*) = \frac{1}{n^*}$.
2. Draw $\boldsymbol{\gamma}_1 \sim \mathsf{MVN}(\boldsymbol{\mu}_{\boldsymbol{\gamma}_1}, \Sigma_{\boldsymbol{\gamma}_1})$.
3. Calculate $\gamma_0^* = -\mathbf{x}^{*T}\boldsymbol{\gamma}_1$.
4. Draw $\epsilon \sim \mathsf{N}(0, \sigma_\epsilon^2)$.
5. Set $\gamma_0 = \gamma_0^* + \epsilon$.

It can be shown that the joint probability density function of the proposed gating parameter $\boldsymbol{\gamma}=(\gamma_0, \boldsymbol{\gamma}_1)^T$ can be written as

$$q(\boldsymbol{\gamma}) = \sum_{i=i_1}^{i_{n^*}} \frac{1}{n^*} \times \phi_{\boldsymbol{\mu}_{\boldsymbol{\gamma}|\mathbf{x}_i}, \Sigma_{\boldsymbol{\gamma}|\mathbf{x}_i}}(\boldsymbol{\gamma}), \tag{6.4}$$

where $\phi_{\boldsymbol{\mu},\Sigma}(\cdot)$ is the multivariate Gaussian density function with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix Σ (please refer to the Appendix C for the details of the derivation).

It is evident that one is required to select several parameters for the proposal. Firstly, the mean, $\boldsymbol{\mu}_{\boldsymbol{\gamma}_1}$, and variance-covariance matrix, $\Sigma_{\boldsymbol{\gamma}_1}$, for the proposal distribution of the logistic regression slopes $\boldsymbol{\gamma}_1$ are to be chosen. The latter can be guided by the initial exploration of the data. For example, for a centered-at-zero proposal with a larger variance would be more likely to capture varying abruptness in the separations between potential experts. Similarly, smoother transitions between the experts would be explored with smaller values of $\Sigma_{\boldsymbol{\gamma}_1}$. The second proposal parameter to be selected is the variance for the noise added to the logistic regression intercept, σ_ϵ^2. It is encouraged to keep this value small in order to not move too far away from the inferred intercept whilst still introducing a random element to the proposal.

The forward jump proposal generation algorithm offers an alternative to the naive approach, which is based on uninformed random draws of the gating parameters. By selecting a reference point from the existing observations in the initial step 1 of the algorithm, one uses information provided by the observed data. This information is then

intelligently combined with the elements of a standard proposal approach, which is used in step 2. The latter development is thus expected to lead to more favorable forward jump proposals. The proposed forward jump generation algorithm is formally evaluated on an HME model with Gaussian experts in Section 6.7.

Next, the incorporation of prior beliefs on the size of an HME model tree is discussed.

6.5.3 Model Size Prior for HME Models

As before, let E denote an expert from the set of all experts in the model, $\mathcal{E}$. Further denote the number of elements in the set $\mathcal{E}$ as $N_{\mathcal{E}}$. One might wish to impose a prior on the total number of experts in an HME model. The proposed prior on the tree size can be written as follows:

$$f(N_{\mathcal{E}}) = \frac{\lambda^{N_{\mathcal{E}}} \exp(-\lambda)}{N_{\mathcal{E}}!},$$

i.e., $N_{\mathcal{E}} \sim \mathsf{Poi}(\lambda)$. Let us further denote the set of all experts after the proposed RJ step as $\mathcal{E}^*$ and let $N_{\mathcal{E}^*}$ denote the number of elements in the set. The effect of the model size prior on the acceptance probability of the RJ jump can then be quantified as

$$\frac{f(N_{\mathcal{E}^*})}{f(N_{\mathcal{E}})} = \frac{\frac{\lambda^{N_{\mathcal{E}^*}} \exp(-\lambda)}{N_{\mathcal{E}^*}!}}{\frac{\lambda^{N_{\mathcal{E}}} \exp(-\lambda)}{N!}} = \frac{N_{\mathcal{E}}!}{N_{\mathcal{E}^*}!} \lambda^{(N_{\mathcal{E}^*} - N_{\mathcal{E}})}. \tag{6.5}$$

In a case of a binary tree, $N_{\mathcal{E}^*}$ can be either $N_{\mathcal{E}^*} = N_{\mathcal{E}} + 1$ (after split) or $N_{\mathcal{E}^*} = N_{\mathcal{E}} - 1$ (after merge). For the two cases, (6.5) can then be written as

$$\frac{f(N_{\mathcal{E}^*})}{f(N_{\mathcal{E}})} = \begin{cases} \frac{\lambda}{N_{\mathcal{E}^*}}, & \text{if } N_{\mathcal{E}^*} = N_{\mathcal{E}} + 1, \\ \frac{N_{\mathcal{E}}}{\lambda}, & \text{if } N_{\mathcal{E}^*} = N_{\mathcal{E}} - 1. \end{cases} \tag{6.6}$$

Imposing a prior on the total number of experts in the model is a way of controlling the overall size of the tree. The latter might be valuable when it comes to avoiding overfitting/underfitting. Selecting which experts to split and which ones to merge is discussed next.

6.5.4 Choosing Experts to Split and Merge

Having outlined the theory for the RJ algorithm, the methodology for selecting experts to be split and merged is to be developed. Following the divide and conquer strategy, an

assumption is made that experts containing fewer observations are less likely to benefit from a split compared to a merge. Similarly, experts with more observations are assumed to be less likely to benefit from a merge compared to a split.

In a binary tree case, any expert can be split into two experts. Consider splitting expert E and let $n_E = \sum_i z_i^{(E)}$ denote the total number of observations in E. The proposed probability of choosing expert E to split is then proportional to

$$p^{split}(E) \propto \frac{n_E + \delta}{N_{\mathcal{E}}}, \tag{6.7}$$

where δ is a very small constant and $N_{\mathcal{E}}$ is the number of elements/experts in the set $\mathcal{E}$.

In a binary tree case, any two sibling experts can be merged into one expert. Denote the set of all gates which are parents to two experts as $\mathcal{G}^*$. Further denote the total number of elements in $\mathcal{G}^*$ as $N_{\mathcal{G}^*}$. Consider merging the kids of gate G^* from $\mathcal{G}^*$ and let $n_{G^*} = \sum_{E \in \mathcal{P}_>^{(G^*)}} \sum_i z_i^{(E)}$ denote the total number of observations assigned to the children of G^*. The proposed probability of choosing children of G^* to merge is then proportional to

$$p^{merge}(G^*) \propto \frac{1}{n_{G^*} + \delta}, \tag{6.8}$$

where δ is a very small constant.

It is important to note that the quantities stated in (6.7) and (6.8) must also be calculated for the jump in the opposite direction of the proposed jump.

The addition of informed expert selection for reversible jumps was also noted to encourage the merging of empty experts. Given (6.8), the gates with a smaller number of observations assigned to their children have a higher probability of being picked for a merge. Thus, gates with empty experts as children are more likely to be selected. Eliminating empty experts improves on the overall fit of the tree since such experts do not have an effect on the overall likelihood of the tree and are thus are an unnecessary complication of the model architecture. On the other hand, (6.7) ensures a lower probability of picking an empty expert for a split thus avoiding wasting computational time.

This section concludes the theory and proposed extensions required for implementing the reversible jump for the HME models. Next, the special case of normal experts is discussed in detail and applied to simulated data as well as a real life application.

6.6 Reversible Jump for Normal Expert HME Models

6.6.1 Competing Models for Normal Expert HME Models

Using notation introduced in Section 6.5.1, further assume that experts E', E^* and E^{**} are normal experts. Let $\phi_{\mu,\sigma^2}(\cdot)$ denote the normal density with the mean μ and variance σ^2_E. In such case the density function of model M_1 with expert E' can be written as

$$\begin{aligned} f_{M_1}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{M_1}\right) &= f\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E')}\right) \\ &= f^{(E')}\left(y_i|\mathbf{x}_i, \boldsymbol{\beta}_{E'}, \sigma^2_{E'}\right) \\ &= \phi_{\mu_{iE'},\sigma^2_{E'}}(y_i), \end{aligned}$$

where $\boldsymbol{\theta}^{M_1} = \left(\boldsymbol{\theta}^{(E')}\right)^T = \left(\boldsymbol{\beta}_{E'}, \sigma^2_{E'}\right)^T$ and $\mu_{iE'} = \mathbf{x}_i^T\boldsymbol{\beta}_{E'}$. The corresponding likelihood function is

$$L_{M_1} = \prod_{i \in E'} \phi_{\mu_{iE'},\sigma^2_{E'}}(y_i).$$

Finally, the posterior distribution function for model M_1 with expert E' can be written as

$$f^{post}_{M_1}\left(\boldsymbol{\theta}^{M_1}\right) \propto L_{M_1} \times f(\boldsymbol{\beta}_{E'}) \times f\left(\sigma^2_{E'}\right), \tag{6.9}$$

where $f(\boldsymbol{\beta}_{E'})$ and $f\left(\sigma^2_{E'}\right)$ correspond to the expert parameter prior distribution density functions. Similarly, the density function for the second model M_2 with experts E^* and E^{**} can be written as

$$\begin{aligned} f_{M_2}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{M_2}\right) &= \sum_{E \in (E^*, E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right) \\ &= \sum_{E \in (E^*, E^{**})} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i, \boldsymbol{\beta}_E, \sigma^2_E\right) \\ &= \sum_{E \in (E^*, E^{**})} \pi_i^{(E)} \phi_{\mu_{iE},\sigma^2_E}(y_i), \end{aligned}$$

where $\boldsymbol{\theta}^{M_2} = \left(\boldsymbol{\theta}^{(E^*)}, \boldsymbol{\theta}^{(E^{**})}, \boldsymbol{\gamma}\right)^T = \left(\boldsymbol{\beta}_{E^*}, \sigma^2_{E^*}, \boldsymbol{\beta}_{E^{**}}, \sigma^2_{E^{**}}, \boldsymbol{\gamma}\right)^T$ and $\mu_{iE} = \mathbf{x}_i^T\boldsymbol{\beta}_E$ for $E \in (E^*, E^{**})$. Alternatively, if the allocation variables are known

$$
\begin{aligned}
f_{M_2}^{(c)}\left(y_i, \mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\theta}^{M_2}\right) &= \pi_i^{(E(i))} f^{(E(i))}\left(y_i | \mathbf{x}_i, \boldsymbol{\theta}^{(E(i))}\right) \\
&= \pi_i^{(E(i))} f^{(E(i))}\left(y_i | \mathbf{x}_i, \boldsymbol{\beta}_{E(i)}, \sigma^2_{E(i)}\right) \\
&= \pi_i^{(E(i))} \phi_{\mu_{E(i)}, \sigma^2_{E(i)}}(y_i),
\end{aligned}
$$

where $\mu_{E(i)} = \mathbf{x}_i^T \boldsymbol{\beta}_{E(i)}$ for $E \in (E^*, E^{**})$. In the case of known allocation variables, the corresponding likelihood function is

$$
L_{M_2} = \prod_{i \in (E^*, E^{**})} \pi_i^{(E(i))} \phi_{\mu_{E(i)}, \sigma^2_{E(i)}}(y_i).
$$

Finally, the posterior distribution function for model M_2 with experts E^* and E^{**} can be written as

$$
f_{M_2}^{post}\left(\boldsymbol{\theta}^{M_2}\right) \propto L_{M_2} \times f(\boldsymbol{\beta}_{E^*}) \times f(\sigma^2_{E^*}) \times f(\boldsymbol{\beta}_{E^{**}}) \times f\left(\sigma^2_{E^{**}}\right) \times f\left(\boldsymbol{\gamma}\right), \tag{6.10}
$$

where $f(\boldsymbol{\beta}_E)$ and $f\left(\sigma^2_E\right)$ for $E \in (E^*, E^{**})$ correspond to the expert parameter prior distribution density functions and $f\left(\boldsymbol{\gamma}\right)$ denotes the gating parameter prior distribution density function.

The steps of the reversible jump algorithm for a special case of normal experts are outlined next.

6.6.2 Forward and Backward Jumps for Normal Expert HME Models

In a special case of normal experts, the steps of the reversible jump algorithm, split by direction, are outlined in Sections 6.6.2.1 and 6.6.2.2.

6.6.2.1 Forward Jump (Split Move)

Current Model M_1, Proposed Model M_2

1. Update parameters of the current model M_1, i.e. $\boldsymbol{\theta}^{M_1} = (\boldsymbol{\beta}_{E'}, \sigma^2_{E'})^T$. Record the corresponding probability densities $q\left(\boldsymbol{\beta}_{E'}\right)$ and $q\left(\sigma^2_{E'}\right)$, respectively.

2. Identify the points that have reached expert E', i.e, $i \in E'$. Denote the total number of points that have reached E' by $n_{E'}$. All subsequent steps are applied to this subset of points only. [2]

3. Propose a gating parameter value $\boldsymbol{\gamma}$ for the new gate G^* as follows

   (a) Draw $\mathbf{x}^*$ from all $\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_{n_{E'}}}$ at random. It then follows that $q(\mathbf{x}^*) = \frac{1}{n_{E'}}$.

   (b) Draw $\boldsymbol{\gamma}_1 \sim \mathsf{MVN}(\boldsymbol{\mu}_{\boldsymbol{\gamma}_1}, \Sigma_{\boldsymbol{\gamma}_1})$.

   (c) Draw $\epsilon \sim \mathsf{N}(0, \sigma_\epsilon^2)$.

   (d) Set $\gamma_0 = -\mathbf{x}^{*T}\boldsymbol{\gamma}_1 + \epsilon$.

   (e) Evaluate the joint probability density function of the proposed gating parameter $\boldsymbol{\gamma}=(\gamma_0, \boldsymbol{\gamma}_1)^T$ as

   $$q(\boldsymbol{\gamma}) = \sum_{i=i_1}^{i_{n_{E'}}} \frac{1}{n} \times \phi_{\boldsymbol{\mu}_{\boldsymbol{\gamma}|\mathbf{x}_i}, \Sigma_{\boldsymbol{\gamma}|\mathbf{x}_i}}(\boldsymbol{\gamma}),$$

   where $\phi_{\boldsymbol{\mu},\Sigma}(\cdot)$ is the multivariate Gaussian density function with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix Σ.

4. Calculate the mixing proportions $\pi_i^{(E^*)}$ and $\pi_i^{(E^{**})}$ for $i \in E'$ as

   $$\pi_i^{(E^*)} = \frac{1}{1 + \exp\left(\mathbf{x}_i^T\boldsymbol{\gamma}\right)} \text{ and } \pi_i^{(E^{**})} = \frac{\exp\left(\mathbf{x}_i^T\boldsymbol{\gamma}\right)}{1 + \exp\left(\mathbf{x}_i^T\boldsymbol{\gamma}\right)}.$$

5. Assign each point $i \in E'$, i.e., set $z_i^{(E)} = 1$ and $z_i^{(E'')} = 0$ for $E'' \neq E$, to experts E^* and E^{**} with probabilities of $\pi_i^{(E^*)}$ and $\pi_i^{(E^{**})}$ respectively.

6. Given the allocations from the previous step, obtain the MLE estimates for the expert parameters of M_2, i.e. $\hat{\boldsymbol{\beta}}_{E^*}, \hat{\sigma}^2_{E^*}, \hat{\boldsymbol{\beta}}_{E^{**}}, \hat{\sigma}^2_{E^{**}}$.

7. Draw the new values

   $$\boldsymbol{\beta}_E \sim \mathsf{MVN}\left(\hat{\boldsymbol{\beta}}_E, \hat{\sigma}^2_E \left(X^{(E)^T} X^{(E)}\right)^{-1}\right),$$

   where $\hat{\boldsymbol{\beta}}_E$ is obtained in the previous step, $\hat{\sigma}^2_E \left(X^{(E)^T} X^{(E)}\right)^{-1}$ is the best linear unbiased estimator for $\mathrm{Var}(\boldsymbol{\beta}_E)$, and $X^{(E)}$ denotes a subset of the design matrix containing points that have reached expert E for $E \in (E^*, E^{**})$.

[2] Though steps 1 and 2 are not part of a transdimensional move, the proposals in the reverse step are identical to the regular Gibbs update of the expert parameters, thus the proposal densities from step 1 can be reused in the calculation of the acceptance probability of the reverse move.

8. Draw the new values σ_E^2 given $\boldsymbol{\beta}_E$ from the conditional Inverse-Gamma posterior distribution

$$\sigma_E^2 | X^{(E)}, \mathbf{y}^{(E)} \sim \mathsf{IG}\left(a_E^{(post)}, b_E^{(post)}\right),$$

where $\mathbf{y}^{(E)}$ is a subset of the response vector containing points that have reached expert E for $E \in (E^*, E^{**})$ and the posterior parameters $\left(a_E^{(post)}, b_E^{(post)}\right)$ are obtained as per (5.2).

9. Calculate the probability of drawing the new values of $\boldsymbol{\beta}_E$, for $E \in (E^*, E^{**})$ as

$$q\left(\boldsymbol{\beta}_E\right) = \phi_{\hat{\boldsymbol{\beta}}_E, \hat{\sigma}_E^2 \left(X^{(E)^T} X^{(E)}\right)^{-1}}(\boldsymbol{\beta}_E),$$

where $\phi_{\boldsymbol{\mu},\Sigma}(\cdot)$ is the multivariate Gaussian density function with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix Σ.

10. Calculate the probability of drawing the new values of σ_E^2, i.e. $q(\sigma_E^2)$, for $E \in (E^*, E^{**})$ by evaluating the appropriate Inverse-Gamma density function as per step 8.

11. Calculate the probability of drawing the chosen values for the allocation variables

$$q(\mathbf{z}) = \prod_{i \in (E^*, E^{**})} \pi_i^{(E(i))}.$$

12. Calculate the model size priors $f(N_{\mathcal{E}})$ and $f(N_{\mathcal{E}^*})$ for the total number of experts before and after the jump, respectively. For a binary forward jump, as per (6.6),

$$\frac{f(N_{\mathcal{E}^*})}{f(N_{\mathcal{E}})} = \frac{\lambda}{N_{\mathcal{E}^*}},$$

where the number of experts in the model $N_{\mathcal{E}}$ follows a Poisson distribution with rate λ, i.e., $N_{\mathcal{E}} \sim \text{Poi}(\lambda)$.

13. Calculate $p^{split}(E')$, the probability of choosing to split E', as per (6.7) and $p^{merge}(G^*)$, the probability of merging the two children of G^*, i.e. E^* and E^{**}, as per (6.8).

14. Accept the forward jump with probability

$$\alpha = \min\left(1, \frac{f_{M_2}^{post}\left(\boldsymbol{\theta}^{M_2}\right) \times q\left(\boldsymbol{\beta}_{E'}\right) \times q\left(\sigma_{E'}^2\right) \times f(N_{\mathcal{E}^*}) \times p^{merge}(G^*)}{f_{M_1}^{post}\left(\boldsymbol{\theta}^{M_1}\right) \times q(\boldsymbol{\beta}_{E^*}) \times q(\boldsymbol{\beta}_{E^{**}}) \times q(\sigma_{E^*}^2) \times q(\sigma_{E^{**}}^2) \times q(\mathbf{z}) \times q(\boldsymbol{\gamma}) \times f(N_{\mathcal{E}}) \times p^{split}(E')}\right),$$

where $q(\boldsymbol{\gamma})$ is obtained as per step 3e, while $f_{M_1}^{post}(\boldsymbol{\theta}^{M_1})$ and $f_{M_2}^{post}\left(\boldsymbol{\theta}^{M_2}\right)$ are calculated as per (6.9) and (6.10), respectively.

6.6.2.2 Backward Jump (Merge Move)

Current Model M_2, Proposed Model M_1

1. Update parameters of the ccurrent model M_2, i.e. $\boldsymbol{\theta}^{M_2} = (\boldsymbol{\beta}_{E^*}, \sigma^2_{E^*}, \boldsymbol{\beta}_{E^{**}}, \sigma^2_{E^{**}}, \boldsymbol{\gamma})$. Record the corresponding probability densities $q(\boldsymbol{\gamma})$, $q(\mathbf{z})$, $q\left(\boldsymbol{\beta}_E\right)$, and $q\left(\sigma^2_E\right)$ for $E \in (E^*, E^{**})$.

2. Update allocations for all data points $i = 1, ..., n$ as follows

   (a) Calculate
   $$\alpha_i^{(E)} = \left(\prod_{(G,H)\in P_E} \pi_i^{(G,H)} \right) f^{(E)}\left(y_i | \mathbf{x}_i, \boldsymbol{\theta}^{(E)}\right),$$
   for all $E \in \mathcal{E}$.

   (b) Assign the i-th point to expert E, i.e., set $z_i^{(E)} = 1$ and $z_i^{(E'')} = 0$ for $E'' \neq E$, with probability $\alpha_i^{(E)}$ where $\sum_{E\in\mathcal{E}^*} \alpha_i^{(E)} = 1$.

3. Assign all points from E^* and E^{**} to the merged expert E'.

4. Given the allocations from the previous step, obtain the MLE estimates for the expert parameters of M_1, i.e. $\hat{\boldsymbol{\beta}}_{E'}, \hat{\sigma}^2_{E'}$.

5. Draw the new values
   $$\boldsymbol{\beta}_{E'} \sim \mathsf{MVN}\left(\hat{\boldsymbol{\beta}}_{E'}, \hat{\sigma}^2_{E'}\left(X^{(E')^T} X^{(E')}\right)^{-1}\right),$$
   where where $\hat{\boldsymbol{\beta}}_{E'}$ is obtained in the previous step, $\hat{\sigma}^2_{E'}\left(X^{(E')^T} X^{(E')}\right)^{-1}$ is the best linear unbiased estimator for $\text{Var}\left(\boldsymbol{\beta}_{E'}\right)$, and $X^{(E')}$ denotes a subset of the design matrix containing points that have reached expert E'.

6. Draw the new values $\sigma^2_{E'}$ given $\boldsymbol{\beta}_{E'}$ from the conditional Inverse-Gamma posterior distribution
   $$\sigma^2_{E'} | X^{(E')}, \mathbf{y}^{(E')} \sim \mathsf{IG}\left(a_{E'}^{(post)}, b_{E'}^{(post)}\right),$$
   where $\mathbf{y}^{(E')}$ is a subset of the response vector containing points that have reached expert E', and the posterior parameters $\left(a_{E'}^{(post)}, b_{E'}^{(post)}\right)$ are obtained as per (5.2).

7. Calculate the probability of obtaining the value of $\boldsymbol{\beta}_{E'}$ as

$$q(\boldsymbol{\beta}_{E'}) = \phi_{\hat{\boldsymbol{\beta}}_{E'},\hat{\sigma}^2_{E'}\left(X^{(E')^T}X^{(E')}\right)^{-1}}(\boldsymbol{\beta}_{E'}),$$

where $\phi_{\boldsymbol{\mu},\Sigma}(\cdot)$ is the multivariate Gaussian density function with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix Σ.

8. Calculate the probability density of the new value of $\sigma^2_{E'}$, i.e. $q\left(\sigma^2_{E'}\right)$ by evaluating the appropriate Inverse-Gamma density as per step 6.

9. Calculate the model size priors $f(N_{\mathcal{E}})$ and $f(N_{\mathcal{E}^*})$ for the total number of experts before and after the jump, respectively. For a binary backward jump, as per (6.6),

$$\frac{f(N_{\mathcal{E}^*})}{f(N_{\mathcal{E}})} = \frac{N_{\mathcal{E}}}{\lambda},$$

where the number of experts in the model $N_{\mathcal{E}}$ follows a Poisson distribution with rate λ, i.e., $N_{\mathcal{E}} \sim \text{Poi}(\lambda)$.

10. Calculate $p^{merge}(G^*)$, the probability of merging the two children of G^*, i.e. E^* and E^{**}, as per (6.8) and $p^{split}(E')$, the probability of choosing to split E', as per (6.7).

11. Accept the backward jump with probability

$$\alpha = \min\left(1, \frac{f^{post}_{M_1}\left(\boldsymbol{\theta}^{M_1}\right) \times q\left(\boldsymbol{\beta}_{E^*}\right) \times q\left(\sigma^2_{E^*}\right) \times q\left(\boldsymbol{\beta}_{E^{**}}\right) \times q\left(\sigma^2_{E^{**}}\right) \times q(\mathbf{z}) \times q(\boldsymbol{\gamma}) \times f(N_{\mathcal{E}^*}) \times p^{split}(E')}{f^{post}_{M_2}\left(\boldsymbol{\theta}^{M_2}\right) \times q(\boldsymbol{\beta}_{E'}) \times q(\sigma^2_{E'}) \times f(N_{\mathcal{E}}) \times p^{merge}(G^*)}\right),$$

where $f^{post}_{M_1}(\boldsymbol{\theta}^{M_1})$ and $f^{post}_{M_2}\left(\boldsymbol{\theta}^{M_2}\right)$ are calculated as per (6.9) and (6.10), respectively.

The steps outlined above can now be applied to automatically grow HME models. Before the full implementation is carried out, the reversible jump proposal generation algorithm for HME model with normal experts is evaluated first.

6.7 Evaluation of the Reversible Jump Proposal Generation Algorithm

The reversible jump proposal generation algorithm, outlined in the previous sections, is evaluated on simulated data shown in Figure 6.4. The desired model fit is a hierarchical mixture of three normal experts, where the three experts $E1$, $E2$ and $E3$ are represented by black, red and green colours respectively. For the purpose of algorithm evaluation, an additional expert $E4$ is introduced and shown in blue.

It is evident that the three potential experts, as shown in Figure 6.4 (i), would be well separated as there are abrupt changes present in the relationship between the response and explanatory variable. The black and green experts are also anticipated to have notably different intercept and slope parameters when compared to the red expert. Finally, there is a low level of variability in the response present for each anticipated expert.

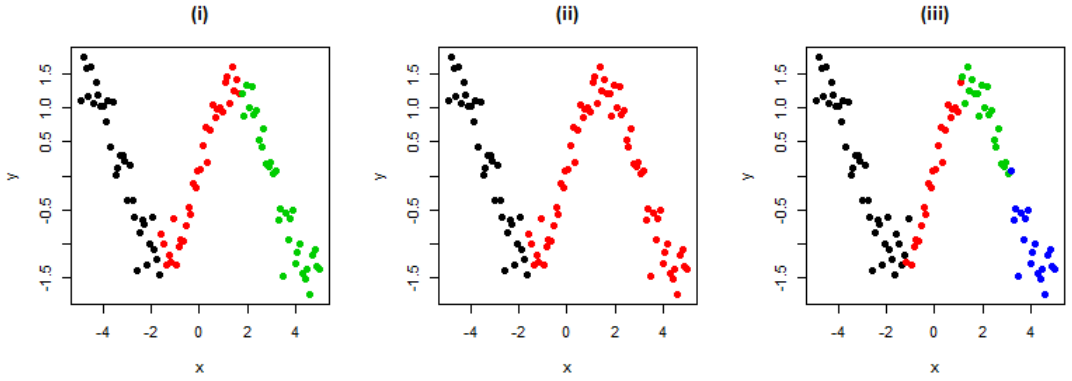


FIGURE 6.4: Hierarchical mixture of experts with (i) three; (ii) two; (iii) four experts for simulated data. Experts $E1$, $E2$, $E3$ and $E4$ represented by colours black, red, green and blue respectively.

The forward jump parameters are drawn from two Gaussian distributions ,i.e., $\gamma_1 \sim \mathsf{N}(0, 100)$, which ensures that the proposed slope parameters are steep enough, and $\epsilon \sim \mathsf{N}(0, 0.5)$, which introduces a random element to the proposed intercept parameter (see Section 6.5.2 for details). Given the anticipated abrupt separation between experts, the following weakly informative prior distribution is selected for the gating parameters

$$\boldsymbol{\gamma}^{(G,H)} \sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}\right),$$

for all $(G, H) \in P_E$. Taking into account the notable variability present across the potential individual expert fitted regression lines, wide weakly informative priors are

chosen for the intercept and slope expert parameters. A moderate prior is set for the expert variance parameter to reflect the anticipated tighter allocation to experts. The NIG prior is thus chosen as

$$\boldsymbol{\beta}_E, \sigma_E^2 \sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, 1, 0.1\right),$$

for all $E \in \mathcal{E}$.

The following scenarios are then considered:

Case A Proposing a split, where it should be accepted.

Scenario: split expert $E2$ (red) from Figure 6.4 (ii) into two new experts.

Case B Proposing a split, where it should be rejected.

Scenario: split expert $E3$ (green) from Figure 6.4 (i) into two new experts.

Case C Proposing a merge, where it should be accepted.

Scenario: merge experts $E3$ and $E4$ (green and blue) from Figure 6.4 (iii) into one expert.

Case D Proposing a merge, where it should be rejected.

Scenario: merge experts $E2$ and $E3$ (red and green) from Figure 6.4 (i) into one expert.

Each of the cases above is ran $1,000$ times and the occurrence of the desired outcome is recorded in Table 6.1. It can be seen that the algorithm has behaved as expected. The majority of beneficial splits and merges has been accepted (77.6% and 88.9% respectively) while only a small proportion of detrimental splits (7%) and none of undesired merges have been accepted. It is worth noting that this is a simple problem, which has been simulated to test and showcase the proposed method. The acceptance rates might not be as high when used on more complicated, multidimensional data sets.

TABLE 6.1: Results for the evaluation of the reversible jump proposal generation algorithm.

Case	**Proposal**	**Expected Acceptance**	**Recorded Acceptance**
A	Split	High	77.6%
B	Split	Low	7%
C	Merge	High	88.9%
D	Merge	Low	0%

It is of interest to look at the accepted forward jumps, or splits, to understand if the proposed algorithm is working as anticipated. Two examples of accepted jumps are shown in Figures 6.5 and 6.6.

Plots (i) in the corresponding figures depict the same initial state with two experts (black and red). Plots (ii) in turn show the fit straight after an accepted forward jump. It is important to note that the fitted lines shown in these plots are not fitted to the data, but those proposed by the reversible jump efficient proposal generation algorithm. The gate and expert parameters are then improved upon by multiple MCMC runs and the resulting regression lines and allocations are shown plots (iii). The colors present in plots (ii) and (iii) distinguish the three experts present in the HME tree post-split.

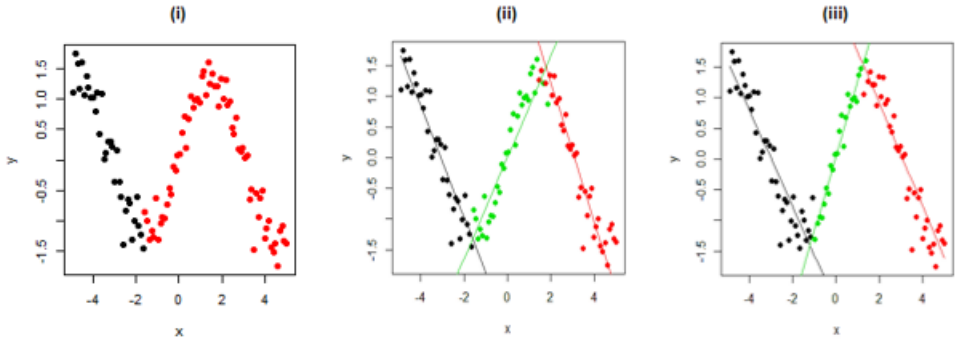


FIGURE 6.5: First example of accepted forward jump state from initial state shown in (i); Fit immediately after the jump shown in (ii); Fit after 100 MCMC runs shown in (iii). Experts $E1$, $E2$, $E3$ represented by colours black, red and green respectively.

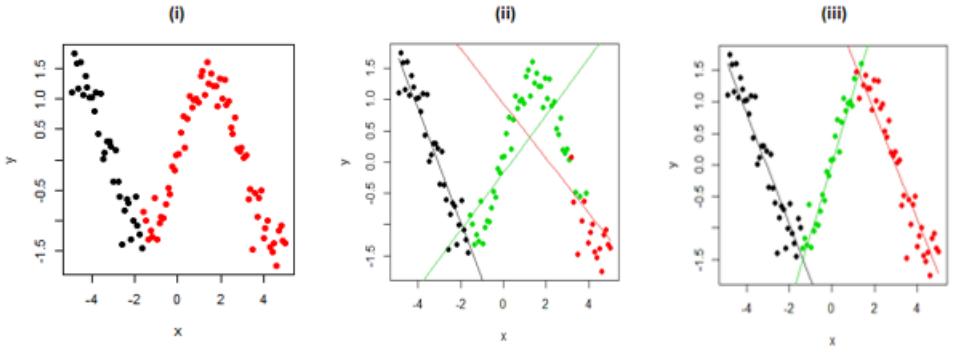


FIGURE 6.6: Second example of accepted forward jump state from initial state shown in (i); Fit immediately after the jump shown in (ii); Fit after 100 MCMC runs shown in (iii). Experts $E1$, $E2$, $E3$ represented by colours black, red and green respectively.

In the first example from Figure 6.5 (ii), the proposed split is a fortunate one, because the space is cut in the right region and the proposed expert parameters already fit the

data well. In the case of a second example from Figure 6.6 (ii), the initial cut appears to be away from the desired region and the proposed expert parameters do not seem to fit data well. However, this state is an excellent start for MCMC to then improve upon resulting in an excellent fit shown in Figure 6.6 (iii). The latter example highlights that the reversible jump proposal is not expected to provide a perfect solution to the problem, but to assist MCMC by proposing a reasonable start. Having shown that informed forward jumps work on a simple simulated example, the algorithm is next evaluated on a real-life data set.

6.8 Evaluation of the Reversible Jump MCMC on Motorcycle Accident Data

This section evaluates the performance of the RJ MCMC on the motorcycle accident data set. The naive approach, which does not incorporate the developments proposed throughout this chapter, is presented in Section 6.8.1 and is compared to the informed RJ methodology in Section 6.8.2. The previously introduced unhelpful intitial states case (see Section 5.5) is revisited and evaluated with respect to mixing and convergence in Section 6.8.3. In Section 6.8.4, we move away from starting the chain with a pre-set tree architecture and investigate automatic tree growth with RJ MCMC. The automatically grown tree is then visualised using an interactive `R-Shiny` application in Section 6.8.5. The effects of the frequency and number of reversible jumps are then interrogated in Section 6.8.6 before making final remarks in Section 6.9.

6.8.1 Naive Reversible Jump MCMC Results

The reversible jump MCMC for hierarchical mixture of experts model is evaluated on motorcycle accident data set (standardised data shown in Figure 6.7). The detailed exploratory analysis and description of the data is covered in Section 5.5.1. In a nutshell, it is evident that the motorcycle data is of a heteroscedastic nature with the variance of observations increasing as time increases. Moreover, there are a number of change points present in the data. These characteristics of the data make it a good candidate for showcasing the main strengths of HME models.

First, an informed HME model architecture is chosen to evaluate the performance of the naive RJ algorithm, which does not incorporate any of the developments proposed throughout this thesis. Following initial observations made in Section 5.5.1, an initial state consisting of 5 normal experts is chosen and shown in Figure 6.7. The initial

allocations are pre-set by visual inspection with initial model parameters set as the corresponding MLE estimates.

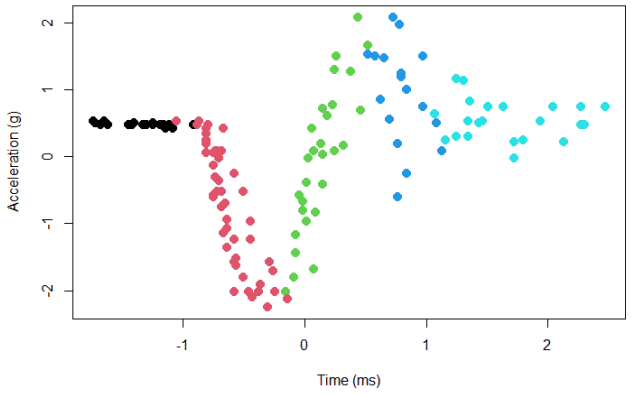


FIGURE 6.7: Initial informed allocations for evaluation of the naive reversible jump against the proposed method on standardised motorcycle accident data.

For this illustration, the RJ MCMC is run for $5,000$ iterations, with the first 500 runs discarded as burn-in. In all evaluations performed in this chapter, the convergence of the chains is assessed by visual inspection of the predictions as outlined in Section 4.3.4.2. A single reversible jump is proposed every 10 MCMC iterations, which serves as a starting point for the detailed frequency and number of jumps investigation undertaken in Section 6.8.6. The prior distributions used here are stated in Section 5.5.2 and applied to the motorcycle accident data throughout the thesis.

Naive RJ MCMC chain with non-informative jumps is set-up is as follows:

1. The direction of the jump is chosen at random.
2. Experts to split/merge are chosen at random.
3. In case of a split, the new gating parameters are drawn from

$$\boldsymbol{\gamma}^{(G,H)} \sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}\right),$$

   for all $(G, H) \in P_E$ and for all $E \in \mathcal{E}$. A wide proposal is chosen to allow for the observed abrupt separation between the potential experts.

4. The new expert parameters are drawn from

$$\boldsymbol{\beta}_E \sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}\right) \text{ and } \sigma_E^2 \sim \mathsf{IG}\left(1, 0.1\right),$$

for all $E \in \mathcal{E}$. A smaller variance for the proposal distribution of the intercept and slope is used here to ensure that the initial starting values are not too far from zero for the standardised data.

The predictions produced by the naive RJ MCMC algorithm are shown in Figure 6.8 while the associated acceptance rates are given in Table 6.2. It can be seen that the naive RJ MCMC was not able to escape three unfortunate merges resulting in the depicted state. Table 6.2 also reveals a low overall rate acceptance of 0.6%. It is evident that none of the proposed splits have been accepted, which is not a surprising result given the uninformed split proposal. From Figure 6.9, it is clear that the chain has spent the most time exploring models with two experts. Although the individual predictions (black lines) appear to be tightly clustered suggesting convergence has been achieved, it is evident that the chain has not explored all modes of the posterior distribution. These results are next compared to the methodology proposed in Section 6.5.

TABLE 6.2: Acceptance rates of the naive RJ algorithm jumps for the motorcycle accident data.

	Splits	**Merges**	**All Jumps**
Number Proposed	244	256	500
Number Accepted	0	3	3
Acceptance	0%	1.17%	0.6%

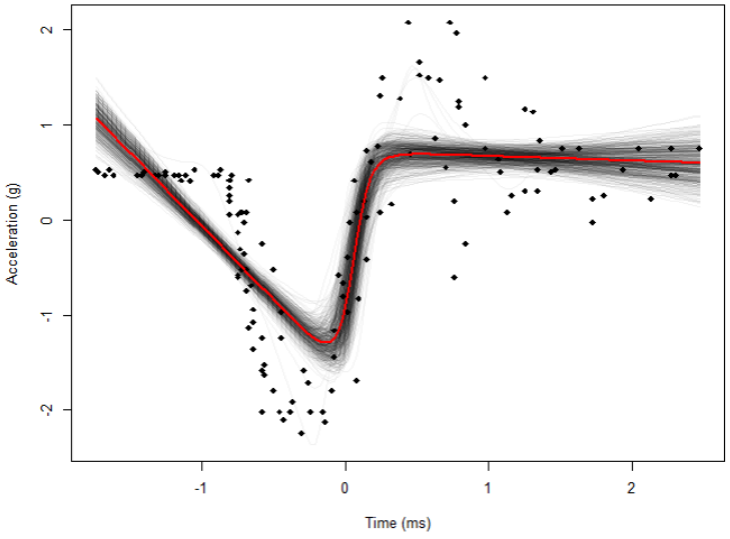


FIGURE 6.8: Naive RJ MCMC predictions for the motorcycle accident data. Every 10-th prediction shown. Average predictions shown in red.

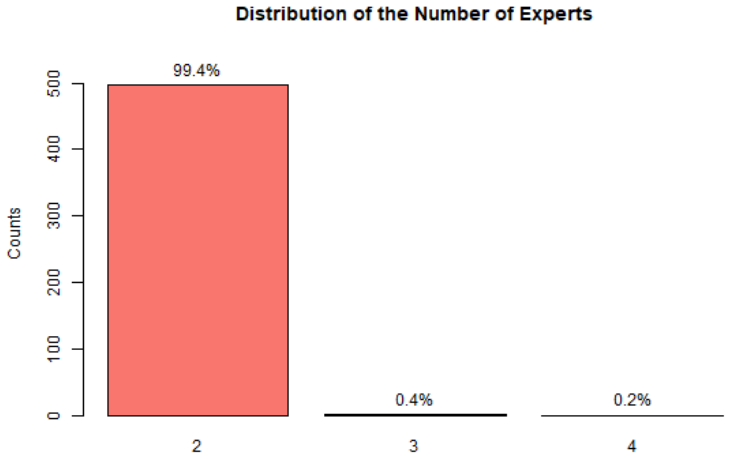


FIGURE 6.9: Distribution of the number of experts in the naive RJ MCMC chain for the motorcycle accident data.

6.8.2 Informed Reversible Jump MCMC Results

Next, all new methods proposed in this chapter are applied to the same problem with the exception of the model size prior. At this stage, it is of interest to find out what number of experts the chain spends most time in without the influence of model size prior. The obtained results can then help inform whether such prior is required for this application.

The forward jump parameters are drawn from two Gaussian distributions, i.e., $\boldsymbol{\gamma}_1 \sim \mathsf{N}(0, 100)$, $\epsilon \sim \mathsf{N}(0, 0.5)$ to ensure the comparability of the two methods (see Section 6.5.2 for forward jump proposal details) . The resulting predictions are shown in Figure 6.10 while the corresponding acceptance rates are given in Table 6.3.

TABLE 6.3: Acceptance rates of the informed RJ algorithm jumps for the motorcycle accident data.

	Splits	**Merges**	**All Jumps**
Number Proposed	250	250	500
Number Accepted	30	31	61
Acceptance	12%	12.4%	12.2%

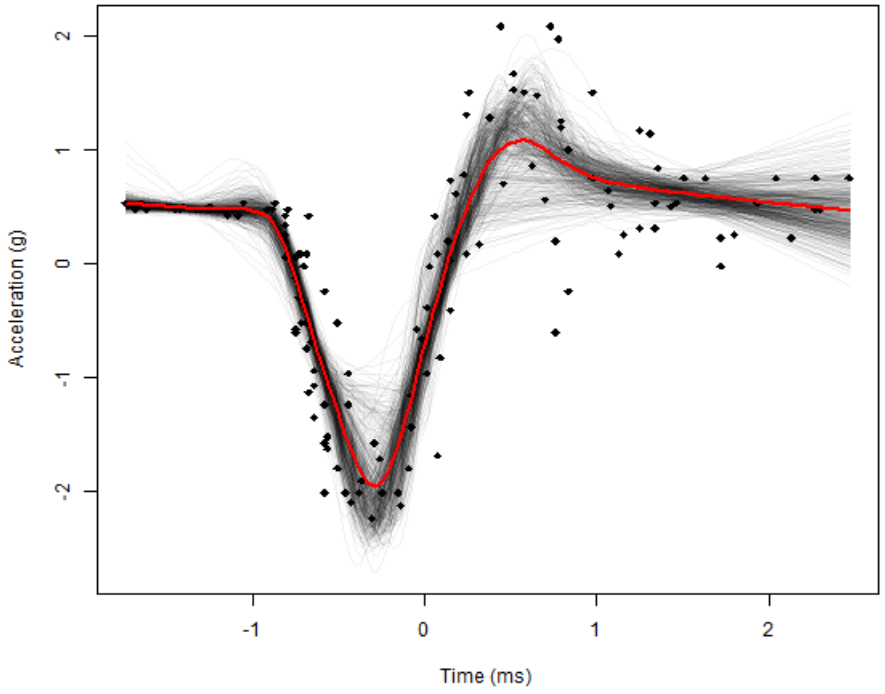


FIGURE 6.10: Informed RJ MCMC predictions for the motorcycle accident data. Every 10-th prediction shown. Average predictions shown in red.

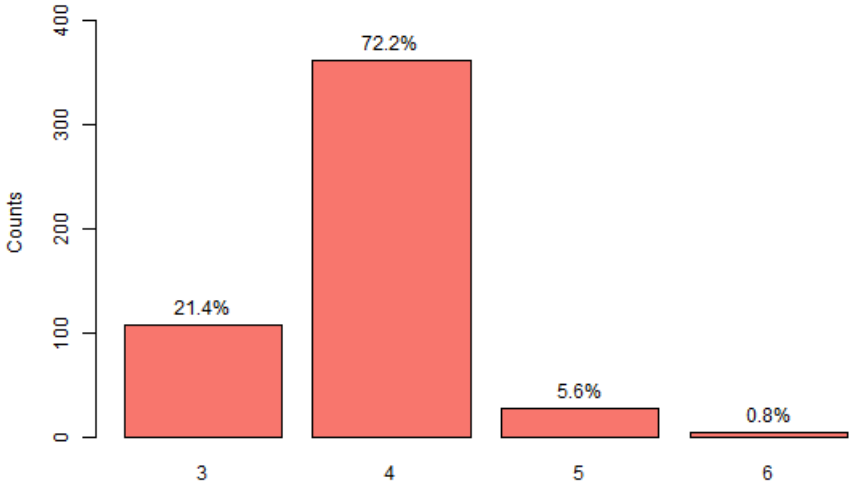


FIGURE 6.11: Distribution of the number of experts in the informed RJ MCMC chain for the motorcycle accident data.

It is evident that the informed RJ MCMC predictions are tracking the data really well with the red line showing the average predictions. All modes of the posterior distribution appear to have been visited and investigated by the MCMC chain. It can further be seen that 12% of the proposed splits and 12.4% of the proposed merges have been accepted yielding an overall acceptance rate of 12.2% for the reversible jump proposals. The latter jump acceptance rates are a vast improvement on those seen for the naive RJ MCMC (0%, 1.17% and 0.6% respectively). The distribution of the number of experts shown in Figure 6.11 reveals that 72.2% of the time, the HME model consisted of 4 experts. It is also evident that the RJ MCMC explored models with three to six experts in them.

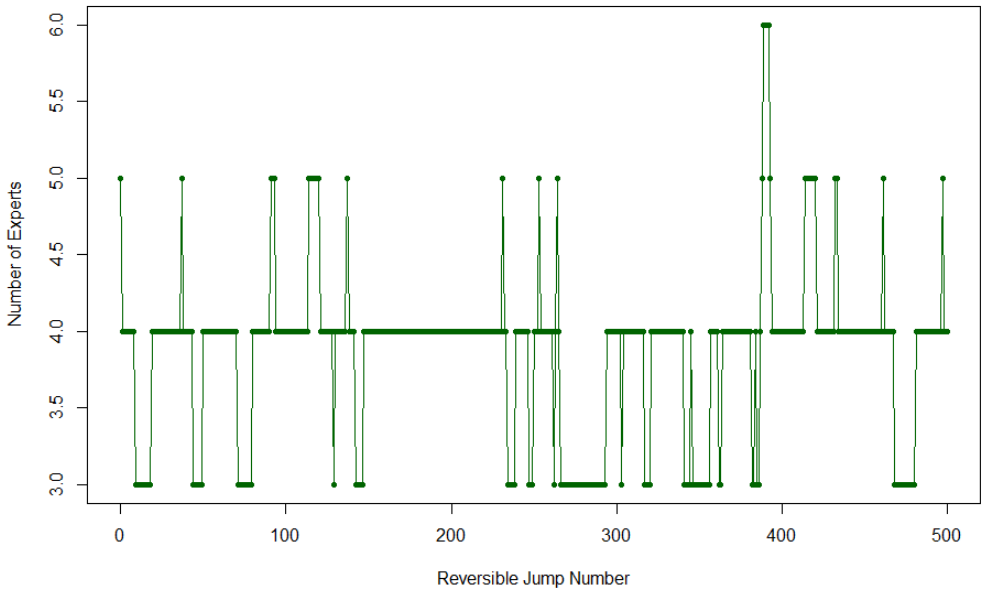


FIGURE 6.12: The number of experts in the HME tree after each informed RJ proposal step for the motorcycle accident data.

Figure 6.12 shows the number of experts present in the HME tree after each informed reversible jump. In agreement with observations made thus far, there is evidence of a changing architecture and movement between three and five experts for the majority of the time. The architecture, however, does not settle on a particular number of experts. Increasing the number of MCMC iterations has also been investigated and has resulted in the same outcome. This means that there is no meaning to commenting on posterior parameter estimates and thus the average model fit is discussed as a whole.

To further illustrate a HME model fit, consider an example of a model fit at a randomly chosen iteration shown in Figure 6.13.

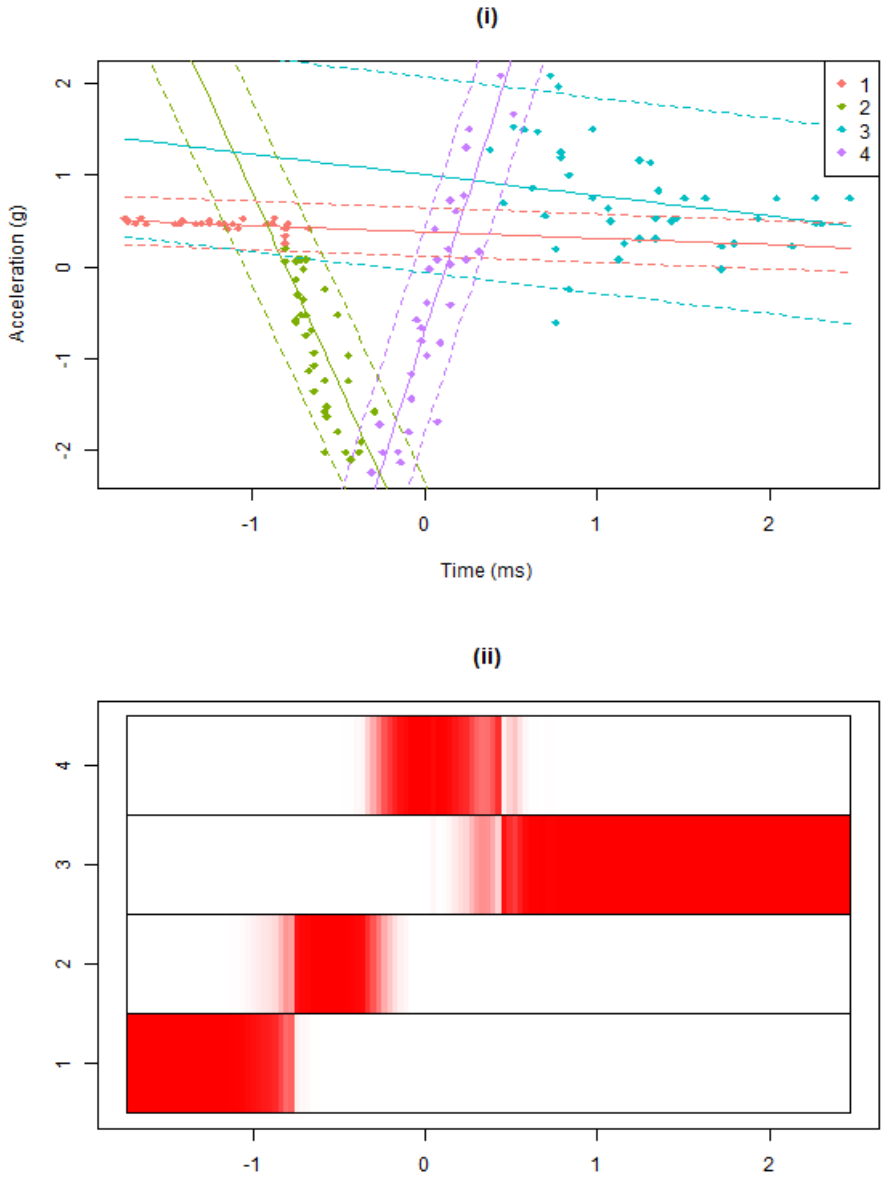


FIGURE 6.13: (i) HME model fit at a randomly selected iteration using informed RJ MCMC for the motorcycle accident data; (ii) representation of the expert activity in the explanatory variable space, where red bars correspond to responsibilities.

From plot (i), it can be seen that at this particular iteration, there were 4 experts present in the model each represented by a different color. The solid lines are equivalent to the fitted linear regression lines obtained using the posterior intercept and slope expert parameters. As anticipated, the fitted lines vary in both location and steepness thus accurately representing the changing relationship between time and acceleration. Similarly, the dashed lines represent the uncertainty bounds defined by the posterior expert variance. It is immediately obvious that these are not constant across experts with the smallest variance present for expert 1 (pink) and the largest for expert 3 (blue).

The observations are consistent with those made at the exploratory stage. The level of abruptness in the separation between experts can be visualised using individual responsibilities, which are equivalent to the product of the path probabilities and expert densities for each point, shown in plot (ii). Overall, it appears that experts are covering a distinct range of the explanatory variable, however, there is a slight overlap between experts 3 (blue) and 4 (purple) at the point of change. Figure 6.13 showcases one of many plots available in the bespoke `R-Shiny` application, which is discussed in detail in Section 6.8.5. Next, the prediction intervals are investigated in order to assess the effects of heteroscedasticity on the model fit (Figure 6.14).

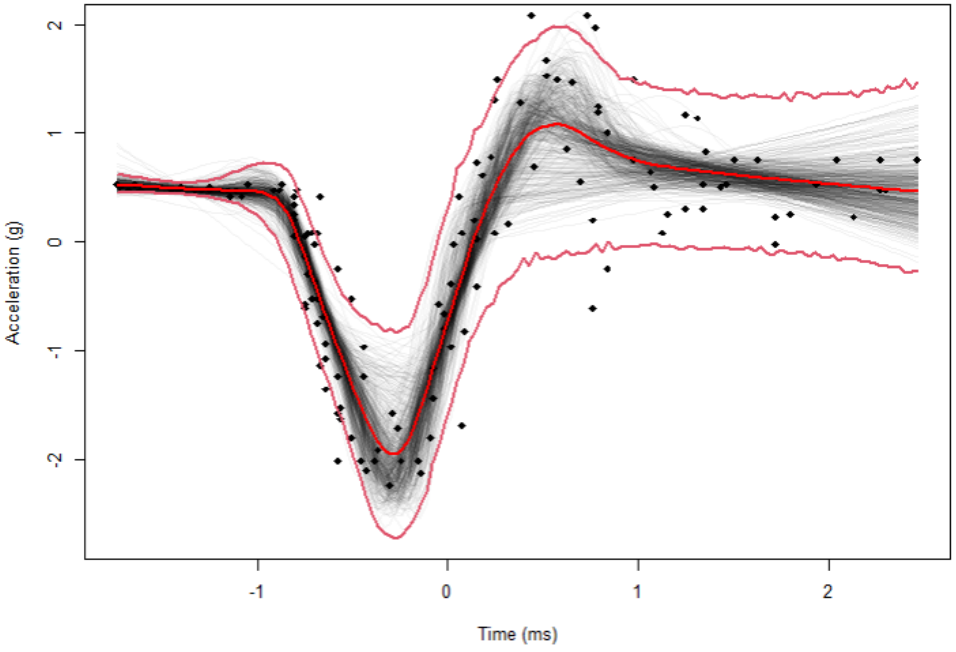


FIGURE 6.14: HME prediction intervals for the motorcycle accident data. The thin red lines show the 2.5-th and 97.5-th percentiles of predictions made during the informed RJ MCMC iterations. The thick red line shows the average predictions. For more details on obtaining predictions, please refer to Appendix E.

It can be seen that the prediction intervals account for the change in variance and are broader for areas with more uncertainty and narrower for the areas with tighter clustered observations. This is one of the features that makes HME models so appealing. In Chapter 8, the competitors of HME models are also assessed in the context of accounting for heteroscadasticity.

Next, the three unhelpful starting points presented in Section 5.5 are evaluated and compared to the previously obtained outcome.

6.8.3 Mixing and Convergence

The deliberately unhelpful starting point evaluation for the motorcycle accident data undertaken in Section 5.5 is revisited in this section. It has been shown that running an MCMC chain with pre-set architecture from the three starting points yields inconsistent results across the chains as well as highlights poor mixing, where only a subset of posterior distribution modes is explored by each of the chains (Figure 6.15). It is now of interest to investigate whether the addition of the informed reversible jump helps chains investigate modes of the posterior distribution despite their unfortunate starting points.

The same starting parameter values are now used to initialise the three informed RJ MCMC chains as per methodology proposed in this chapter. From this point onwards, the term naive RJ MCMC refers to the uninformed RJ MCMC while the terms RJ MCMC and informed RJ MCMC are used interchangeably. To ensure comparability of the two methods, the pre-set HME architecture used in this evaluation is consistent with the one seen for MCMC with no reversible jump.

The results of the evaluation are shown in Figure 6.16. It is evident that the addition of the reversible jump has notably improved the overall HME model fit for all three chains. Consistent predictions are produced across all three cases hence showing that the RJ MCMC has explored all modes of the posterior distribution and removed the effect of the unfortunate starting point.

It can be deduced that the addition of the reversible jump can improve mixing for HME models. The predictions obtained for all three starting points appear to be consistent and tightly clustered together suggesting that the chains have converged to the same stationary distribution. To formally check that convergence has been reached across and within the chains, the Gelman-Rubin diagnostic is examined next.

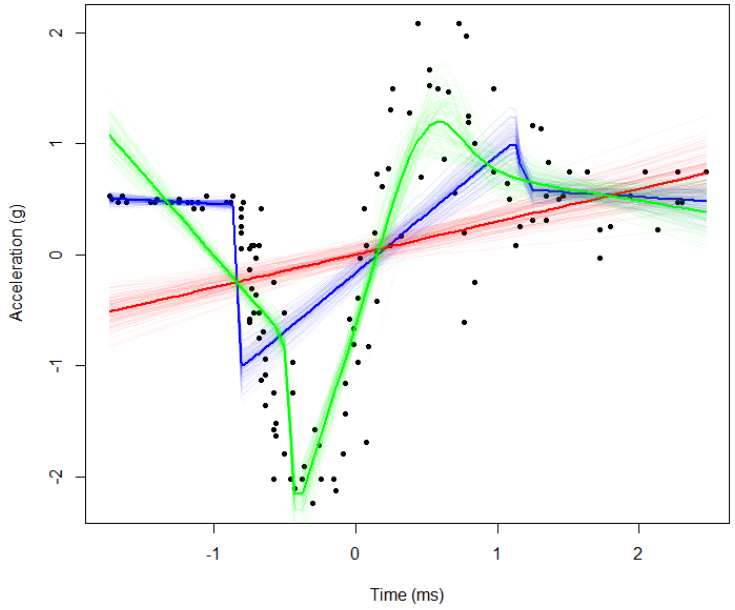


FIGURE 6.15: Equivalent to Figure 5.5. MCMC predictions with the three starting points reflected by color for motorcycle accident data. The thick lines represent the average predictions while the thin lines represent every 10-th prediction.

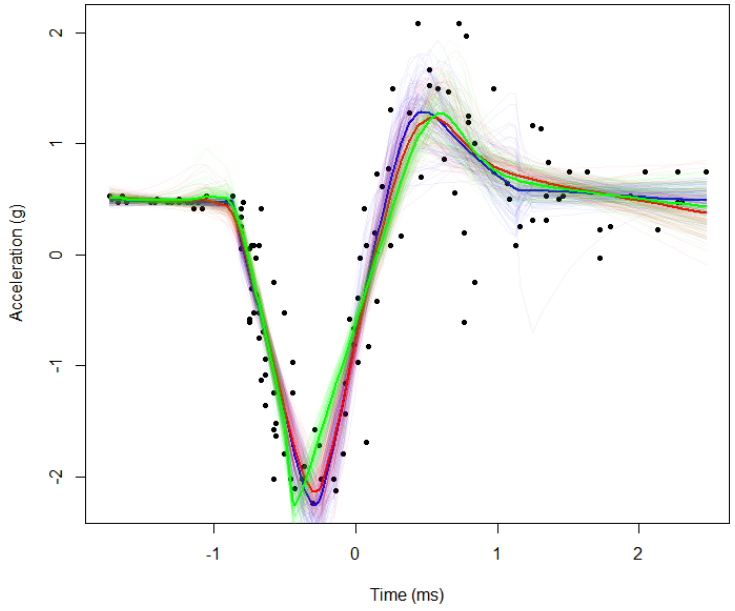


FIGURE 6.16: RJ MCMC predictions obtained with the three starting points reflected by color for motorcycle accident data. The thick lines represent the average predictions while the thin lines represent every 10-th prediction.

The Gelman-Rubin diagnostic yields a value of 1.13, which strongly suggests that the convergence has been achieved as well as is a clear improvement over the value of 5.02 as seen for the MCMC with no reversible jump (see Table 6.4).

TABLE 6.4: The potential scale reduction factor (PSRF) for chains with three randomly drawn parameter starting values obtained using MCMC and RJ MCMC for motorcycle accident data.

	MCMC	**RJ MCMC**
PSFR	5.02	1.13

Having shown that the addition of the informed reversible jump step can improve mixing and convergence, the simplest starting point of the HME model with one expert is considered next.

6.8.4 Automatic HME Tree Growth for Motorcycle Accident Data

In the previous sections, an initial state for HME model consisted of a pre-set architecture. Setting a starting number of experts also requires providing the initial allocation variables and model parameter values, which becomes challenging when working with more than two dimensions. In this section, we start with only one normal expert and let the RJ MCMC guide us to the preferred number of experts. Similarly as seen before, Figure 6.17 showcases the resultant predictions while the acceptance rates are given in Table 6.5.

TABLE 6.5: Acceptance rates of the RJ algorithm jumps with initial start of 1 expert for motorcycle accident data.

	Splits	**Merges**	**All Jumps**
Number Proposed	257	243	500
Number Accepted	21	18	61
Acceptance	8.17%	7.41%	7.8%

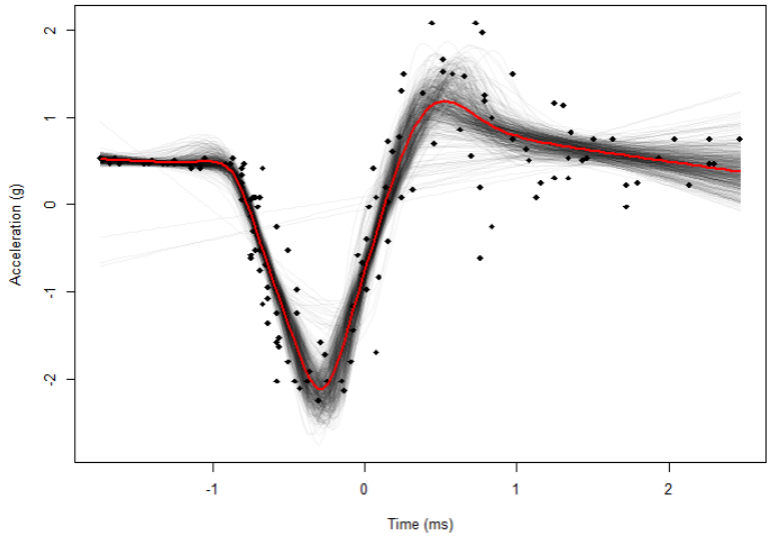


FIGURE 6.17: RJ MCMC predictions with initial start of 1 expert for the motorcycle accident data. Every 10-th prediction shown. Average predictions shown in red.

The predictions appear consistent with the ones seen in the previous section. On the other hand, the acceptance rates appear slightly lower than those seen for the initial start of HME with 5 experts. A closer look at the distribution of experts, shown in Figure 6.18, reveals that the majority (86.4%) of HME models within the RJ MCMC chain consisted of 4 experts, which is in agreement with the results seen before. Understandably, when starting with 1 expert, more time is spent growing the tree, while starting with 5 experts results in not accepting the merges that would yield to a model with less than 3 experts (see Figure 6.11).

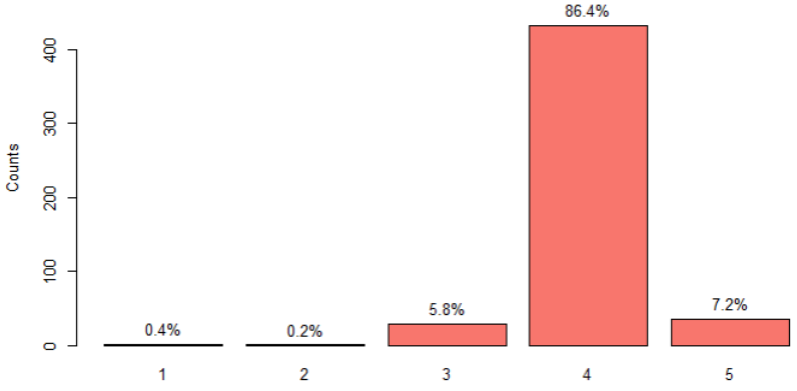


FIGURE 6.18: Distribution of the number of experts in the RJ MCMC chain with an initial start of 1 expert for the motorcycle accident data.

Figure 6.19 further reveals that the number of experts increases to four within the first 10 jumps. Following that, the number of experts ranges from three to five, which is consistent with results seen for an initial start with five experts (Figure 6.12).

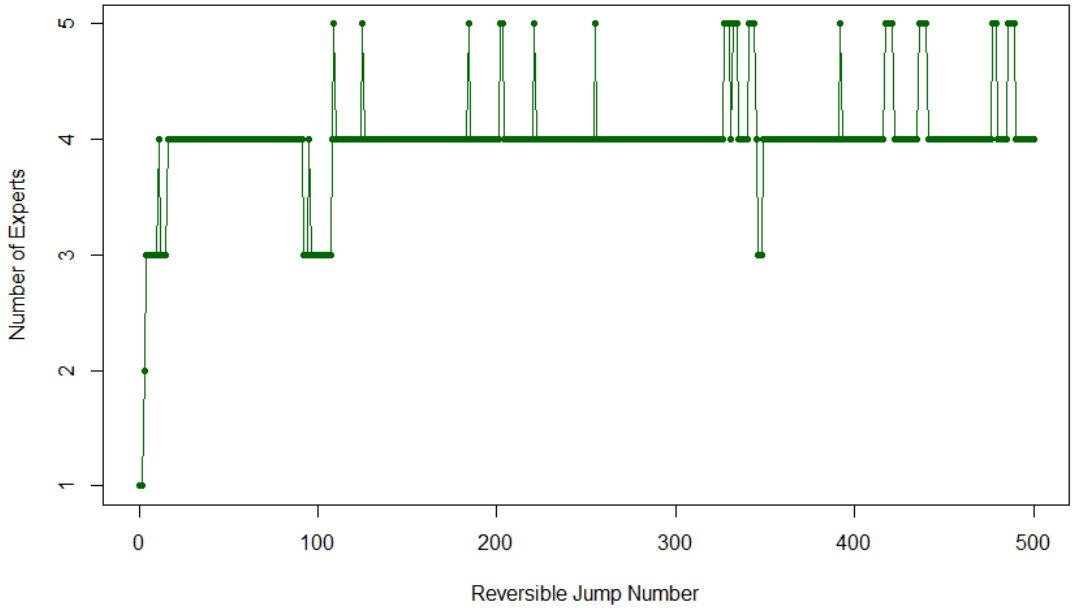


FIGURE 6.19: The number of experts in the HME tree after each informed RJ step proposal with an initial start of 1 expert for the motorcycle accident data.

Overall, the results presented in this section suggest that starting with the simplest model and allowing the RJ MCMC to select the required number of experts is a viable tactic that leads to models consistent with those achieved with an informed pre-set architecture. It thus seems preferable to start with one expert in the tree and hence reduce issues arising from selecting the number of experts, their arrangement in the tree, and the associated initial model parameters.

Thus far, it has been shown that, for motorcycle accident data, the proposed methodology for informed reversible jumps outperforms the naive approach as well as improves mixing and convergence. It has also been demonstrated that the RJ MCMC can grow HME model trees from the simplest starting point of one expert thus improving on the strategy of initialising the RJ MCMC chains with an informed tree architecture. It then follows that the recommended strategy is to start with a one expert architecture and use the informed RJ MCMC methodology for fitting HME models. An interactive interrogation of such strategy for the motorcycle data is further illustrated in the next section, where the previously referred to bespoke `R-Shiny` application is presented.

6.8.5 Interactive Illustration of HME Model Fit for Motorcycle Accident Data

An `R-Shiny` application, also referred to as the app, has been created to allow for a closer investigation of the HME model fit. The tool could be used for interpreting any HME model fitted using RJ (GS) MCMC as well as assessing convergence. This section provides an overview of each tab available within the app (Figure 6.20), highlights the key benefits, and discusses the arising features of HME models.

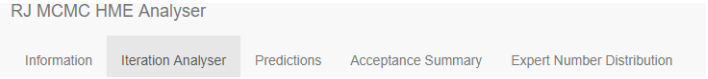


FIGURE 6.20: Tabs available within the `R-Shiny` application.

For illustration, the motorcycle accident data application, where an HME model is fitted using the RJ MCMC with an initial start of one expert, is used. Additional functionality that allows for uploading new data to the app could be implemented in order to use the app outside of the discussed example.

Iteration Analyser

The `Iteration Analyser` tab of the app provides a number of control options shown in Figure 6.21. Each of those options is next discussed in turn.

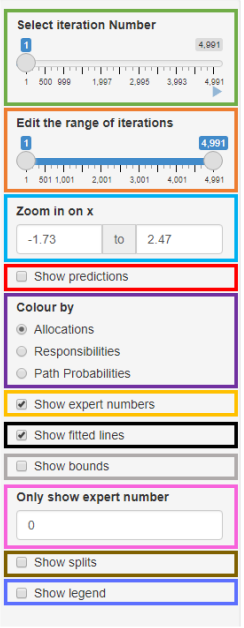
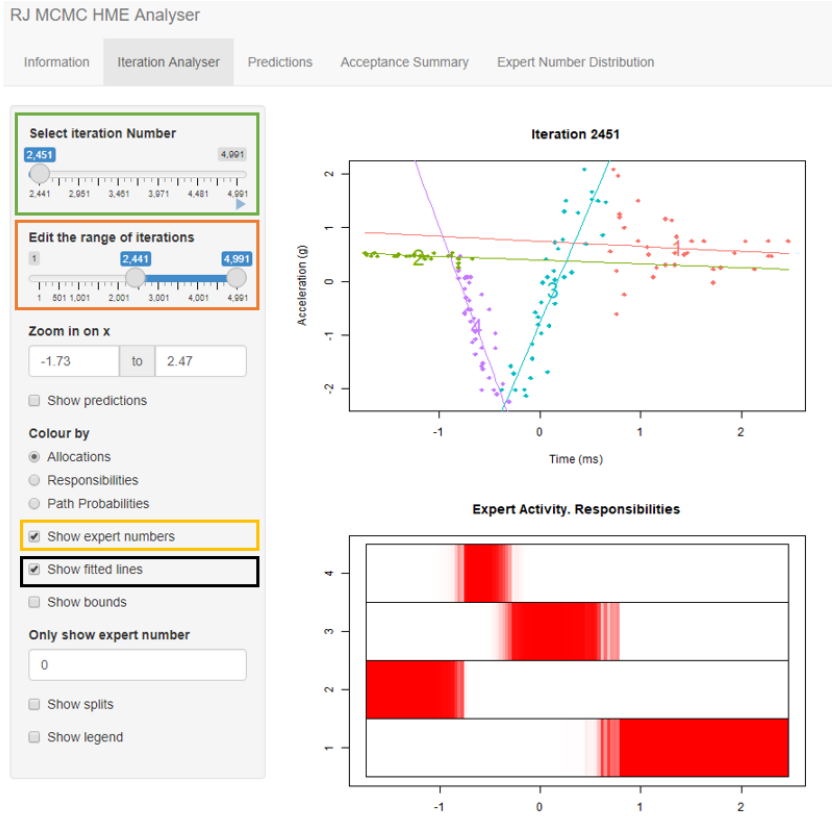


FIGURE 6.21: Control options in the `Iteration Analyser` tab of the `R-Shiny` application.

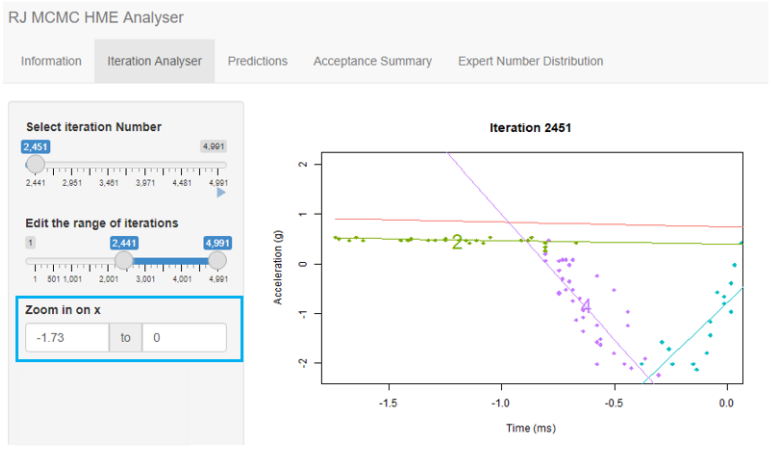
1. **Select iteration number** (green box);
   This control allows for picking an iteration to be displayed. In this case, a user can choose from a set of iterations recorded after each reversible jump. For a reversible jump, which is proposed every 10-th iteration in a chain of length $5,000$, the app allows viewing iterations $1, 11, 21, ..., 4991$. The blue triangle at the bottom right of the green box initiates an animation that goes through the chosen range of iterations.

2. **Edit the range of iterations** (orange box).
   This control allows for zooming in on a range of iterations of interest. The output displayed for a randomly selected iteration is shown in the figure below.

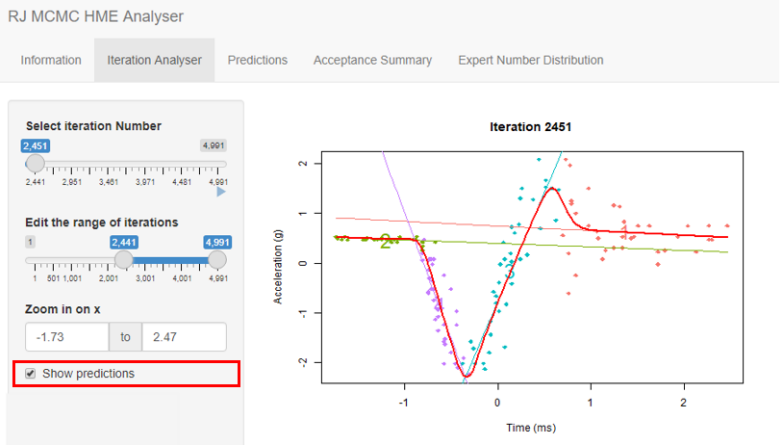


In the above, the top plot depicts the allocations to the experts present in the tree at a chosen iteration as well as shows the fitted regression lines corresponding to the experts. The fitted lines can be removed by unticking **Show fitted lines** (black box). It can also be seen that experts are marked by numbers in the plot, which can be removed by unticking **Show expert numbers** (yellow box). The bottom plot shows expert activity across explanatory variable space measured by a default metric of responsibilities.

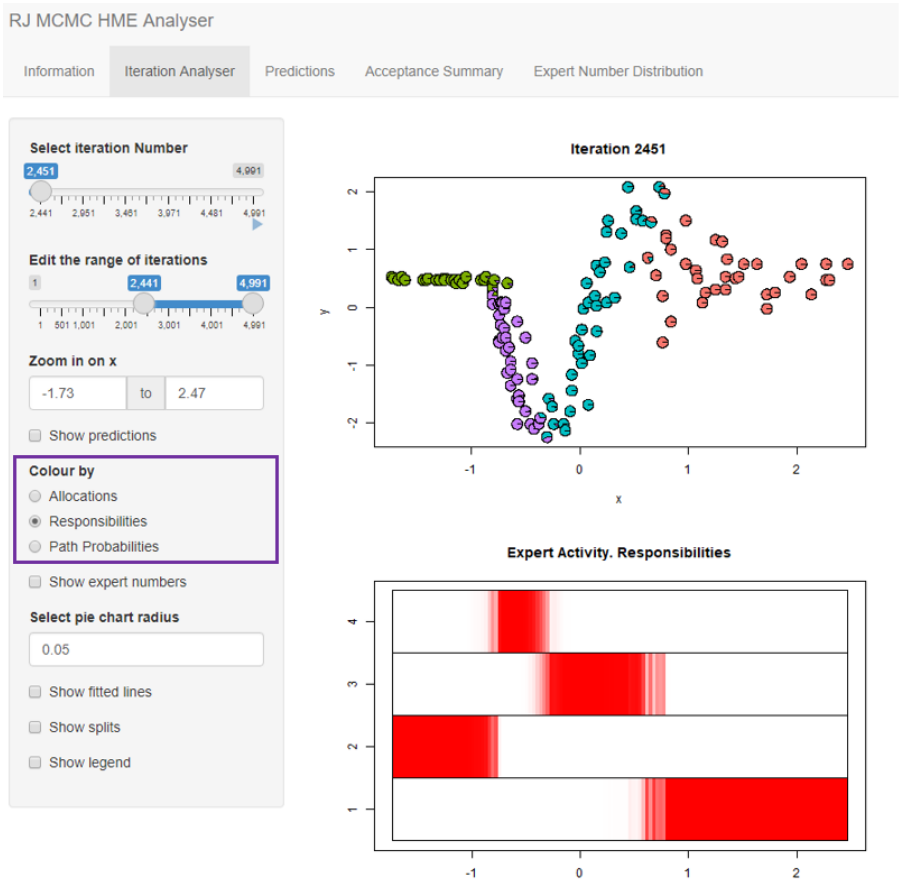
3. **Zoom in on x.** (light blue box).
   The default appearance of the output can be altered by zooming in on the explanatory variable. For example, the negative values of the standardised time can be zoomed in on as shown in the figure below.

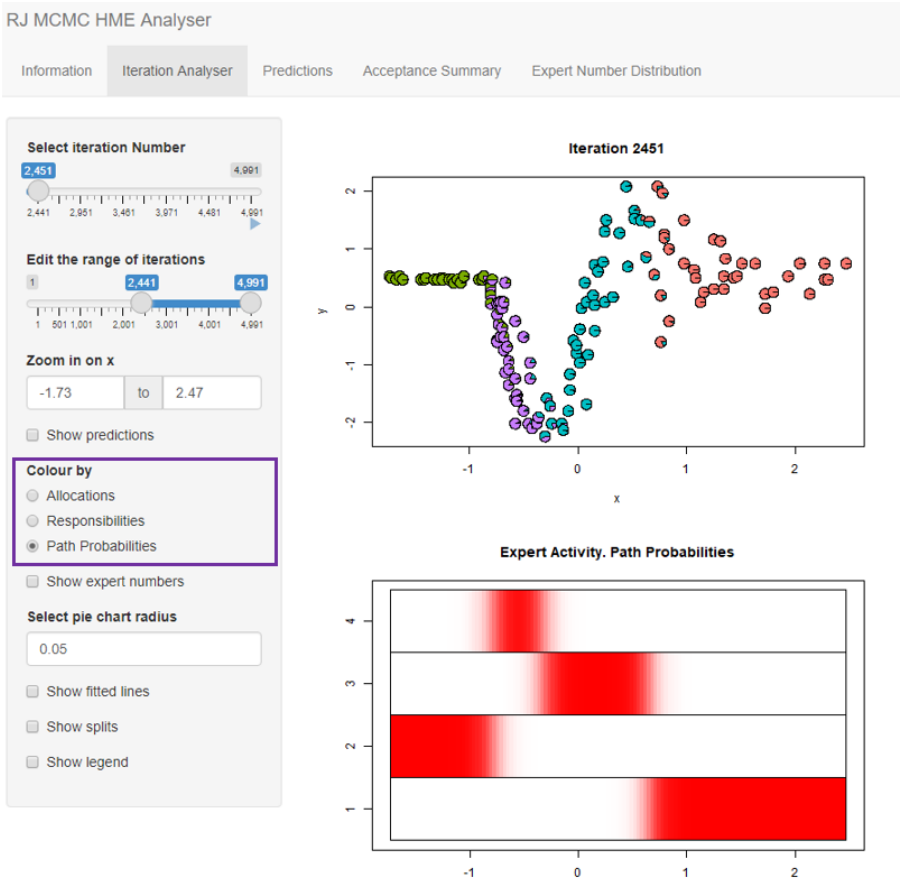


4. **Show predictions** (red box).
   This control displays the predictions for the selected iteration as demonstrated below.



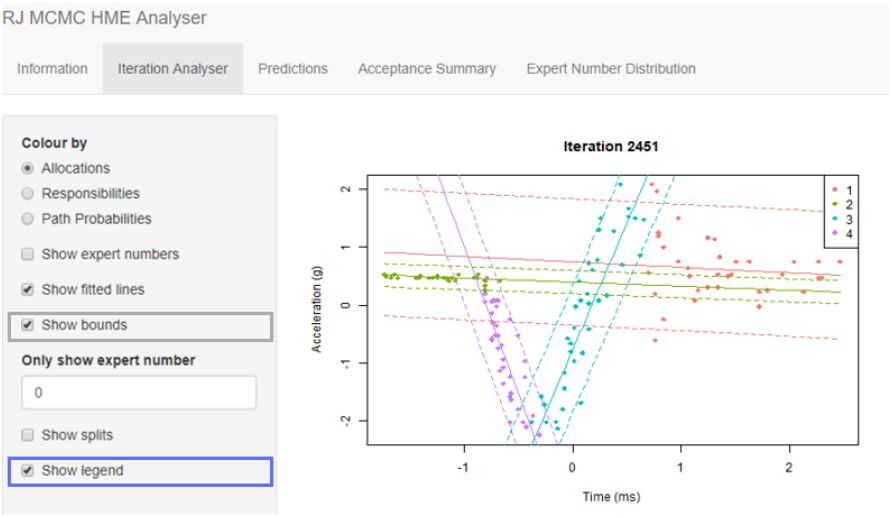
5. **Colour by** (purple box).
   This control switches the default coloring of points by allocations to a view, where each point is represented by a piechart. The slices of each piechart then correspond to either path probabilities or responsibilities (selected by user) across all experts. The bottom plot also adjusts to showcase expert activity with respect to a chosen metric. This view reveals that path probabilities, which depend solely on the gating parameters and the explanatory variable, suggest a smoother transition between experts than that seen after expert densities are taken into account.


RJ MCMC HME Analyser
Information
Iteration Analyser
Predictions
Acceptance Summary
Expert Number Distribution
Select iteration Number
2,451
4,991
Edit the range of iterations
Zoom in on x
-1.73
to
2.47
Show predictions
Colour by
Allocations
Responsibilities
Path Probabilities
Show expert numbers
Select pie chart radius
0.05
Show fitted lines
Show splits
Show legend
Iteration 2451
Expert Activity. Responsibilities


RJ MCMC HME Analyser
Information
Iteration Analyser
Predictions
Acceptance Summary
Expert Number Distribution
Select iteration Number
2,451
4,991
Edit the range of iterations
Zoom in on x
-1.73
to
2.47
Show predictions
Colour by
Allocations
Responsibilities
Path Probabilities
Show expert numbers
Select pie chart radius
0.05
Show fitted lines
Show splits
Show legend
Iteration 2451
Expert Activity. Path Probabilities

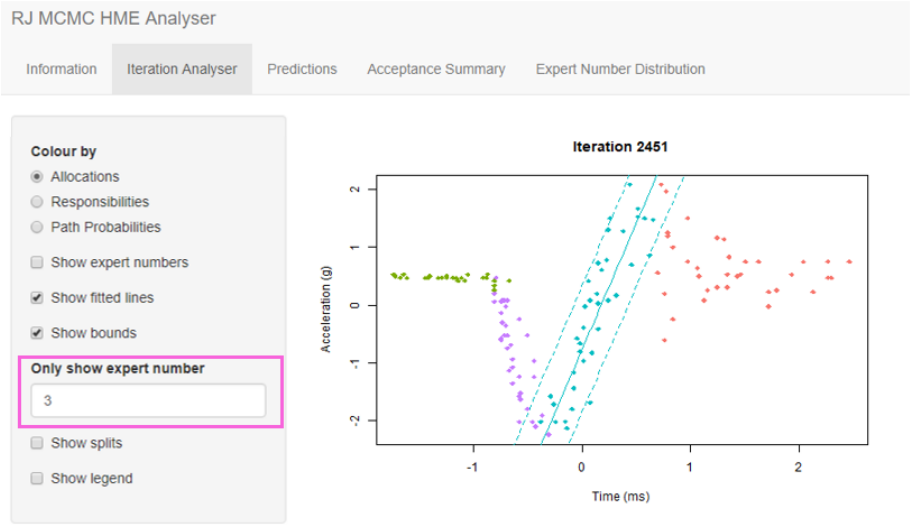
6. **Show bounds** (grey box)

   This control displays the uncertainty bounds defined by the expert variance parameter. This alternative to viewing expert numbers printed in the plot can be selected by ticking **Show legend** (dark blue box).



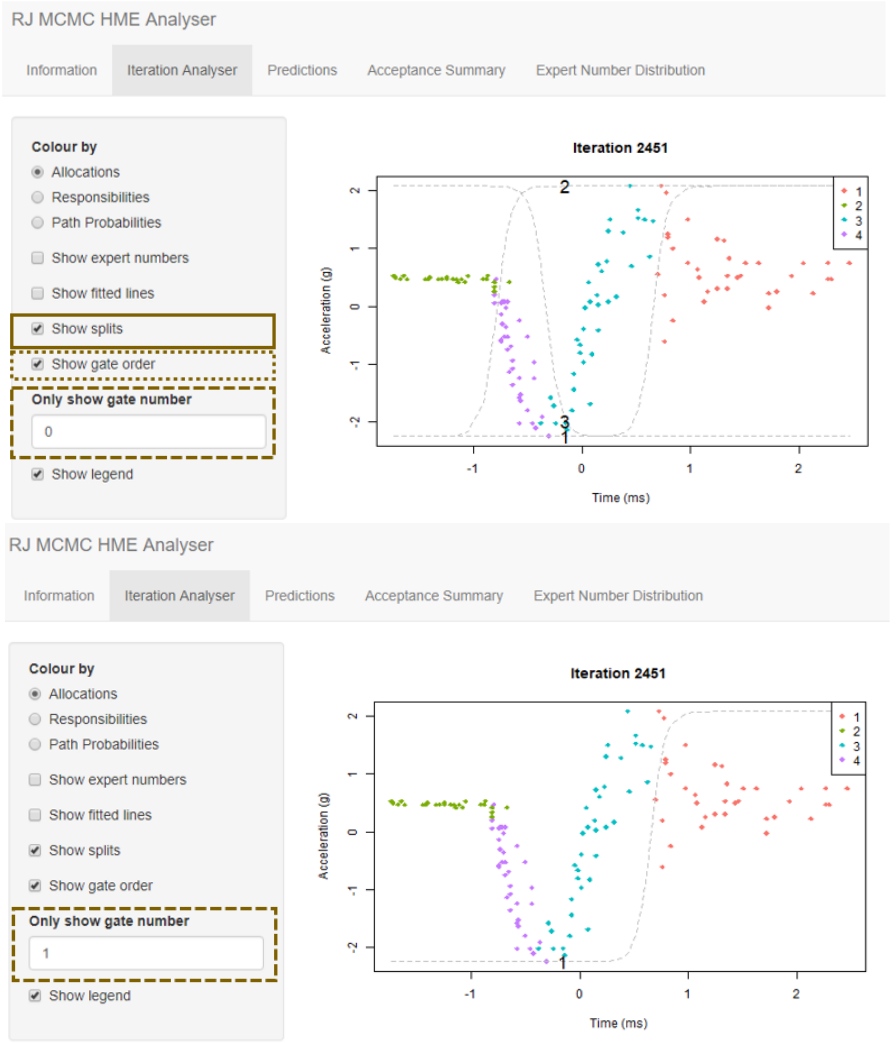
7. **Only show expert number** (pink box)

   This control allows for viewing the regression fit for one expert at a time. For example, picking expert 3 yields the view shown below.



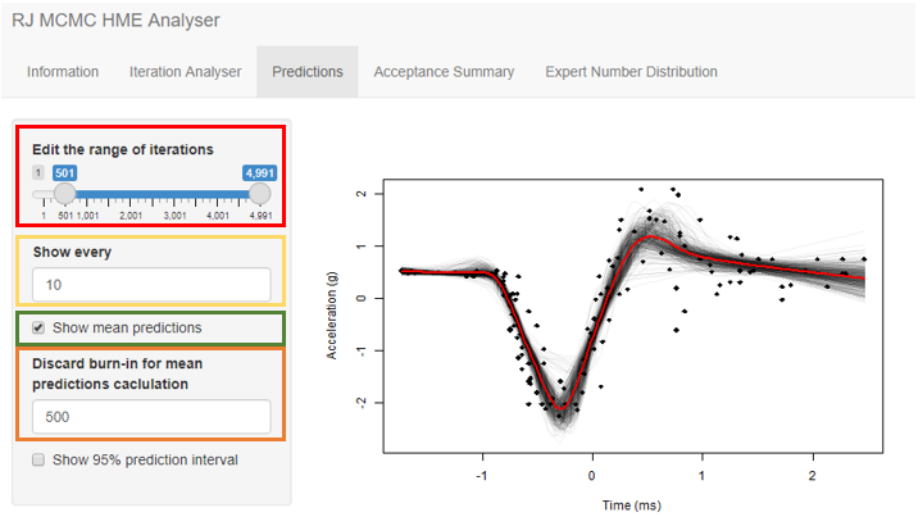
8. **Show splits** (brown box).

   This control allows for viewing the location and abruptness of the separation between experts. Ticking **Show splits** prompts the appearance of **Show gate order** (dotted brown box), which displays the order in which the splitting of the space occurs, and **Only show gate number** (dashed brown box), which allows for viewing one split at a time. This feature introduces the option to reconstruct the tree architecture at each iteration.

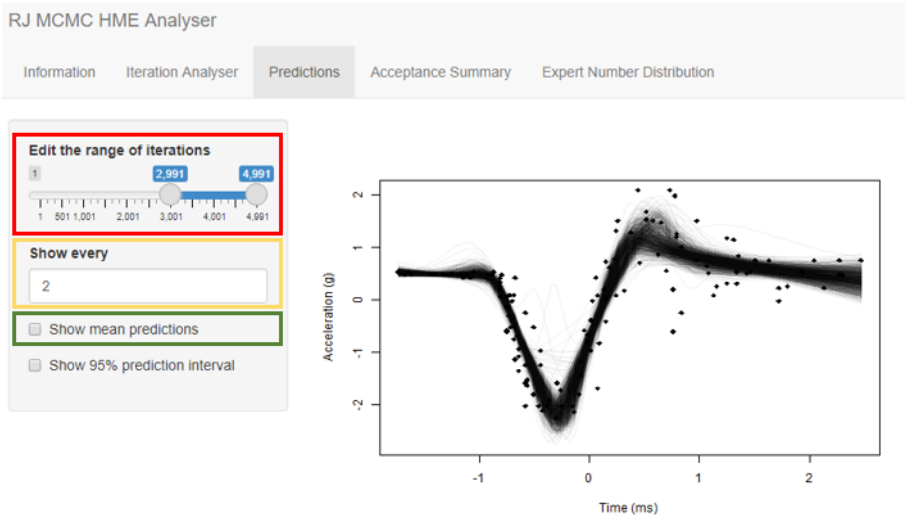


Predictions

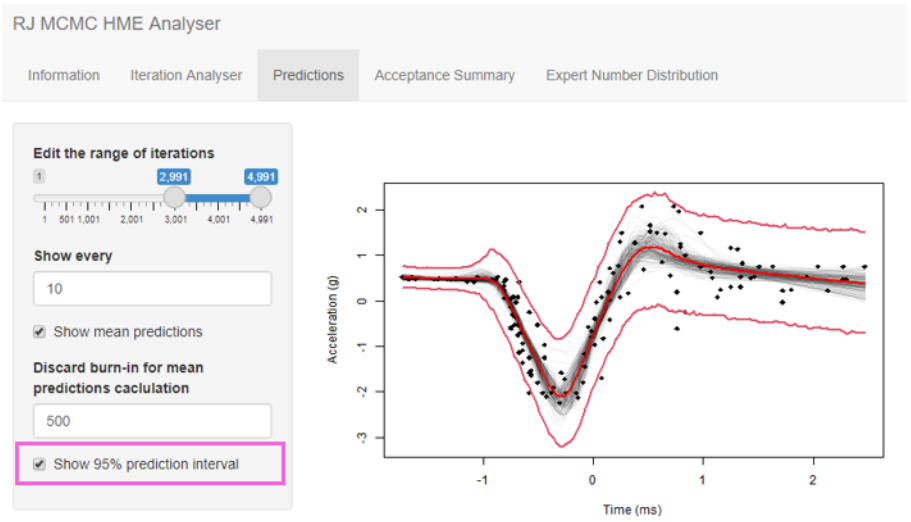
The `Predictions` tab allows for viewing the predictions made at each iteration on one plot. The range of predictions can be adjusted using the **Edit the range of iterations** (red box) while the value k entered in **Show every** (yellow box) results in every k-th iteration being plotted. The mean of predictions can be added/removed from the plot by ticking/unticking **Show mean predictions** (green box) with an option to control how many runs are discounted for burn-in when calculating the mean in **Discard burn-in for mean predictions calculation** (orange box). For instance, a plot for iterations $501, ..., 4991$ plotting every 10-th prediction and the mean for predictions calculated discounting first 500 runs can be obtained as shown below.



Consider the situation where one is interested in predictions made later on in the chain. Changing the range of iterations to only view the last 2000 iterations and plotting every second prediction without the mean predictions can be set up as shown below.

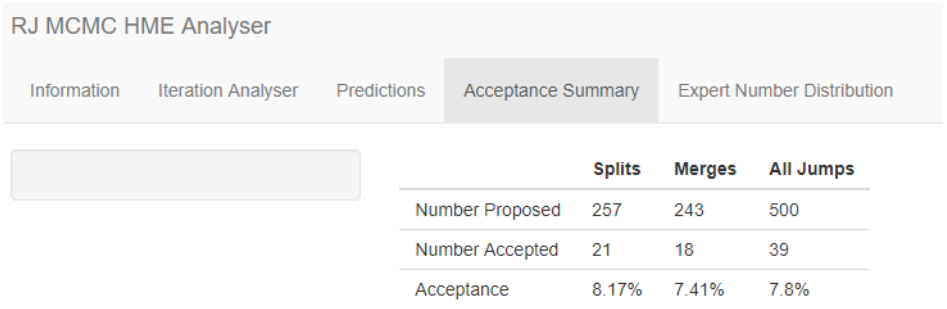


This tool is particularly useful for assessing the convergence of predictions. It can be used to investigate whether and when the predictions start to appear consistent (see Section 4.3.4.2 for more details). Finally, the control **Show 95% prediction interval** (pink box) can be used to visualise prediction intervals.



Acceptance Summary

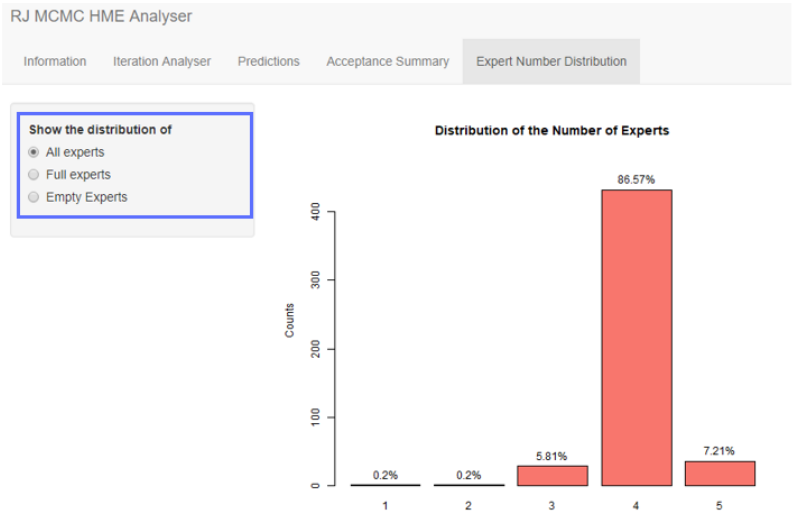
The `Acceptance Summary` tab provides a simple summary of the reversible jump acceptance rates as demonstrated in the figure below.

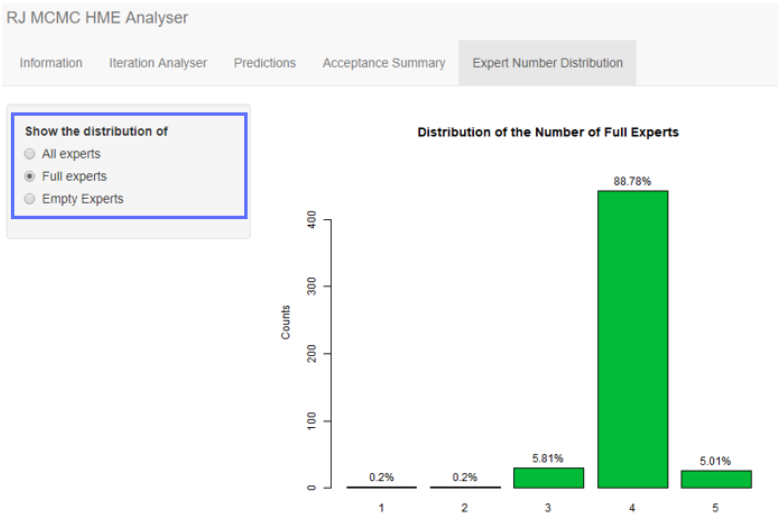


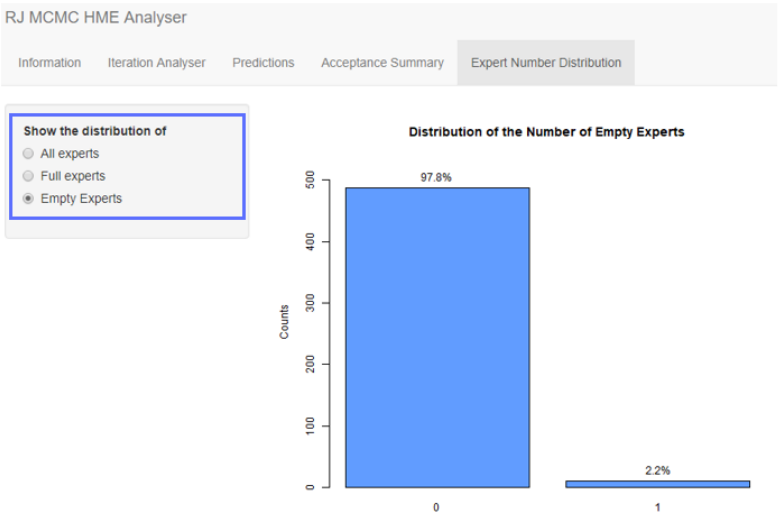
	Splits	Merges	All Jumps
Number Proposed	257	243	500
Number Accepted	21	18	39
Acceptance	8.17%	7.41%	7.8%

Expert Number Distribution

The `Expert Number Distribution` tab provides a view of the expert number distribution in the MCMC chain. The **Show the distribution of** (blue box) control provides an option to flick between the distribution of all experts, full experts, and empty experts as shown in the respective figures below. This extension provides insight into how often the RJ MCMC chain retains experts with no observations in them. An infrequent appearance of empty experts suggests that the merges are working as expected within the RJ methodology.


RJ MCMC HME Analyser
Information
Iteration Analyser
Predictions
Acceptance Summary
Expert Number Distribution
Show the distribution of
All experts
Full experts
Empty Experts
Distribution of the Number of Experts
Counts
0
100
200
300
400
0.2%
0.2%
5.81%
86.57%
7.21%
1
2
3
4
5


RJ MCMC HME Analyser
Information
Iteration Analyser
Predictions
Acceptance Summary
Expert Number Distribution
Show the distribution of
All experts
Full experts
Empty Experts
Distribution of the Number of Full Experts
Counts
0
100
200
300
400
0.2%
0.2%
5.81%
88.78%
5.01%
1
2
3
4
5


RJ MCMC HME Analyser
Information
Iteration Analyser
Predictions
Acceptance Summary
Expert Number Distribution
Show the distribution of
All experts
Full experts
Empty Experts
Distribution of the Number of Empty Experts
Counts
0
100
200
300
400
500
97.8%
2.2%
0
1

Access

The `R-Shiny` app presented in this section can be accessed here.

So far, proposing a single jump every 10 MCMC iterations has been considered. As shown in this section, such choice of number and frequency of reversible jumps provided a good starting point for the visualisation and evaluation of the proposed methodology. It is, however, important to understand how the frequency and quantity of proposed jumps affect the overall model fit.

6.8.6 Frequency and Number of Jumps in Automatic HME Architecture Selection for Motorcycle Accident Data

The application of the RJ MCMC on the motorcycle data in the previous section is based on proposing 1 jump every 10 MCMC iterations. It is important to consider the impact and the meaning of the latter tuning parameters. Let L denote the number of consecutive jumps proposed (without MCMC inbetween) and let K denote the frequency of these jumps, i.e. propose L jumps every K-th MCMC iteration.

The larger the value of K, the longer the MCMC chain has to improve on the new state proposed by the jump. If the value of L is larger than 1, the option of escaping an unfortunate jump becomes available, however, one is then facing the risk of discarding a beneficial jump before MCMC has had a chance to improve on it. We start by investigating the effect of the number of jumps proposed with no MCMC in between (varying L) in the motorcycle accident data with results given in Table 6.6.

TABLE 6.6: The number of jumps proposed within the RJ MCMC for the motorcycle accident data vs acceptance rates, average number of experts in the model, and run time.

K=10		**Total Jumps**	**Overall Acceptance**	**Split Acceptance**	**Merge Acceptance**	**Average Number of Experts**	**Run Time (s)**
	1	500	7.8%	8.17%	7.41%	3.92	239.67
	2	1000	8.10%	8.33%	7.86%	3.90	233.28
L	*3*	1500	13.53%	13.31%	13.77%	3.72	236.34
	5	2500	11.40%	11.71%	11.1%	3.75	238.81
	10	5000	19.34%	19.41%	19.27%	3.60	236.43

In general, it can be seen that all scenarios yield a similar number of experts on average. It is likely that in the motorcycle data case an HME model with at most 4 experts is

sufficient. Overall, the acceptance rates do not follow a strict pattern as the value of L increases. The highest acceptance rate is achieved with $L = 10$, which might indicate that the accepted proposals are escaped, or *undone*, before getting a chance to be improved on by MCMC. This investigation suggests that the number of jumps proposed every k-th iteration does not appear to have an effect on the overall preferred HME model size. On the other hand, it highlights the importance of choosing a value of L that allows for an escape from unfortunate jumps while giving the MCMC an opportunity to improve on the proposed state. Another important point to note is that there is no apparent difference in the run time of the algorithm as L increases, which implies that the cost of additional jumps is relatively low. Next, the effect of the frequency of the reversible jumps is investigated. It is of interest to understand whether allowing for longer MCMC chains in between the jumps results in better intermediate states thus improving the acceptance rates.

TABLE 6.7: The frequency of jumps proposed within the RJ MCMC for the motorcycle accident data vs acceptance rates, average number of experts in the model, and run time.

L=1		**Total Jumps**	**Overall Acceptance**	**Split Acceptance**	**Merge Acceptance**	**Average Number of Experts**	**Run Time (s)**
	5	1000	12.50%	12.57%	12.42%	3.89	262.16
	10	500	7.8%	8.17%	7.41%	3.92	239.67
K	*25*	200	14.5%	15.09%	13.83%	3.96	228.43
	50	100	7.00%	10.00%	4.00%	3.81	211.54
	100	50	18.00%	19.35%	15.79%	3.83	204.34

From Table 6.7, it is evident that the average number of experts is similar across all values of K and the results seen in Table 6.6. This suggests that the values of K and L do not seem to have an effect on the preferred average number of experts in the HME model for this data set. The acceptance rates seen for varying values of K do not follow a consistent pattern. It is important to note that as the total number of jumps decreases, the number of splits and merges also decreases and is equal to approximately half of the total number of jumps. The latter means that the acceptance rates are more sensitive to small changes in the number of accepted/rejected jumps. Keeping that in mind, the rates seen here are in a similar range to the ones seen in Table 6.6. Finally, a consistent decrease in the run time of the RJ MCMC is observed for the total number of jumps of $1,000$ and below. Performing $1,000$ jumps takes 57.82 seconds longer than performing 50 jumps. We have observed in Table 6.6 that adding more jumps past the $1,000$ threshold

does not result in a significant run time increase, hence it is preferable to explore and propose more states in any given scenario.

6.9 Summary

So far, the following key benefits of the RJ MCMC for HME models have been discussed and illustrated:

1. Ability to accurately model changes in the data patterns.
2. Allowing for different variances across experts addresses heteroscedasticity.
3. The size of the HME tree can be selected automatically.
4. The addition of the informed jump generation algorithm vastly improves the reversible jump acceptance rates when compared to the naive reversible jump approach.
5. The choice of the frequency and number of reversible jumps does not seem to have an effect on the overall preferred fit of the HME model.
6. The addition of the reversible jump seems to improve mixing and convergence.

In this section, a method for automatic growing and pruning of the HME model tree has been proposed and assessed. The RJ MCMC, however, does not directly address changing the order in which the nodes are arranged in the model tree. This functionality is tackled by the proposed addition to the reversible jump MCMC methodology, which is discussed in the next chapter.

Chapter 7

Automatic Architecture Internal Adjustment

7.1 Introduction to Gate Swaps

So far, the methodology for automatically growing and pruning HME trees has been proposed and evaluated. In this chapter, the idea of changing the architecture of an existing HME model tree is discussed. As covered previously, the gate nodes in HME models perform the crucial task of partitioning a complex problem into smaller subproblems. The novel idea introduced here involves changing the order in which the splits occur after they have been selected.

The reversible jump methodology proposed in the previous section relies on the jumps being indeed *reversible*. In a binary tree, this means that any expert in the tree can be split into two new experts. The reversibility of the jump in turn means that the two experts that can be merged into a single expert must be siblings. That is, the reversible jump is only operating in the leaves of the tree.

Consider the example shown in Figure 7.1 once more in order to illustrate the potential scenario that the design of reversible jumps fails to address. Assume that the split at $G2$ is not a beneficial one. If the current state of the model tree is as depicted in the figure, two merges would need to occur before a new, potentially more beneficial split could be proposed instead. That is, firstly, $E4$ and $E5$ would need to be merged into a single expert $E4$ to replace the gate $G4$. Secondly, the newly formed $E4$ would need to be merged with $E1$ to form a single expert $E1$ and replace $G2$. At this stage, a new split could be proposed to the merged expert $E1$ to start over again. Since the proposed

reversible jump methodology allows for merging at the leaves only, two siblings such as an expert $E1$ and a gate $G4$ cannot be merged in one step.

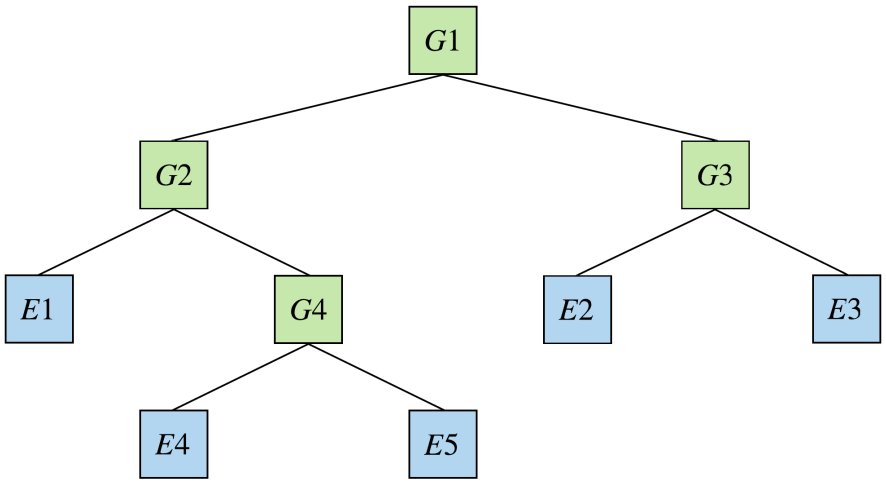


FIGURE 7.1: Illustration of an HME model with five experts equivalent to Figure 1.1.

Of course, there is no guarantee that the RJ MCMC will indeed lead to the merges outlined above, especially if the split at $G4$ provides a reasonable separation between $E4$ and $E5$. An alternative way of proposing a new split in the circumstances discussed involves altering the order of splits by swapping the gate nodes. The development of this additional step in the MCMC chain, called the *Gate Swap* (GS) algortihm, has a number of potential advantages. Firstly, it would make escaping the unfortunate previously made splitting decisions possible. The method also has the potential to improve the exploring of the model space and hence improve the mixing. When implemented together with the reversible jump, gate swaps would propose tree architectures that would have not been considered otherwise. The latter could result in a simpler architecture that yields a better overall fit. This section proposes a framework for swapping gate nodes as an additional step in the RJ MCMC chain.

An illustration of the proposed method is presented in Section 7.2. The steps of the GS algorithm are then formally outlined in Section 7.3, and an application is presented in Section 7.4. The GS algorithm is then evaluated on the same real-life data as the RJ algorithm in Section 7.5. Finally, some concluding remarks are made in Section 7.6.

7.2 Illustration of Gate Swaps for HME Models

Let us refer to the illustration of the HME model with five experts as shown in Figure 7.1 once more. Following the example from the previous section, let's say that one is interested in swapping gate $G2$ with gate $G4$. Firstly, a decision has to be made on which child of $G4$ should be replaced by $G2$. In our example, $G4$ has two children, so there are two options. If it is decided to replace $E4$, which is the first child of $G4$, the resulting tree will look like the one shown in Figure 7.2. On the other hand, replacing $E5$, the second child of $G4$, will result in a tree depicted in Figure 7.3.

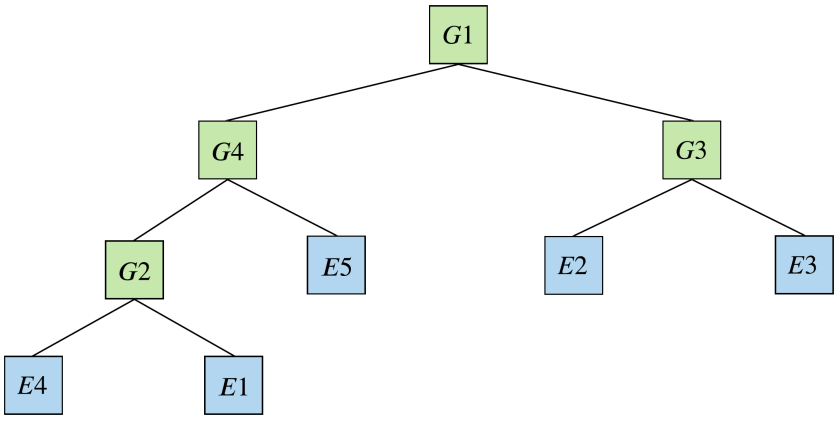


FIGURE 7.2: Illustration of swapping gates $G2$ and $G4$ from the HME model shown in Figure 7.1 by replacing $E4$.

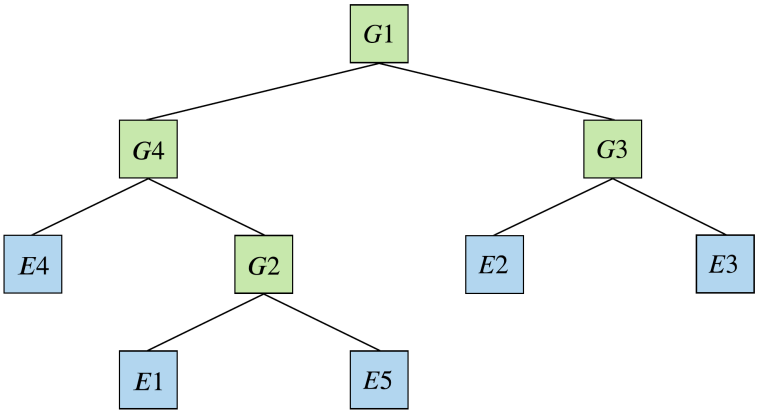


FIGURE 7.3: Illustration of swapping gates $G2$ and $G4$ from the HME model shown in Figure 7.1 by replacing $E5$.

In all cases of gate swaps, the decision on which child of the more junior gate is to be replaced has to be made. A reasonable starting point is to pick one child at random

with all children having an equal probability of being picked. In the above example of a binary tree, each child would be picked with a probability of 0.5.

It could be argued that the most important split is the very first one, i.e., $G1$, because it divides all observations into the smallest number of groups with the largest amount of observations in each. Gate swaps provide an option of escaping from a bad choice of the first split as well as subsequent splits. Let's now assume that we are interested in swapping the root gate $G1$ with its descendant gate $G2$ (Figure 7.1). As before, in the case of a binary tree, we have two ways of performing the swap. We could choose to replace $E1$, the first child of $G2$, by $G1$ as shown in Figure 7.4. Alternatively, we could choose to replace $G4$, the second child of $G2$, with $G1$ as shown in Figure 7.5.

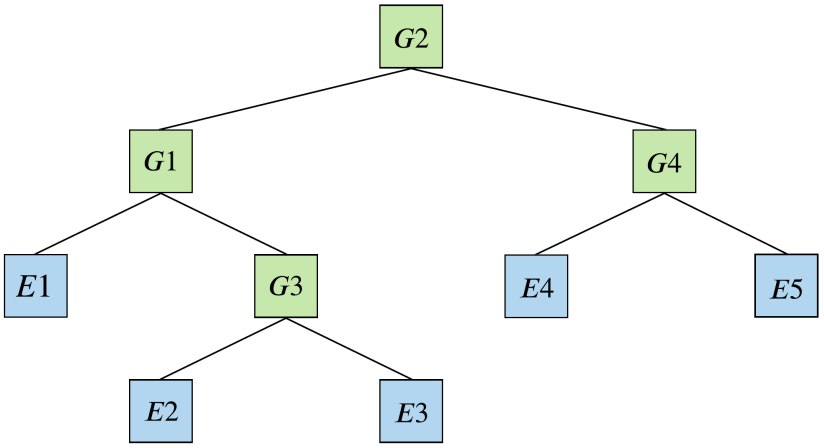


FIGURE 7.4: Illustration of two ways of swapping gates $G1$ and $G2$ from the HME model shown in Figure 7.1 by replacing $E1$.

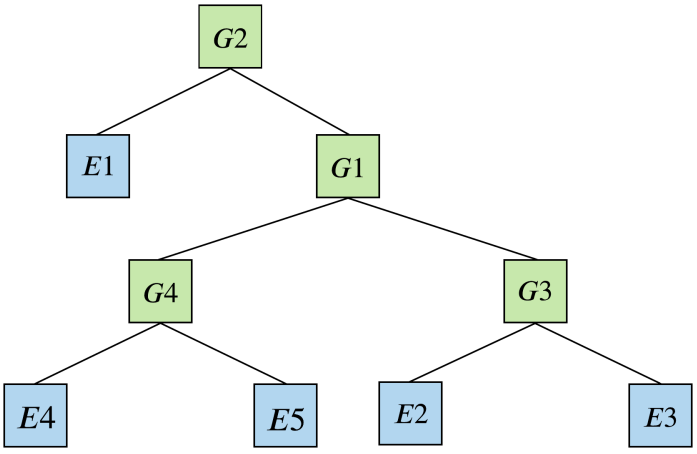


FIGURE 7.5: Illustration of two ways of swapping gates $G1$ and $G2$ from the HME model shown in Figure 7.1 by replacing $G4$.

The architecture shown in Figure 7.5 could be beneficial if there was one distinct subgroup present in the data. In that case, assigning those observations to $E1$ and separating them from the rest of the observations early on would be a substantial improvement on the original tree structure. The toy example used here illustrates how the hypothetical unfortunate split at $G2$ could be improved upon in four different ways, each resulting in a completely new tree architecture. The steps of the Gate Swap algorithm, that allows for such proposals, are formally outlined in the next section.

7.3 The Gate Swap Algorithm

The steps of the gate swap algorithm are as follows:

1. Check that there is more than one gate in the tree.
2. Select two gates in the tree, such that one gate is senior to the other, to be swapped at random.
3. Check which gate is more senior. Call the senior gate G and the junior gate G^*.
4. Record likelihood for the original tree
$$L(\boldsymbol{\phi}|\mathbf{y}) = \prod_{i=1}^{n} f(y_i|\mathbf{x}_i, \boldsymbol{\phi}) = \prod_{i=1}^{n} \left[\sum_{E \in \mathcal{E}} \pi_i^{(E)} f^{(E)} \left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)} \right) \right].$$
5. Randomly choose which child of G^* should be replaced by G.
6. Perform the swap and record the likelihood for the new proposed tree
$$L^*(\boldsymbol{\phi}|\mathbf{y}) = \prod_{i=1}^{n} f(y_i|\mathbf{x}_i, \boldsymbol{\phi}^*) = \prod_{i=1}^{n} \left[\sum_{E \in \mathcal{E}} \pi_i^{*(E)} f^{(E)} \left(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(E)} \right) \right],$$
where $\pi_i^{*(E)}$ for $E \in \mathcal{E}$ are the new path probabilities obtained after the gate swap.
7. Accept the swap with probability
$$\alpha = \min \left(1, \frac{L^*(\boldsymbol{\phi}|\mathbf{y})}{L(\boldsymbol{\phi}|\mathbf{y})} \right),$$
where $L(\boldsymbol{\phi}|\mathbf{y})$ and $L^*(\boldsymbol{\phi}|\mathbf{y})$ are results from steps 4 and 6 respectively.

It is evident that the acceptance probability, α, is made up solely of a ratio of model likelihoods. The latter, rather unusual in Bayesian setting, acceptance probability arises

from the fact that there are no modifications made to the existing model parameters or the size of the tree so the prior distributions are identical and cancel out. In fact, the only aspect of the tree that changes is the order in which its nodes are arranged and thus the resultant post-swap path probabilities.

Having outlined the steps of the GS formally, an illustration of the GS algorithm on a simulated data is presented next.

7.4 Application of the Gate Swaps Algorithm on Simulated Data

Consider a simulated three-dimensional example of an HME model with three Gaussian experts in the tree as depicted in Figure 7.6. In this model, z is the response variable while x and y are the explanatory variables. Let us assume that the first split, represented by $G1$, separates $E1$ (black points) from the remaining experts by cutting x at 0. The second split, governed by $G2$, then further separates $E2$ (red points) and $E3$ (green points) by cutting y at 0 for $x < 0$. In such case, a single model is fitted to the black points which fails to capture the abrupt change in the response variable values happening at around $y = 0$. The latter also leads to the fitted plane missing the extreme values of y.

At this stage, the RJ MCMC could improve on the fit in two ways. Firstly, $E1$ could be split into two experts, which would solve the issue, however, create an unnecessary extra expert that would be likely to get merged in the future. The same outcome could be achieved in reverse, that is, $E2$ and $E3$ could be merged first followed by the split of $E1$. Of course, there is no guarantee that the reversible jump moves would yield the discussed moves in the specified order. This means that improving this fit with the RJ MCMC would require at least two accepted reversible jump steps.

Using the GS algorithm outlined in Section 7.3, consider a single proposed step suggesting swapping $G1$ and $G2$ by replacing the first child of $G2$, which is $E2$, with $G1$. Such swap is accepted, with the initial model log-likelihood value of -345.725 and the proposed model likelihood value of -337.821 yielding an acceptance probability of $\alpha = \min(1, \exp(7.90)) = 1$. The resulting tree architecture, allocation variables, and the fitted plane are depicted in Figure 7.7. It is evident that in one step, the gate swap was able to propose an architecture with the same amount of experts, but a more advantageous split. After as little as 5 MCMC iterations post the acceptance of the gate swap, the fitted plane shown in plot (iii) captures the previously discussed abrupt change at $y = 0$ for $x > 0$ as well as fits the data well at the extremes of y.

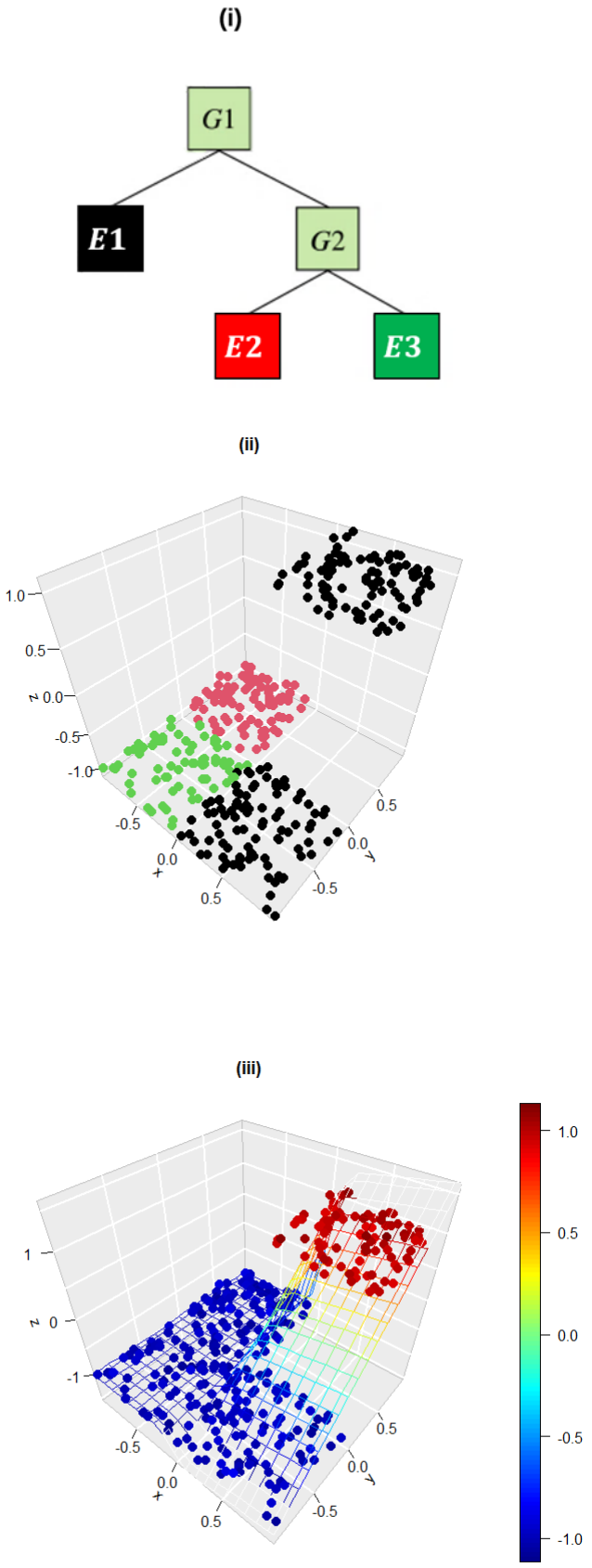


FIGURE 7.6: An example of a HME model with three experts in the tree. (i) depicts the underlying tree architecture; (ii) shows the assignment of the observations to the three experts; (iii) shows resulting fitted plane.

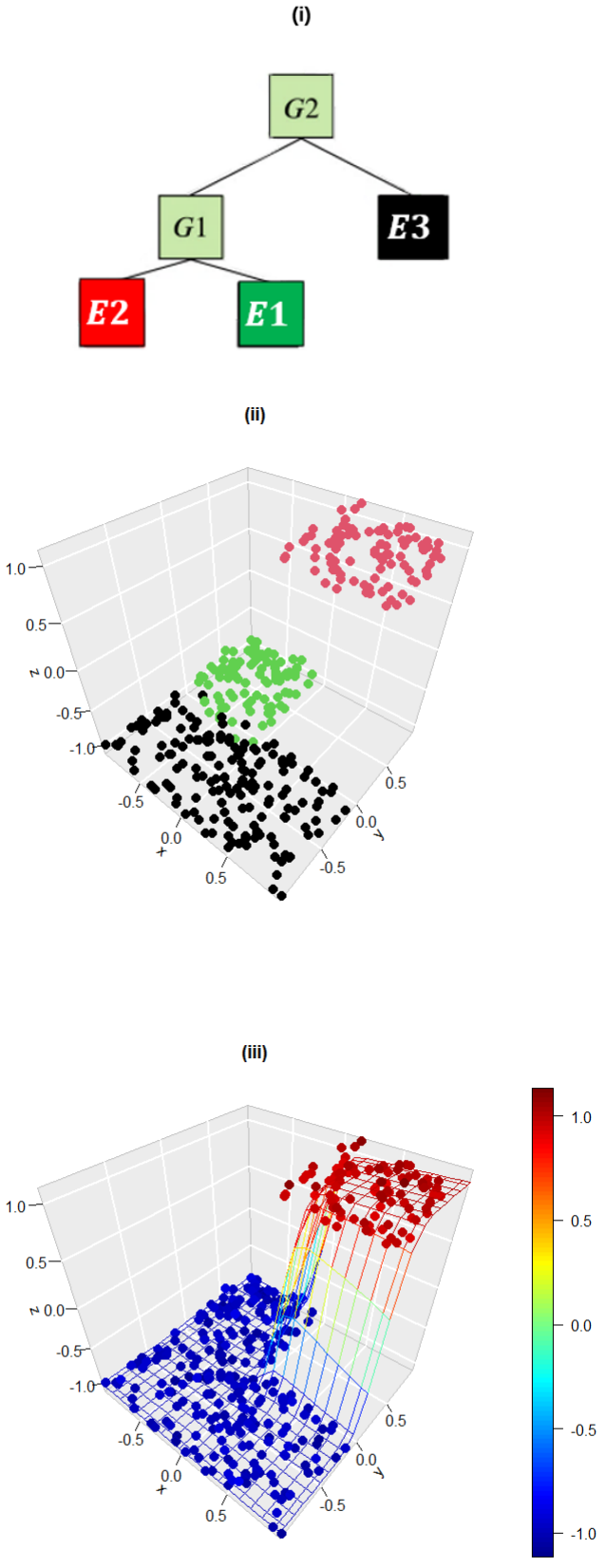


FIGURE 7.7: An example of a HME model with three experts in the tree after the proposed swap is accepted. (i) depicts the underlying tree architecture; (ii) shows the assignment of the observations to the three experts 5 MCMC iterations post swap; (iii) shows resulting fitted plane 5 MCMC iterations post swap.

The toy example presented in this section illustrates how the addition of the GS step to the RJ MCMC can benefit the overall model fit, allow for escaping unfortunate splits and consider architectures that might have been missed otherwise. All of the above has the potential to speed up the convergence as well as improve mixing. The GS algorithm is next evaluated on the motorcycle crash helmet data set alongside the reversible jump algorithm.

7.5 Evaluation of the RJ GS MCMC for Motorcycle Accident Data

7.5.1 Introduction to the RJ GS MCMC Evaluation

In this section, the GS algorithm is evaluated on the motorcycle accident data set. In the first scenario, an HME model is fitted using the reversible jump MCMC methodology and is equivalent to the fit discussed in Section 6.8.2. In the second scenario, in addition to a reversible jump proposed every 10th iteration, a swap is also proposed every 100th iteration. Such frequency for proposed swaps is chosen as a starting point for the detailed swap frequency investigation undertaken in Section 7.5.4. To insure comparability, both MCMC chains are run for $5,000$ iterations with the first 500 runs discarded as burn-in. In all evaluations performed in this section, the convergence of the chains is assessed by visual inspection of the predictions as outlined in Section 4.3.4.2. Both models are then used to make predictions and compared. Due to a small number of observations available (133), the data is not split into training and test sets so the predictions are made on the whole data set. The prior distributions used for both cases here are stated in Section 5.5.2 and applied to the motorcycle accident data throughout the thesis. In both cases, the starting model tree architecture consists of one expert, which means that the architecture selection is performed automatically.

7.5.2 Reversible Jump Gate Swap MCMC Results

The results of both the RJ MCMC and the RJ GS MCMC runs are summarised in Table 7.1. It is important to remember that gate swaps are only possible when there is more than one gate present in the model tree, thus the total number of swaps proposed is not equal to the number of MCMC iterations divided by the frequency of swaps. Firstly, it is evident that the acceptance rate for the gate swaps is very high (43.48%). High acceptance rate might be pointing to moves that are not leading to dramatic changes in the likelihood of the tree. A large number of such moves, however, is undoubtedly

improving mixing in the architecture, because a high number of accepted gate swaps indicates considering a larger number of model tree architectures.

It can also be seen that the acceptance rates of the reversible jumps have increased after the introduction of the gate swaps. This result is expected since the gate swaps change the architecture of the tree which then leads to new potential splits and merges to be considered. The mean squared error (MSE) reveals that the RJ MCMC slightly outperforms RJ GS MCMC. The latter coupled with high acceptance rates might indicate that, while gate swaps improve mixing in architecture, their acceptance also causes greater disturbance to the architecture of the tree thus creating less accurate predictions during the time it takes for the reversible jumps and the MCMC to improve upon the accepted post gate swap state. Having evaluated the predictive performance of the two methods as well as acceptance rates, we proceed by investigating the overall model fit.

TABLE 7.1: Acceptance rates for the implementation of reversible jump only (equivalent to Table 6.5) and with the addition of gate swaps algorithm for motorcycle accident data.

	Splits	**Merges**	**All Jumps**	**Swaps**
RJ MCMC				
Number Proposed	257	243	500	-
Number Accepted	21	18	61	-
Acceptance	8.17%	7.41%	7.8%	-
Mean Squared Error			0.2081	
RJ GS MCMC				
Number Proposed	244	256	500	46
Number Accepted	37	35	70	20
Acceptance	15.16%	13.67%	14.40%	43.48%
Mean Squared Error			0.2301	

The recorded predictions for the RJ GS MCMC are shown in Figure 7.8. It can be seen that the majority of predictions track the data points appropriately, which is also confirmed by the average predictions (thick red line). Comparing the predictions produced with the addition of gate swaps to those produced without (Figure 6.17), it is evident that there are smoother transitions between the experts present for the RJ GS MCMC case. The distribution of the number of experts across the MCMC runs is depicted in Figure 7.9. In agreement with the RJ MCMC (Figure 6.18), the majority of the time is spent exploring the states with 4 experts. On the other hand, trees with 3 experts are investigated notably more when the gate swaps are used. This result is expected

because gate swaps cause disturbance and changes in the overall tree architecture hence encouraging some of the decisions made previously to be reconsidered.

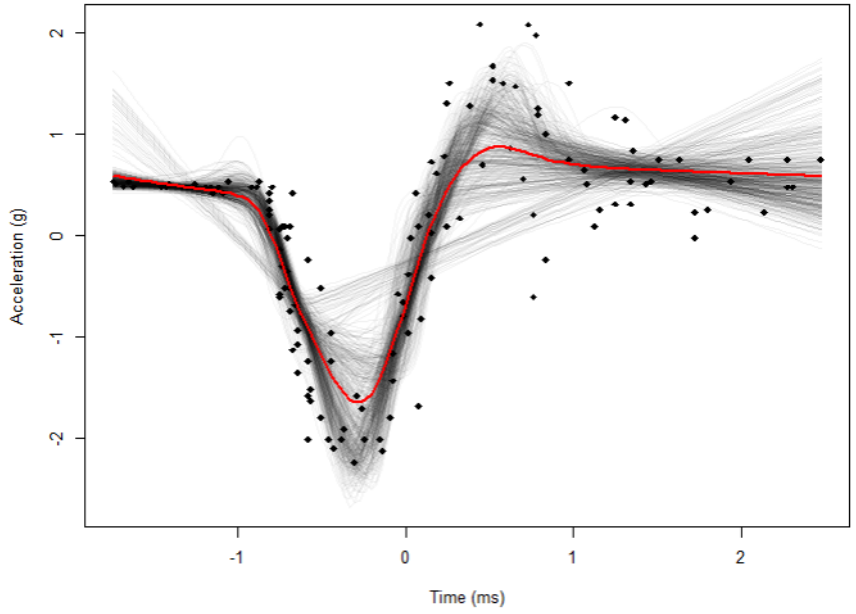


FIGURE 7.8: RJ GS MCMC predictions with initial start of 1 expert for motorcycle accident data. Every 10-th prediction shown. Average predictions shown in red.

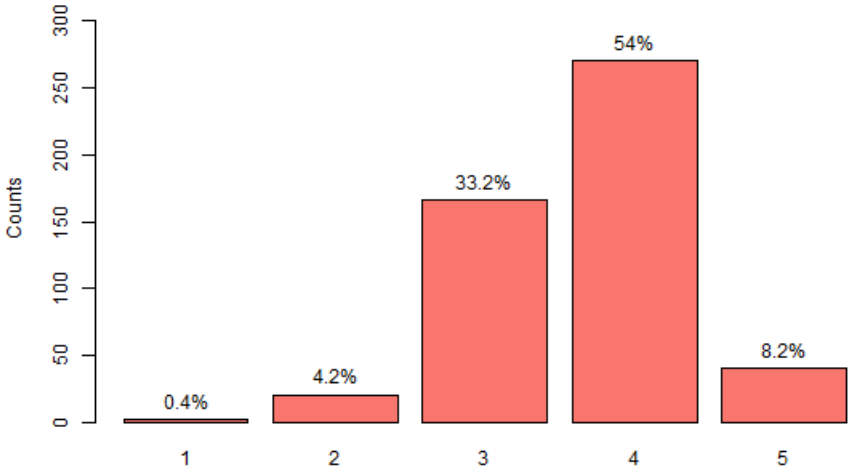


FIGURE 7.9: Distribution of the number of experts in the RJ GS MCMC chain with initial start of 1 expert for the motorcycle accident data.

Figure 7.10 showcases the number of experts in the HME tree after each reversible jump. Iterations at which gate swaps occur are denoted by dashed lines. Those are further color coded for accepted (green) and rejected (red) swaps. There appears to be no clear pattern in the acceptance of the proposed swaps. Similarly as seen for the RJ MCMC, the chain does not appear to settle on a specific number of experts in the tree and, after the initial tree growing, explores states with two to five experts in them.

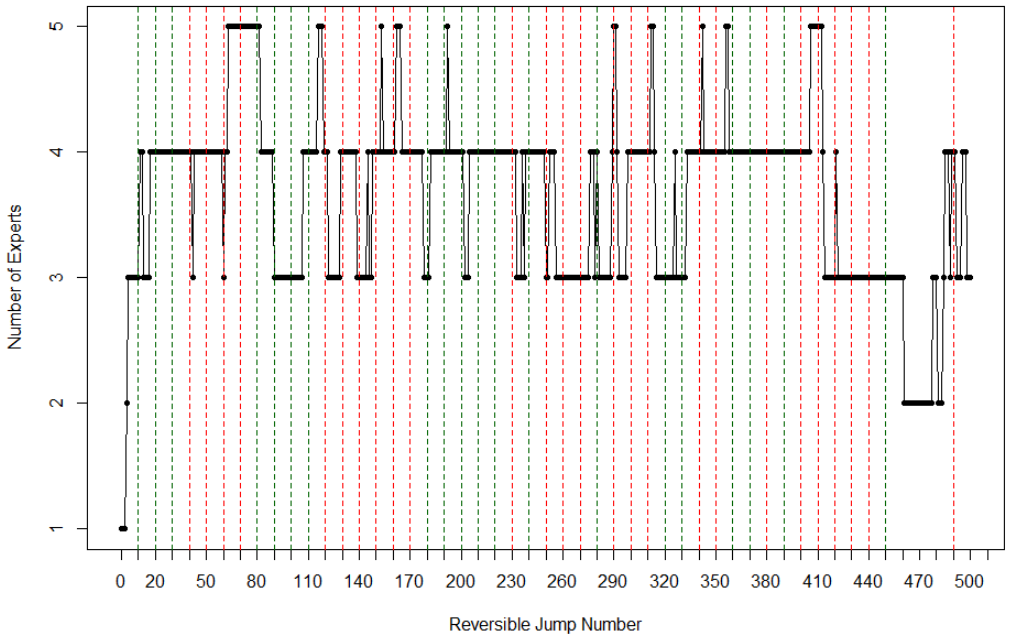


FIGURE 7.10: Number of experts in the tree after each reversible jump proposal for the motorcycle accident data. Gate swap proposals are marked as dashed lines with green indicating an accepted and red indicating a rejected gate swap proposal.

Having investigated the key features of the fitted model, the effect of gate swaps on mixing and convergence of the MCMC chains is investigated next.

7.5.3 Mixing and Convergence

Once again, the random starting point evaluation undertaken in Section 5.5 is revisited. Please refer to Section 6.8.3 for the results obtained by randomly initialising three MCMC chains with and without the reversible jump. The same starting parameter values are now used for the three RJ GS MCMC chains resulting in the predictions shown in Figure 7.11.

As seen before, the addition of the gate swaps has produced a smoother overall fit. Although the individual average predictions appear to track data well for each of the three chains, there is more between-chain variability seen at the points of change. Once a swap is accepted, the overall architecture of the tree is disturbed, which might result in more time spent exploring smoother separations between experts before arriving to the more confident and steeper separations, which is evident for green and red chains. Given the observed minor between-chain variability, a formal assessment of convergence is required.

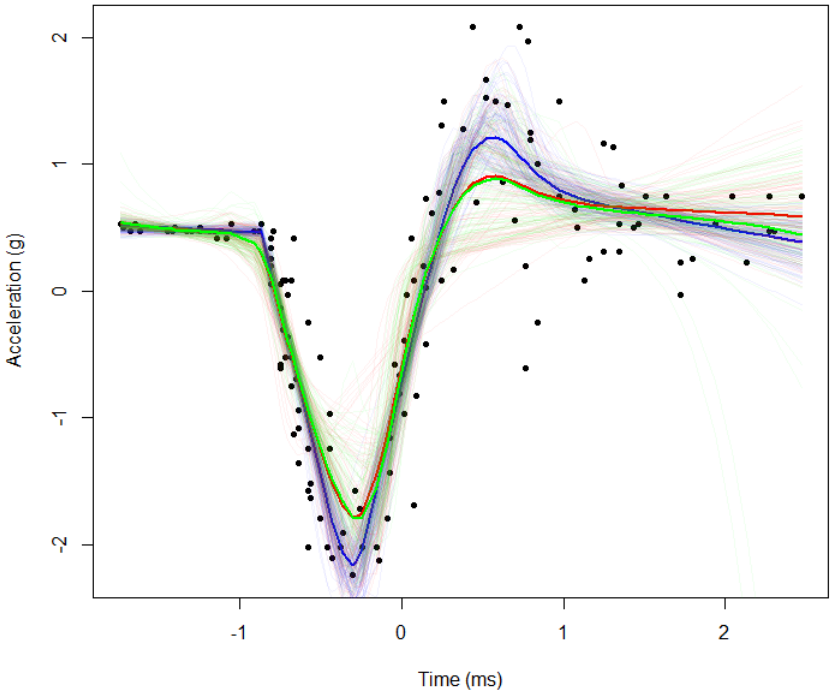


FIGURE 7.11: RJ GS MCMC predictions obtained with the three starting points reflected by colour for the motorcycle accident data. The thick lines represent the average predictions while the thin lines represent every 10-th prediction.

The Gelman-Rubin diagnostic yields a value of 1.01 thus further improving on the one seen for the RJ MCMC and strongly suggesting that convergence has been reached (see Table 7.2).

TABLE 7.2: The potential scale reduction factor (PSRF) for the three randomly drawn parameter starting values for MCMC, RJ MCMC and RJ GS MCMC for the motorcycle accident data.

	MCMC	**RJ MCMC**	**RJ GS MCMC**
PSFR	5.02	1.13	1.01

Having identified that gate swaps can further improve on mixing and convergence, the frequency of swaps is considered next.

7.5.4 Frequency of Swaps in Automatic Architecture Selection for Motorcycle Accident Data

After each swap is accepted, the reversible jumps and parameter updates improve on the proposed state. Thus it is of interest to understand what effect the frequency of swaps has on acceptance rates. The less frequent the swaps, the more time is left for improving on the accepted state. On the other hand, more frequent swaps allow for considering more architectures overall. Let's say that swaps are performed every R-th iteration of RJ GS MCMC and a single reversible jump is still proposed every 10-th iteration. The results for varying values of R are presented in Table 7.3.

TABLE 7.3: Acceptance rates and run-time for a varying gate swap frequency in the RJ GS MCMC for the motorcycle data. The first scenario was re-run to measure the run-time under the same conditions across all tests.

R	**Total Swaps Proposed**	**Swaps Acceptance**	**Overall Reversible Jump Acceptance**	**Average Number of Experts**	**Run Time (s)**
-	0	-	7.80%	3.89	249.67
50	98	36.73%	16.40%	3.69	194.80
100	46	43.48%	14.40%	3.66	226.52
200	24	37.50%	9.40%	4.02	238.70

Firstly, it is evident that the swaps acceptance rates are consistently high across all values of R. In agreement with the results seen in Table 7.1, the acceptance of reversible jumps increases once the gate swaps are active, however, there is no clear pattern linking the reversible jump acceptance and the frequency of swaps. Looking at the average number of experts for $R = 200$ versus the other cases, it appears that the total number

of gate swaps proposed may be associated with a slightly smaller average number of experts in the trees. This result is expected since the addition of gate swaps encourages more moves to happen overall by adding the swaps as well as increasing the number of accepted reversible jumps. Moreover, as seen in Section 7.4, an unfortunate split at the root can cause additional experts to appear in order to counteract the effect, while a successful swap may solve the problem hence resulting in a smaller number of experts overall. Finally, investigating the run-time of each case reveals that it is associated with the average number of experts rather than with the number of swaps. This finding appears intuitive since each additional node in the tree increases the number of sampled parameters as well as results in more potential jumps and swaps. Finally, in this case, swaps do not appear to be expensive in terms of the additional run-time required.

7.6 Summary

Overall, it appears that the addition of the GS algorithm does have merit. A larger number of tree architectures has been considered hence improving mixing in architecture. The improved exploration of tree architecture space is also confirmed by the high swap acceptance rates. Furthermore, Gelman-Rubin diagnostic is used to showcase the improvement in convergence for the unfortunate starting point experiment. It has also been shown that running RJ GS MCMC does not result in a notable increase in run-time when compared to the RJ MCMC. Moreover, the addition of gate swaps has been associated with a smaller average number of experts in the tree hence making the model fit simpler. The design of the GS algorithm would arguably benefit more complex and deeper trees than those considered so far as the addition of the gate swaps allows for escaping unfortunate splits, which are in turn less likely to occur in shallow trees.

Having outlined and evaluated the proposed methodology for the second type of automatic architecture selection, as an addition to methodology covered in Chapter 6, the next chapter covers two competitors of the HME models and assesses their performance against HME model fit.

Chapter 8

Competitors for HME

8.1 Introduction

This chapter discusses two competitors of HME models - Generalised Additive Model (GAM) and Bayesian Additive Regression Trees (BART). A fundamental model with a response variable y_i and predictor variables $\mathbf{x}_i = (x_{1i}, ..., x_{pi})$ is considered throughout this chapter:

$$y_i = f(\mathbf{x}_i) + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. In the subsequent sections, the function $f(\cdot)$ is defined for GAM (Section 8.2) and BART (Section 8.3). The model-fitting process as well as pros and cons of each method are also discussed in the dedicated sections. The two competitors are then fitted to the motorcycle accident data set in Section 8.4 and compared to the RJ MCMC HME model fit produced in Chapter 6. The resulting models are then assessed in the context of heteroscedasticity (Section 8.4.1), accuracy of the fitted values (Section 8.4.2), and interpretability (8.4.3).

8.2 GAM

8.2.1 Definition of GAM

The generalised Additive Model (GAM) was originally developed by Hastie and Tibshirani (1990) and is a generalisation of the Generalised Linear Model (GLM), which is

in turn a generalisation of a standard linear model with response variable y_i, predictor variables $\mathbf{x}_i = (x_{1i}, ..., x_{pi})$ and the corresponding model parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, ...\beta_p)^T$:

$$\begin{aligned} y_i &= f(\mathbf{x}_i) + \epsilon_i \\ &= \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i \\ &= \mu_i + \epsilon_i \end{aligned}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ and $\mathbb{E}(y_i|\mathbf{x}_i) = \mu_i$. The idea behind the GLM is to express the response variable y_i as a linear function of the predictor variables $\mathbf{x}_i$, which is done by using a link function $g(\cdot)$ that is related to the mean μ_i as follows

$$\begin{aligned} g(\mu_i) = \eta_i = \mathbf{x}_i\boldsymbol{\beta} = \beta_0 + \beta_1 x_{1i} + ... + \beta_p x_{pi} \\ \mathbb{E}(y_i|\mathbf{x}_i) = \mu_i = g^{-1}(\eta_i), \end{aligned} \tag{8.1}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ...\beta_p)^T$ are the model parameters associated with the predictors (Nelder and Wedderburn, 1972). The general expression showcases how a link function $g(\cdot)$ relates to the mean μ_i, or, on the other hand, how the expected value $\mathbb{E}(y_i|\mathbf{x}_i)$ relates to the so-called *linear predictor*, η_i. It is evident that a standard linear model is a GLM with the link function $g(\mu_i) = \mu_i$, which is called the *identity link* function.

GAM, in turn, extends the GLM framework to allow for nonlinear forms of the explanatory variables in the model. The latter is achieved through an additive modeling technique, in which the response variable depends on unknown smooth functions of the explanatory variables. In other words, the model coefficients from (8.1) are simply replaced with flexible functions, $s(\cdot)$, that allow for nonlinear relationships:

$$g(\mu_i) = \beta_0 + s(x_{1i}) + ... + s(x_{pi}). \tag{8.2}$$

In the case of the response variable following Gaussian distribution, the link function used in (8.2) is the identity link function, that is

$$\begin{aligned} y_i = \beta_0 + s(x_{1i}) + ... + s(x_{pi}) + \epsilon_i. \\ \mathbb{E}(y_i|\mathbf{x}_i) = \mu_i = \beta_0 + s(x_{1i}) + ... + s(x_{pi}), \end{aligned} \tag{8.3}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The smooth functions $s(\cdot)$, can be represented by local regression (loess) (Fox and Weisberg, 2018), regression splines such as B-splines or P-splines, thin plate splines (Perperoglou et al., 2019) or smoothing splines (Wahba, 2011), which are used in this chapter.

8.2.2 Smoothing Splines

Spline functions consist of piecewise polynomials that are joined together at pre-defined subintervals. The points at which the joins occur are known as *knots*. The degree of the spline is equivalent to the degree of the polynomial. For example, a spline degree equal to 1 is equivalent to a chain of line segments while a spline of degree 3, also known as *cubic spline*, is equivalent to a chain of third-degree polynomial segments. For a polynomial of degree p, the spline function can be written as

$$\begin{aligned} f(x_i) &= \sum_{j=0}^{p} B_j b_j(x_i) \\ &= B_0 + B_1 x_i + B_2 x_i^2 + ... + B_p x_i^p, \end{aligned} \tag{8.4}$$

where $b_j(x)$ are referred to as basis functions and B_j as basis coefficients. For a cubic spline, $p = 3$ with $b_0(x) = 1$, $b_1(x) = x$, $b_2(x) = x^2$ and $b_3(x) = x^3$.

The aim of spline smoothing is to fit a smooth, flexible function $f(x)$ that minimises the residual sum of squares defined as $RSS = \sum_i^n (y_i - f(x_i))^2$. It is evident that RSS is minimised when $\hat{f}(x_i) = y_i$, which is a regression that interpolates the points and hence is highly likely to result in overfitting. To avoid this, a *smoothing spline* fits such a smooth function by minimising the *penalised* residual sum of squares defined as

$$RSS(\lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_a^b f''(t) \partial t, \tag{8.5}$$

where λ is a fixed *smoothing* parameter, $a \leq x_1 \leq ... \leq x_n \leq b$, and $f''(\cdot)$ denotes the second derivative of function $f(\cdot)$. For smoothing splines, the observed unique explanatory variable values, i.e., $x_1, ..., x_n$, are the knots.

After taking a closer look at (8.5), it becomes clear that the first term is equivalent to the unpenalised residual sum of squares and it measures closeness of function $f(x)$ to the observed data. The second term, on the other hand, penalises the curvature in the function thus controlling the smoothness of the fit (Denceaux, 2016). In a special case, where $\lambda = 0$, there is no constraint on the smoothness of $f(x)$. As λ increases from

zero to infinity, $f(x)$ is forced to be smoother with respect to the penalised residual sum of squares (Helwig, 2021). In simple terms, the larger the value for λ, the bigger the penalty. The value for λ can be chosen using methods such as cross-validation, Akaike information criterion (AIC), or Bayesian information criterion (BIC) (for more information on selecting the smoothing parameter value see Wood, 2008).

For smoothness penalty $\lambda \int_a^b f''(t)\partial t$ from (8.5) , the unique minimiser solution is a cubic spline (for the proof see Green and Silverman, 1994), which can be obtained as per (8.4) with $p = 3$. This means that the smoothing spline $f(x)$ is a piecewise cubic polynomial in each interval (x_i, x_{i+1}) (for ordered x_i). The polynomial pieces fit together at the points x_i in such a way that the function $f(x)$ and its first and second derivatives are continuous at each knot and hence on the whole of $[a, b]$. In order to ensure that the function is linear beyond the boundary knots, *natural cubic splines* further require that the value of the second and third derivatives of $f(x)$ is equal to zero at the start and end points a and b.

A broadly used method for fitting GAM models is presented next.

8.2.3 GAM Model Fitting

As discussed in the previous section, GAMs consist of multiple smooth functions. Hence, model fitting for GAMs means simultaneously estimating these smooth functions (Larsen, 2015). One of the methods used for fitting GAMs, used in `R` package `gam`, is called the *backfitting* algorithm (Buja et al., 1989). The algorithm is intuitively easy to understand as it is based on solving linear equations (Xia, 2009). Denote $\mathbf{y} = (y_1, ..., y_n)^T$ and $\mathbf{x}_j = (x_{ji}, ..., x_{jn})^T$ for $j = 1, ..., p$. The backfitting algorithm steps for estimating (8.3) are then as follows:

1. Initialise $\beta_0^{(0)} = \mathbb{E}(\mathbf{y})$, $s^{(0)}(\mathbf{x}_1) = ... = s^{(0)}(\mathbf{x}_p) = 0$, $t = 0$.
2. Iterate
   (a) $t = t + 1$.
   (b) For $j = 1, ..., p$ calculate
      i. The j-th partial residuals as

$$R_j = \mathbf{y} - \beta_0 - \sum_{k=1}^{j-1} s^{(t)}(\mathbf{x}_k) - \sum_{k=j+1}^{p} s^{(t-1)}(\mathbf{x}_k)$$

.

ii. Estimate the new value of the j-th smooth function as

$$s^{(t)}(\mathbf{x}_j) = \mathbb{E}(R_j|\mathbf{x}_j).$$

(c) Calculate the Residual Sum of Squares (RSS) as

$$RSS = \frac{1}{n}\left(\mathbf{y} - \sum_{j=1}^{p} s^{(t)}(\mathbf{x}_j)\right)^2.$$

Stop if RSS fails to decrease or satisfies the convergence criterion.

The above algorithm can be adapted to suit non-Gaussian response variables by introducing weights to the smoothers (Xia, 2009). For GAMs with links other than the identity link, other procedures such as the General Local Scoring Algorithm can be used (Hastie and Tibshirani, 1986).

8.2.4 GAM Model Features

In addition to allowing for more flexibility in modeling the predictor variables, GAM also offers an interpretability advantage over GLM. Namely, due to the model being additive, one can visualise the individual fitted smooth functions of each predictor. The latter, however, becomes more complicated when fitting the smooth functions to the interaction terms in the model (see Chang et al. (2020) for a detailed assessment of GAM interpretability). When it comes to heteroscedasticity, the GAM approach does not account for it by design and hence simple approaches, such as taking the log transform of the response variable, are often used. Given that the homoscedasticity assumption is often violated in real-life applications, care must be taken when managing heteroscedasticity (Rosopa et al., 2013). Lastly, due to the flexible nature of splines, GAMs are prone to over-fitting, which should be taken into account when assessing the model fit (Wood (2008) discusses penalisation methods to aid the issue). In cases where GAM models fail to capture the abrupt changes in the response, alternative nonparametric approaches such as change-point regression might be of value (Shaban, 1980).

GAMs are often described as a flexible and purely frequentist framework (Miller, 2019) making them a notable non-Bayesian competitor to HME models. It is anticipated that the advantages of increased interpretability and accounting for heteroscedasticity in the response offered by the HME over GAM have the potential to tip the scales in favor of the former. Contrary to GAM, the second competitor comes from Bayesian school of thought and is discussed next.

8.3 BART

8.3.1 Definition of BART

Developed by Chipman et al. (2010b), Bayesian Additive Regression Trees (BART) is a nonparametric Bayesian regression approach that approximates the response variable by a sum of regression trees, which split the problem space into smaller subproblems and provide a numeric output at the leaves of the tree. The latter is achieved by approximating the mean of the response variable y_i, given the predictor variables $\mathbf{x}_i = (x_{1i}, ..., x_{pi})$, by a sum of m regression trees, i.e., $E(y_i|\mathbf{x}_i) \approx h(\mathbf{x}_i) \equiv \sum_{j=1}^{m} g_j(\mathbf{x}_i)$, where each g_j denotes a regression tree. The model can then be written as

$$y_i = h(\mathbf{x}_i) + \epsilon_i \tag{8.6}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The idea behind BART is to fit a number of such trees, known as *weak learners*. The weak learners are kept simple by imposing a strong prior that keeps the effects of each tree small. Each individual tree might only offer a poor prediction for the response, however, the sum of the trees can provide an accurate prediction, where each tree explains a small and different proportion of $f(\cdot)$ (Chipman et al., 2010b). Similar ideas have been used in ensemble methods in the frequentist context such as boosting (Freund and Schapire, 1997), bagging (Breiman, 1996), or random forests (Breiman, 2001). Boosting relies on subsequently fitting single trees to the data variation not explained by the previous trees. On the other hand, bagging and random forests create a large number of independent trees, which are then used to make predictions that are averaged. Next, the methods of bagging and random forests are explained in more detail.

8.3.2 Bagging and Random Forests

The bagging algorithm, first introduced by Breiman (1996), incorporates the bootstrapping sampling technique, which uses random sampling with replacement to create diverse samples (Efron, 1979). The idea lies in repeatedly drawing large numbers of smaller samples of the same size from the original sample. The resulting bootstrap samples are then used to fit the models independently and in parallel. Finally, the outputs created by the individual models are averaged to obtain a more accurate estimate of the response variable. Each individual model might overfit the data and the hope is that this can be averaged out.

Random forests are an extension of the bagging algorithm, which also randomly selects a subset of the explanatory variables to be used in each sample resulting in less correlated models (Breiman, 2001) thus further reducing the variance of the aggregated estimate. The number of covariates to be randomly selected is a tuning parameter that can be determined by trying different values and using cross-validation to pick the optimal one (Stone, 1974).

The appeal of bagging methods lies in its ease of algorithm implementation as well as the resulting reduction in variance within a learning algorithm (Buja and Stuetzle, 2006). On the other hand, as the number of dimensions and iterations grows, the intensive resampling can lead to the algorithm being computationally expensive (Bühlmann and Yu, 2002). The latter drawback is improved upon by the random forest algorithm, which reduces the problem dimensionality. Finally, a well-known limitation of bagging methods lies in the loss of model interpretability caused by averaging the individual learner outputs.

Typically trees are used as base learners in these algorithms, however, other choices would also be possible. These algorithms are thus sum-of-tree models just like BART. BART is a successful attempt of porting the idea to the Bayesian context. An overview of the BART model fitting process is presented next.

8.3.3 BART Model Fitting

The BART model is usually fitted using Bayesian backfitting MCMC, which in itself is a Gibbs sampler that takes advantage of a few key residual-related observations (see Section 4.3.2 for more details on the Gibbs sampler). Following Chipman et al. (2010b), let T_j denote each binary regression tree and M_j denote the associated terminal node parameters. The sum-of-trees model (8.6) can then be expressed as

$$y_i = \sum_{j=1}^{m} g(\mathbf{x}_i|T_j, M_j) + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The posterior distribution of the model parameters can thus be written as

$$f((T_1, M_1), ..., (T_m, M_m), \sigma|\mathbf{y}),$$

where $\mathbf{y} = (y_1, ..., y_n)^T$. Let us further denote $T_{(-j)}$ as the set of all trees in the sum except the j-th one, and similarly define $M_{(-j)}$ to be the set of model parameters excluding the ones belonging to the j-th tree. The Gibbs sampler should thus perform m successive draws from

$$(T_j, M_j)|T_{(-j)}, M_{(-j)}, \sigma, \mathbf{y} \tag{8.7}$$

for $j = 1, ..., m$ followed by a straight-forward draw of σ. The conjugate inverse chi-square distribution prior is used for σ resulting in a simple draw from the full conditional posterior inverse gamma distribution (see Hastie and Tibshirani (2000) for details).

A key observation is made in order to implement the draws as per (8.7). The conditional distribution of the j-th tree depends on the joint distribution of all the remaining trees only through partial residuals defined as

$$R_j = \mathbf{y} - \sum_{k \neq j} g(\mathbf{x}|T_k, M_k),$$

where $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_p)$ and $\mathbf{x}_j = (x_{ji}, ..., x_{jn})^T$. Thus, the m draws as per (8.7) are equivalent to

$$(T_j, M_j)|R_j, \sigma, \tag{8.8}$$

which is simply the posterior distribution of a single tree model, where R_j is the response. Since a conjugate Gaussian prior is used for M_j, the posterior distribution for T_j

$$f(T_j|R_j, \sigma) \propto f(T_j) \int f(R_j|M_j, T_j, \sigma) f(M_j|T_j, \sigma) \partial M_j \tag{8.9}$$

can be obtained in closed form. Each draw from (8.8) can then be carried out in two steps

$$T_j|R_j, \sigma$$
$$M_j|T_j, R_j, \sigma.$$

An elaborate Metropolis-Hastings algorithm is used to draw T_j (Chipman et al., 1998). Integrating out M_j in (8.9) results in avoiding jumps between a varying number of dimensions, and thus, the need for the reversible jump algorithm implementation. M_j is

sampled by independently drawing from a Gaussian distribution centered at the output of the terminal node.

Having outlined the model fitting process, an overview of the model features is presented next.

8.3.4 BART Model Features

At each MCMC iteration, BART produces a draw from the posterior distribution of the response variable, i.e. produces predictions. The additive nature of the model means that BART implementations cannot usually offer a single model object from which fits and summaries may be extracted hence sacrificing the interpretability of the model (Chipman and McCulloch, 2016). It is evident from the expression of BART model density (8.6) that there is a single error variance applied across all observations, which does not directly address potential issues when dealing with a heteroscedastic response.

BART is a Bayesian divide-and-conquer approach model which makes it the obvious competitor for HME. The architectural difference between the two lies in the fact that BART fits a large number of simple trees that are then combined together while the HME is a one-tree model. BART is an additive model with a high predictive power recorded across multiple applications including the prediction of trip durations in transportation (Chipman et al., 2010a), somatic prediction in tumor experiments (Ding et al., 2011), biomarker discovery in proteomic studies (Hernandez et al., 2015) as well as others. Tan and Roy (2019) state that the success of BART has led to researchers using it as the standard reference model for comparison when proposing new statistical or prediction methods. Given the evident predictive power of BART, as well as over a decade of development, the model is bound to be challenging to outperform when it comes to prediction. The key differentiating point offered by HME models is their interpretability which coupled with accurate predictions could potentially form notable competition for BART.

8.3.5 BART Extensions

There have been a number of BART extensions developed since the introduction of the modeling technique in 2010. Linero and Yang (2017) discuss the development of probabilistic splits in the BART models. The latter extension is also known as Soft Bayesian Additive Regression Trees (SBART). It is proposed to substitute the deterministic path followed by the input variables with a probabilistic path. For SBART, probabilities of going down the tree in each direction can be thought of as weights, which are observation

specific and depend on the selected bandwidth parameter and the cut-off points drawn from the proposal distribution. The individual weak learners in SBART are thus similar to shallow HME trees with differing definitions for the probabilistic splits.

The problem of sparsity, i.e., the number of predictors being larger than the number of observations, for BART is tackled by Linero (2018). The construction of Dirichlet priors that adapt to sparsity in the input variables is proposed. It is also demonstrated that the proposed method allows for a fully Bayesian approach to variable selection.

The previously discussed poor mixing of MCMC samplers for tree-based models is also noted and addressed for Bayesian Regression Trees in Pratola (2016). Issues such as local mode stickiness and poor mixing are said to stem from inefficient MH algorithm proposals. Improved proposals, called the tree rotation proposal and rule perturbation proposal, are developed and demonstrated to be effective in improving mixing.

It has been discussed in the previous section that the unmodified BART does not directly address potential issues arising from a heteroscedastic response. Pratola et al. (2020) address this limitation by developing a nonparametric heteroscedastic elaboration of BART, called HBART. In addition to the mean function being modeled with a sum of trees, it is proposed to model the variance function with a product of trees thus improving on the original constant variance error model.

Further parallels between HME and BART models can be drawn when looking at the model trees BART (MOTR-BART) extension (Prado et al., 2021). Instead of having a unique prediction value at each of the terminal nodes, MOTR-BART uses a linear predictor. Unlike HME, this approach does not use all of the input variables to obtain a prediction. In fact, the prediction is made based on the covariates that are present in the splitting decisions of the corresponding tree. It is shown that MOTR-BART requires fewer trees in order to equal or outperform BART.

8.4 HME Evaluation Against Competitors on Motorcycle Accident Data

In this section, GAM and BART models are fitted to the standardised motorcycle accident data set and compared to the HME fit produced by the RJ MCMC discussed in Chapter 6. As seen previously, the accelerometer readings act as a response variable $\mathbf{y} = (y_1, ..., y_n)^T$ while time is treated as the single explanatory variable $\mathbf{x} = (x_1, ..., x_n)^T$. GAM is fitted in `R` using the default settings of the function `gam` from the package `gam`. Since there is only one explanatory variable present in the model, the GAM model is

equivalent to a simple one-spline model. The default smoothing splines are used to represent the relationship between time and acceleration. BART is fitted to the same data set in `R` using the default settings of the function `bart` from the package `BayesTree`. In this section, the three models are compared in terms of treatment of heteroscedasticity, accuracy of the fitted values, and interpretability.

8.4.1 Heteroscedasticity Assessment

In this section, the prediction intervals are investigated in order to assess the effects of heteroscedasticity on the model fits produced by HME, GAM, and BART (please refer to Appendix E for how these are obtained).

Figure 8.1 showcases the predictions produced by HME model as seen in Chapter 6. It is evident that the prediction intervals account for the change in variance, i.e., are broader for areas with more uncertainty and narrower for the areas with tighter clustered observations. In contrast, a spline-based model and BART fail to account for the heteroscedasticity resulting in too wide prediction intervals for the areas of lower variability (Figures 8.2 and 8.3, respectively).

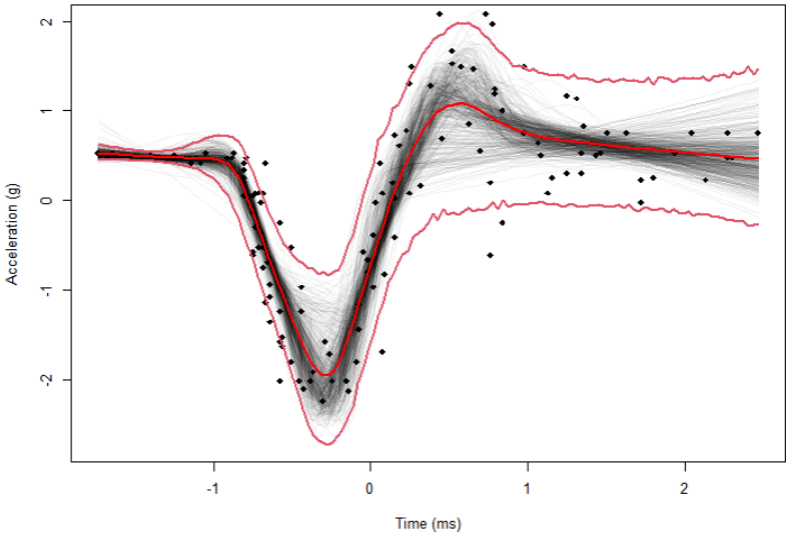


FIGURE 8.1: HME prediction intervals for the motorcycle accident data. The thin red lines show the 2.5-th and 97.5-th percentiles of predictions made during the RJ MCMC iterations. The thick red line shows the average predictions. Thin lines correspond to every 10-th prediction.

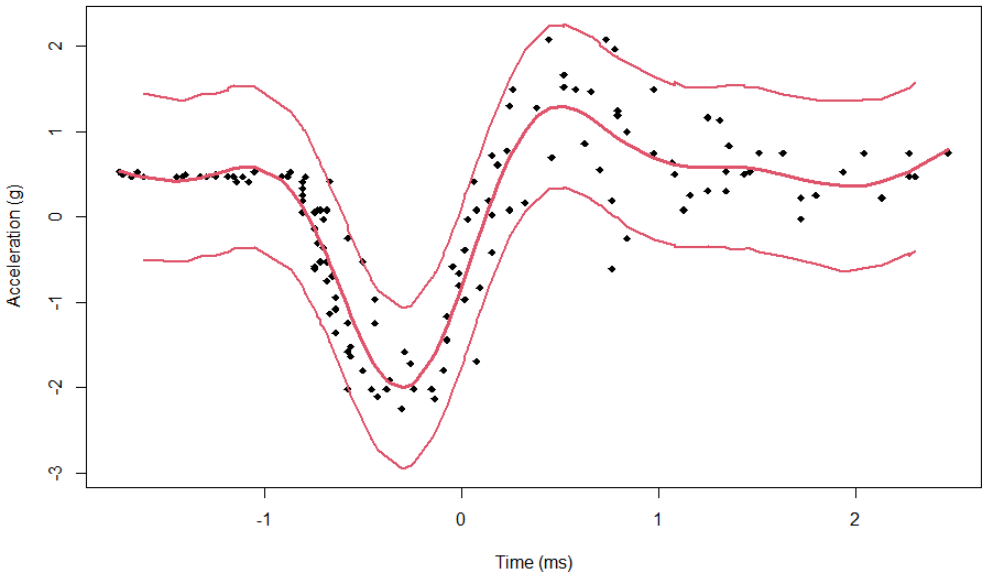


FIGURE 8.2: Spline-based model fitted using `R` package `gam` for motorcycle accident data set. The thick red line corresponds to the fitted values. The thin red lines correspond to the 2.5-th and 97.5-th percentiles of predictions obtained as per Andersen (2019).

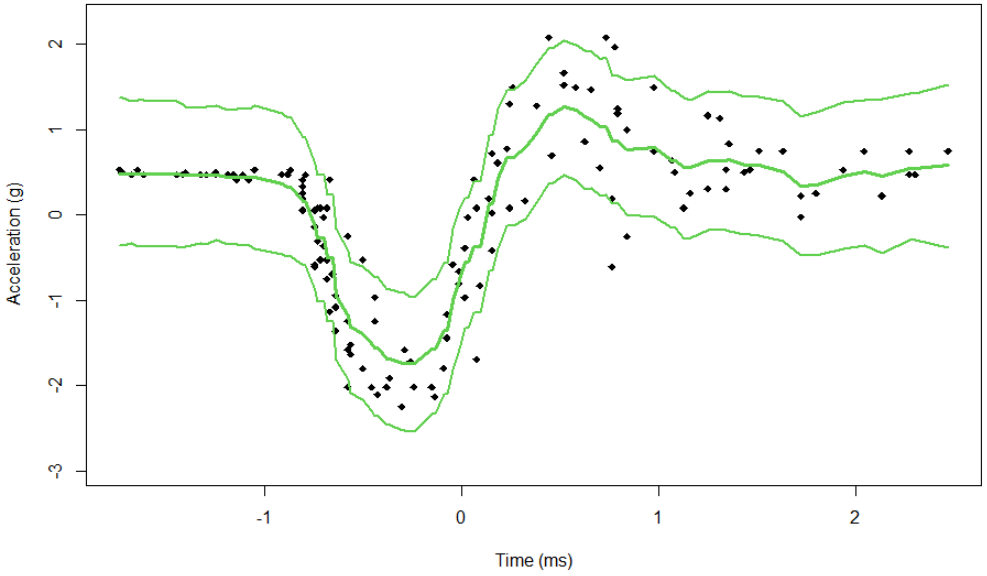


FIGURE 8.3: BART prediction intervals for the motorcycle accident data. The thin red lines show the 2.5-th and 97.5-th percentiles of predictions made during the MCMC iterations. The thick red line shows the average predictions.

It is also interesting to note that the average predictions made by the HME model appear smoother than those of BART. Overall, it appears that BART predictions track the data too closely, especially so in the area of the highest variability in the response. If the motorcycle data set contained more observations, this could be improved upon by splitting the data into training and test sets. Failing to do so due to a small number of observations might result in an overestimated predictive performance, which is investigated next.

8.4.2 Predictive Performance

The accuracy of the fitted values and average predictions for the three models are summarised in Table 8.1. It can be seen that HME model outperforms GAM and comes close to BART (difference in the MSE of 0.025).

TABLE 8.1: Mean squared error obtained from the predictions made on the motorcycle accident data for the following models - Hierarchical Mixture of Experts (HME), Generalised Additive Model (GAM) - one spline model in this application, and Bayesian Additive Regression Tree (BART).

	HME	**GAM**	**BART**
Mean Squared Error	0.2081	0.4512	0.1832

Although the motorcycle data set served as an excellent example to illustrate how the HME model accounts for heteroscedasticity, the data set is rather simple and small, which limits the assessment of predictive performance. The next chapter discusses a more complex case, where predictions are made on the test set, and hence revisits the assessment of the predictive performance of the three models.

8.4.3 Interpretability

The motorcycle accident data set consists of one explanatory variable and a response variable, which makes the visualisation of the model fit rather straightforward. It has been shown in Chapter 6, Section 6.8.5 that the HME model fit produces a large amount of interpretable output such as predictions at each iteration, average predictions, the latent assignment variables, model fit at a chosen iteration (including the normal expert density parameters) as well as an indication of split abruptness between the experts (path probabilities and responsibilities). All of the above leads to an in-depth understanding of the HME model fit and the experts within it.

GAM also allows one to look at the fitted values (shown as a thick red line in Figure 8.2) as well as visualise the smooth function used to represent the relationship between

explanatory and response variables (Appendix F). Since GAM is a frequentist approach, the model output also produces a measure of smooth term significance as well as the standard model fit assessment and assumption validation techniques.

BART creates draws from the posterior distribution of the response variable at each iteration, which are then averaged to obtain the predictions shown as a thick green line in Figure 8.3. Similarly, the posterior variance parameters can be obtained. BART is an additive model and thus one cannot investigate the individual terms/trees using the `BayesTree` BART implementation in `R`. The latter means that it is not possible to visualise where and in what order the tree splits occur or what numeric outputs are produced at the leaves of individual trees.

Even though it is evident that HME allows for deeper interpretability of the fitted model, in a two-dimensional problem, one could argue that the level of interpretability offered by GAM and BART is sufficient for the application. In the next chapter, a multidimensional real-life problem, for which a high level of interpretability is required, is considered.

Chapter 9

Rental Prices in Glasgow

9.1 Introduction to Glasgow Rental Market

With 38% of Scotland's population living in rented accommodation, the opportunity for buy-to-let investments is at an all-time high (Scotland's Census, 2011). According to the supply and demand data collected by Admiral (2022), the second-highest tenant demand in the UK is faced by the Scottish city of Glasgow, where there are 998 people looking to rent for every 100 available rental properties. The latter demand coupled with Glasgow being the most affordable city in Scotland to buy property makes it a perfect candidate for a buy-to-let investment (RW Invest, 2022). A viable first-time investor strategy, requiring minimal capital, relies on purchasing a studio or a one-bed flat. Home.co.uk (2022) estimate the average price of £105, 961 for a one-bed flat in Glasgow making it notably more affordable when compared to the average price of £161, 252 for a two-bedroom flat in Glasgow. Moreover, in the first quarter of 2022, one-bedroom flats were the quickest to let across Glasgow with an average time to let of 10 days hence minimising the cost of property staying vacant (ESPC, 2022a). The average rental price for a one-bedroom property in Glasgow is estimated to be £936 per calendar month, or pm, which generates an attractive annual pre-tax yield of 10.6% on average (ESPC, 2022b). Thus this chapter focuses on rental prices of one-bedroom and studio apartments in the city of Glasgow.

Geographical location is the main factor to be considered when investigating the rental prices of one-bedroom and studio flats in Glasgow. To illustrate, consider the rental listings available for one-bed and studio flats on a property portal website Zoopla on the 13th of September, 2022. The lowest listed rental price of £450 pm is observed in the area of Baillieston (marked in red in Figure 9.1) while the highest rental price of £2, 882 pm is listed for a property on Ingram street, the city center of Glasgow (marked in blue in

Figure 9.1). The latter example illustrates the extremes of rental prices in two areas that are 7 miles apart. However, one does not need to stretch far in order to observe a sharp difference in the rental prices in Glasgow. For instance, consider two areas of Glasgow that are only one subway stop, or half a mile, apart - Partick and Govan (marked in purple and in green, respectively, in Figure 9.1). On the 13th of September, 2022, the average listed rental prices for a one-bedroom flat on Zoopla were £956.67 pm and £650 pm for Partick and Govan, respectively. Given the abrupt rental price changes across many proximities, estimating the rental price given the precise geographic location of the property is of direct interest to buy-to-let investors.

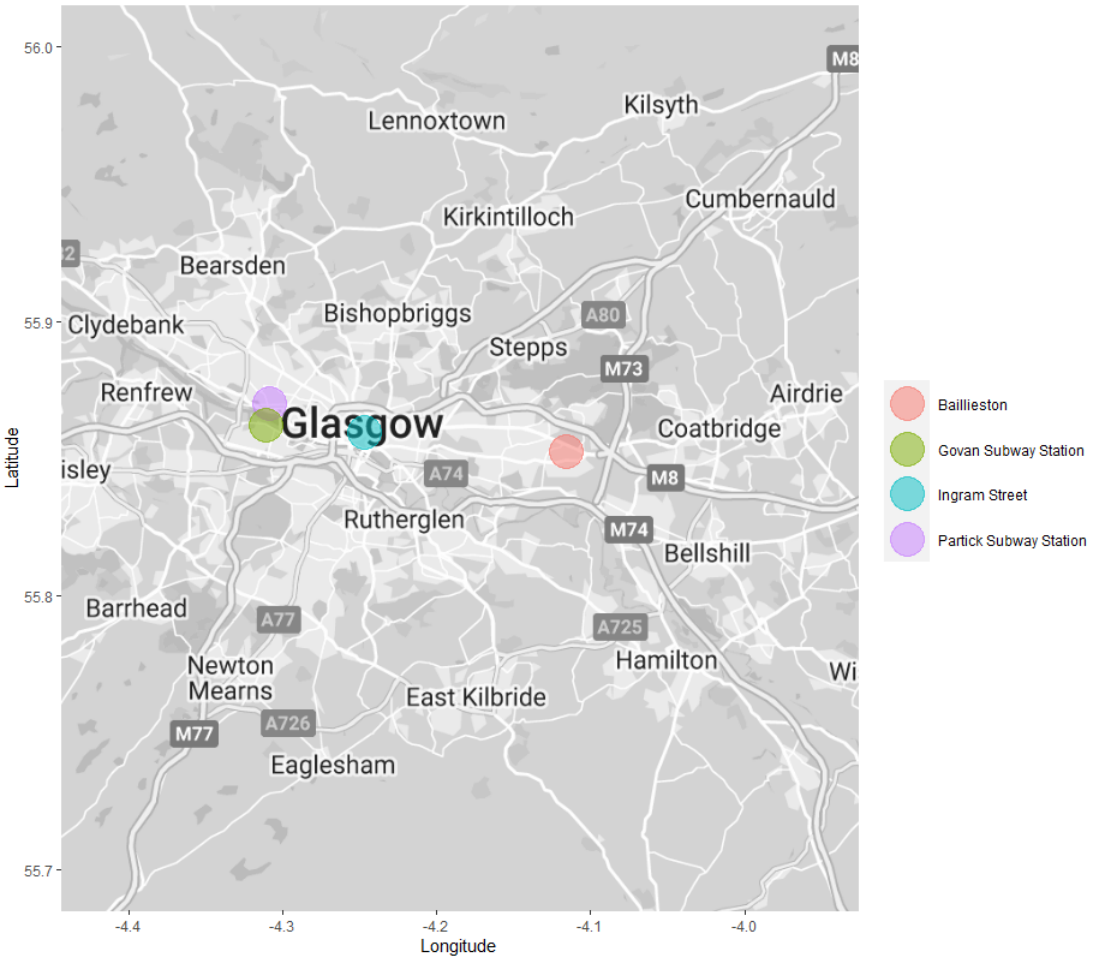


FIGURE 9.1: Map of Glasgow, Scotland. The circles represent points of interest centered at the exact location of subway stations and the central coordinates for the remaining locations.

In this chapter, a hierarchical mixture of experts model with normal experts is fitted to the property rental price data provided by Zoopla Limited (2022). The properties

analysed include those classified as belonging to the Glasgow area by Zoopla, which includes Glasgow city as well as parts of some neighboring local authorities. A subset of the data containing rental prices and geographical locations of studio flats, one-bedroom apartments, and one-bedroom maisonette flats is considered. The data investigated spans three years of the most recent records available, i.e., the period from the 1st of July, 2019 to the 30th of June, 2021. Firstly, some exploratory analysis of the data is performed in Section 9.2. Next, the model fitting process is outlined in Section 9.3. The HME model is first fitted using the RJ MCMC with results presented in Section 9.4. The gate swap extension to the algorithm is added and assessed in Section 9.5. The HME model is then compared to two of its competitors in Section 9.6. Finally, the results and findings are summarised in Section 9.7.

9.2 Exploratory Analysis of Glasgow Rental Prices

This section undertakes the exploratory analysis of the rental prices in Glasgow for all 880 of the studios, one-bed flats, and one-bed maisonettes listed for rent in the period between the 1st of July, 2019 and the 30th of June, 2021. The rental price density histogram and smooth density function, estimated with kernel smoothing using `density` function in `R`, is shown in Figure 9.2. It can be seen that recorded prices range from as low as £240 pm to as high as £1,016 pm.

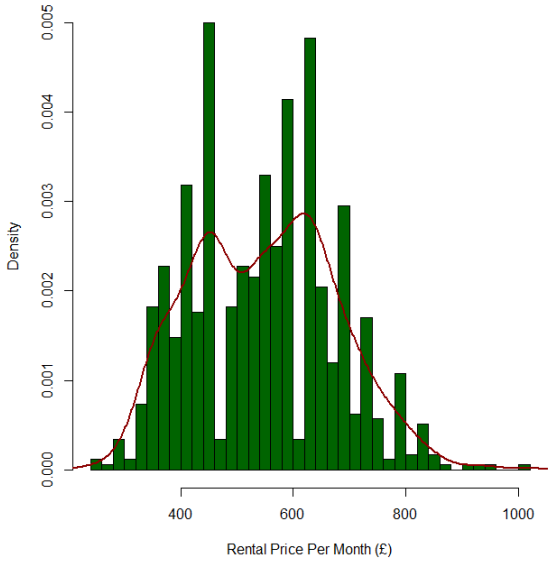


FIGURE 9.2: Histogram of the density of monthly rental prices in Glasgow, Scotland. The dark red line corresponds to the smooth kernel density function of the rental prices.

There are two peaks observed in the density function occurring at around £450 pm and £620 pm. Recall that HME models with normal experts partition the problem into several subproblems, where a simple linear model is sufficient. In fact, the latter property makes the HME models very well-equipped to address the potential multimodality.

Figure 9.3 facilitates further investigation of the rental prices in the context of their geographical location. The plot depicts all 880 properties available in the data set where data points are coloured by the listed rental price per month.

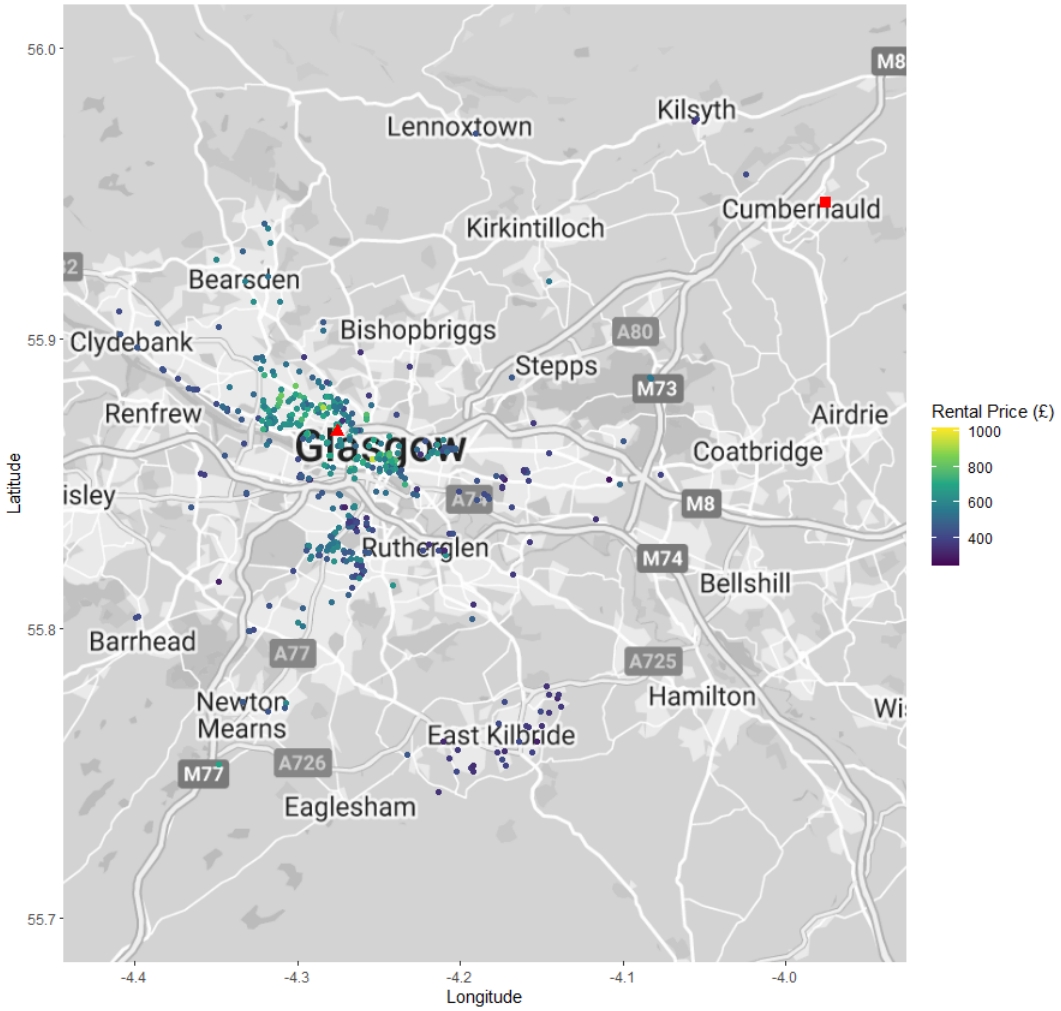


FIGURE 9.3: Map of Glasgow, Scotland, with points representing properties listed for rent. The property points are colored by the listed rental price per month. The red triangle and square points correspond to the most expensive and the cheapest rentals, respectively.

The lowest monthly rental price recorded for the period (£240 pm) belongs to a property in Cumbernauld and the highest rental price (£1,016 pm) to a property in the desirable Park Circus location in the west end of Glasgow (marked as red square and triangle, respectively, in Figure 9.3). Overall, higher rental prices appear to be concentrated in

the city center as well as the west end of Glasgow. The south side of the river appears to have lower rental prices when compared to the north side with the exception of the east end of Glasgow, where rental prices appear to be relatively low on both sides of the river. Figure 9.4 divides the area of Glasgow into nine subsets with the boundaries defined by the subway stations of the city.

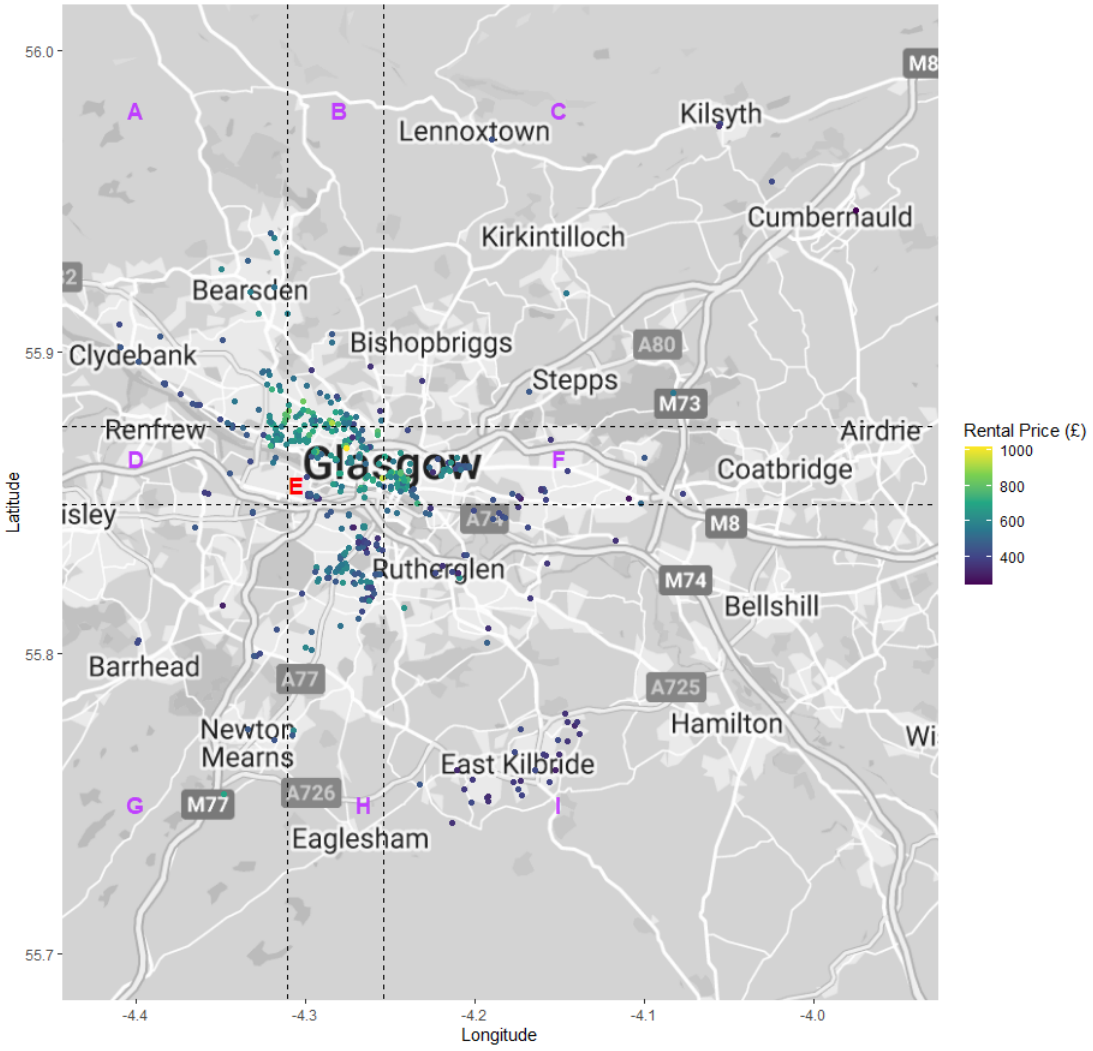


FIGURE 9.4: Map of Glasgow, Scotland, with points representing properties listed for rent. The property points are colored by the listed rental price per month. The dashed lines correspond to the coordinates of the most southern (West Street), eastern (Buchanan Street), northern (Hillhead), and western (Govan) subway stations. These boundaries divide the space into nine partitions marked by letters from A to I.

Area E corresponds to the *inside* of the subway circle. The area features some of the highest rental prices recorded with an exception of the bottom left corner, corresponding to the area of Govan, where rental prices seem to be lower overall. Area A exhibits a divide between the regions of Clydebank with lower prices and Bearsden/Milngavie where most rental prices seem to be middle-range. A similar pattern is seen in area B,

where areas such as Kelvindale and Kelvinside exhibit higher prices than those further north. Area C depicts fewer properties for rent overall with prices closer to the lower end of the scale. Area D reveals higher rental prices for the north of the river Clyde in areas such as Partick when compared to the south side of the river, which has similar rental prices as those seen in area G. Area H exhibits a mixture of rental prices with lower ones present towards the east of the boundary, which reflects the separation between the areas of Pollokshields and Shawlands and the area of Govanhill. Area F shows a clear decrease in the rental price with the increase in the easting of the property. Finally, area I depicts more scattered rental properties with a cluster forming in East Kilbride, where rental prices appear to be consistent with those seen in the east of Glasgow in areas C and F. To summarise:

- The north side of the river inside the subway circle and the northwest of Glasgow are subject to higher rental prices.
- It appears that the location of the property with respect to the river is more important in the west than in the east of the city.
- A pocket of higher rental prices is present in the south of Glasgow in the area of Pollokshields/Shawlands amongst the overall lower rental prices.

A divide and conquer algorithm such as a hierarchical mixture of experts is a good fit for such complicated relationships for several reasons. Firstly, conflicting relationships between the rental prices and geographical locations can be represented by the different models fitted to partitioned subspaces of the problem. Soft probabilistic boundaries allow for both sharp and smooth transitions between the models fitted reflecting the reality of the rental market in Glasgow. The automatic model architecture methodology developed in Chapter 6 decreases issues arising from preselecting the space partitions. In addition to the ability to predict rental prices, HME model also offers a level of interpretability, which is an essential feature for buy-to-let investors. The subsequent sections of this chapter discuss fitting HME model with normal experts to the rental price data as well as assessing the performance of the proposed methodology, predictive performance, and the interpretability of the model.

9.3 HME Model Fitting for Glasgow Rental Prices

A hierarchical mixture of experts model with Gaussian experts is fitted to the standardised Glasgow rental price data. The data set containing 880 observations is randomly

divided into training (70%) and test (30%) subsets. All models discussed in this chapter are fitted on the training data set while their predictive performance is evaluated on the test set. The monthly rental price acts as the response variable, where each y_i corresponds to the monthly rental price of the i-th property. The longitude (x_{1i}) and latitude (x_{2i}) of the property are treated as the explanatory variables forming a vector $\mathbf{x}_i = (x_{1i}, x_{2i})$, which corresponds to the geographical location of the i-th property.

The initial state consists of one expert, which is then improved upon by the automatic tree growth using the reversible jump and gate swap methodology developed in Chapter 7. First, an HME model is fitted using RJ MCMC only, which is then further extended to RJ GS MCMC. Following the initial impression formed by exploratory analysis, a wide, weakly informative gating parameter prior is chosen to reflect the varying level of abruptness in rental prices across many proximities. Similarly, given the anticipated conflicting patterns in rental prices across their geographical locations, a wide weakly informative prior is chosen for the intercept and slopes expert parameters. In order to encourage a tighter allocation to experts across all geographic locations, a moderate prior is selected for the variance. Thus, all variables are standardised with the following prior parameter values chosen:

$$\begin{aligned} \boldsymbol{\beta}_E &\sim \mathsf{NIG}\left(\mathbf{0}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, 1, 0.01\right) \\ \boldsymbol{\gamma}^{(G,H)} &\sim \mathsf{MVN}\left(\mathbf{0}, \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}\right) \end{aligned} \tag{9.1}$$

for all $(G, H) \in P_E$ and for all $E \in \mathcal{E}$.

It has been shown in Section 6.8.6 that the frequency and quantity of reversible jumps do appear to have an effect on the overall preferred fit of the model. As seen previously, in order to propose a large total number of jumps that are given a chance to be improved upon once accepted, a single reversible jump is proposed every 10 MCMC iterations. As seen before, the forward jump parameters are drawn from two Gaussian proposal distributions ,i.e., $\boldsymbol{\gamma}_1 \sim \mathsf{N}(0, 100)$, $\epsilon \sim \mathsf{N}(0, 0.05)$ (see Section 6.5.2 for the forward jump proposal methodology). A wide proposal distribution is selected for the slopes of logistic regression to encourage the exploration of both smooth and abrupt transitions during forward jumps. As usual, a small amount of variation is then added to the slope parameter of the logistic regression in order to add an element of randomness to the proposal. All MCMC chains discussed in this chapter are run for $1,100$ iterations with

the first 100 discarded for burn-in. The convergence of the chains is assessed by visual inspection of the predictions as outlined in Section 4.3.4.2.

The HME model is first fitted using RJ MCMC and the results are outlined in the following section.

9.4 RJ MCMC Results for Glasgow Rental Prices

Table 9.1 summarises the recorded reversible jump acceptance rates while Figure 9.5 depicts the distribution of the number of experts in the model trees. The overall recorded acceptance rate of 14.55% for the reversible jumps is a notable improvement on the notoriously low reversible jump acceptance rates (Al-Awadhi et al., 2004; Ehlers and P. Brooks, 2008; Farr et al., 2015; Brooks et al., 2003). It is evident that the preferred number of experts in the tree is 3 with reversible jumps exploring up to 4 experts. Figure 9.6 further reveals that automatic tree growth took place during the first 50 jumps after which the fit settled on a model with 3 experts.

TABLE 9.1: Acceptance rates of the reversible jumps for the Glasgow rental prices data.

	Splits	**Merges**	**All Jumps**
Number Proposed	60	50	110
Number Accepted	9	7	16
Acceptance	15.00%	14.00%	14.55%

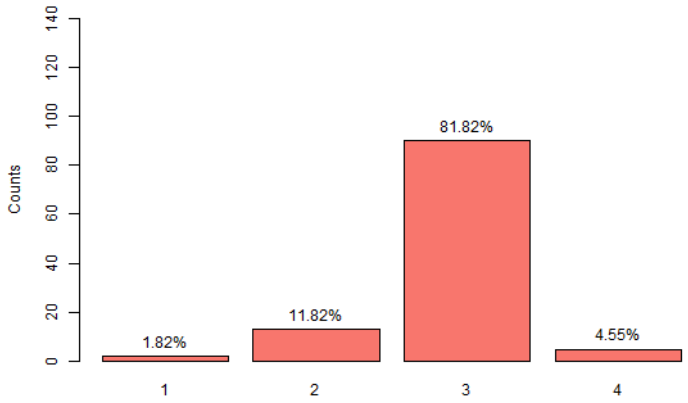


FIGURE 9.5: Distribution of the number of experts in the RJ MCMC chain for the Glasgow rental prices data.

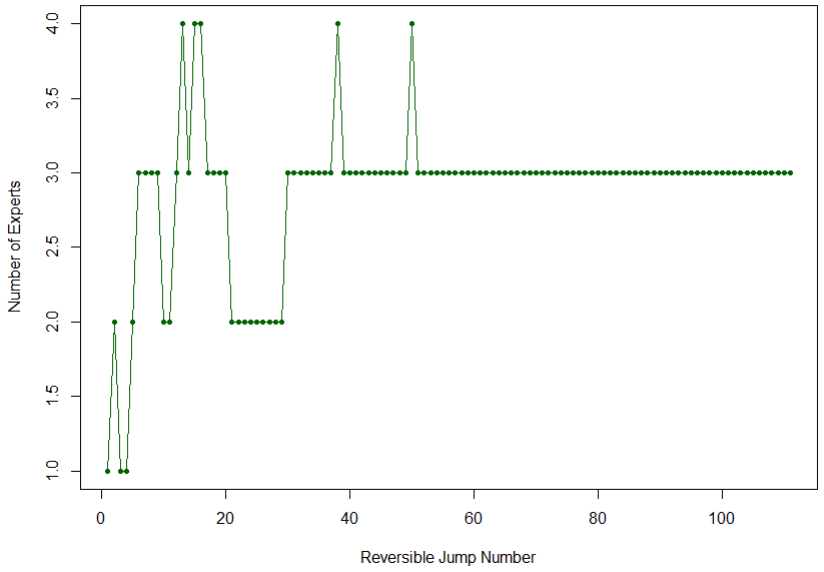


FIGURE 9.6: Number of experts in the tree after each reversible jump proposal step in RJ MCMC for the Glasgow rental prices data.

An additional RJ MCMC run has been created in order to obtain the Gelman-Rubin statistic resulting in the value of 1.06, which strongly suggests that the convergence has been achieved. Given that there were no jumps after the 50-th one, the model fit can be further investigated by focusing on the MCMC iterations after the jumps stopped. Even though model parameters are still updating and changing at each iteration, their posterior point estimates can be investigated and obtained as means of posterior parameter values.

For the allocation variables, one can interrogate the information on which expert each of the properties has been allocated to the majority of the time post the 50th jump. Figure 9.7 offers the previously seen map view of Glasgow with the resulting allocations. It is evident that Expert 1 (pink) contains properties located in the east and southeast of Glasgow bordering with Expert 2 (green), which groups properties located in the city center and northwest of Glasgow while Expert 3 (blue) properties are located in the south, southwest and far west of the city. It is also apparent that the three experts *meet* in the center of Glasgow. It is interesting to note that the river closely follows the border between Experts 2 and 3, however, doesn't seem to be a factor for Expert 1. The separation between Experts 1 and 2 occurs in the previously noted areas of Pollokshields and Govanhill. Similarly, Experts 1 and 3 draw a border between the central Merchant city district and neighboring Carntyne, Bridgeton, and Parkhead areas. All of the above is consistent with the observations made in the exploratory stage of the analysis.

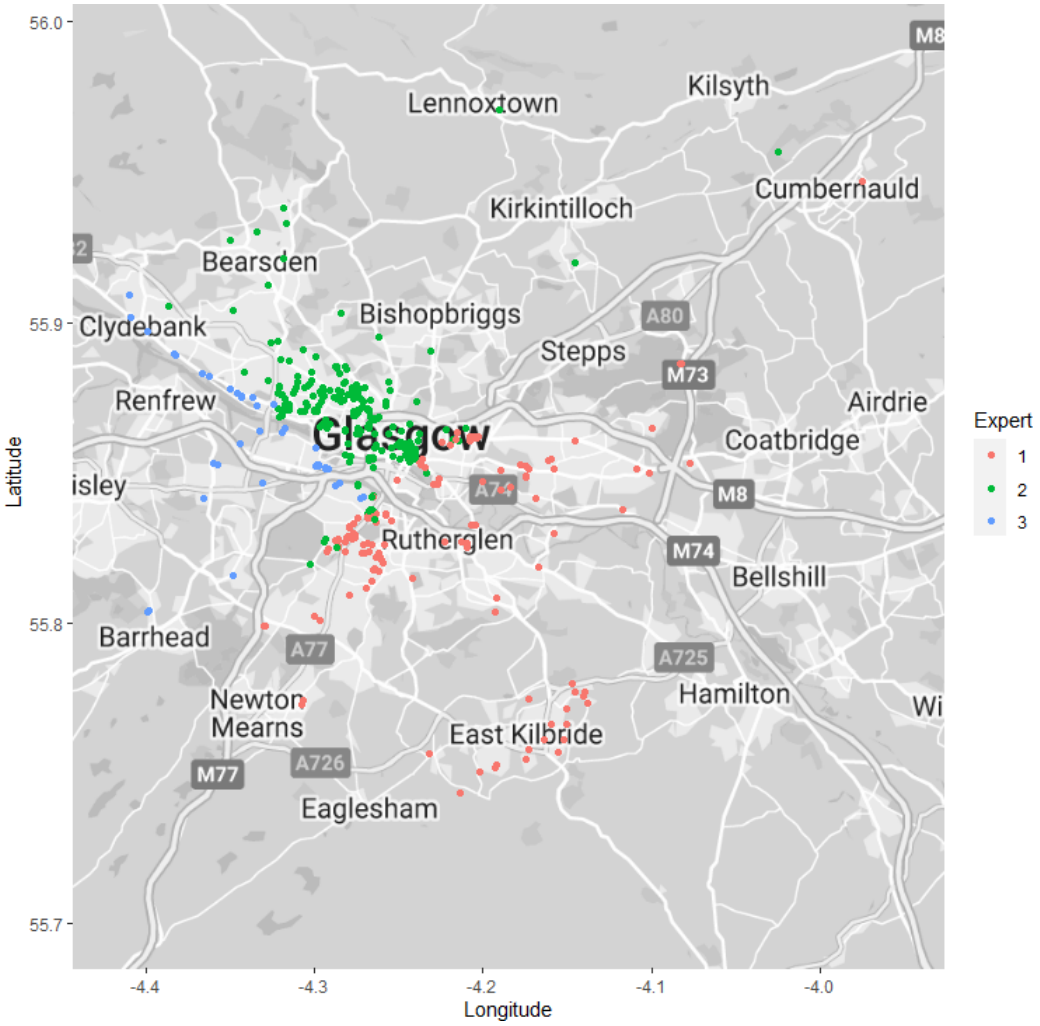


FIGURE 9.7: Assigned allocations to the 3 experts in the HME tree for the Glasgow rental prices training data. The depicted allocations correspond to which expert each of the properties has been allocated the majority of the time.

The three-dimensional view depicted in Figure 9.8 helps to further visualise the average allocations in space after the 50th reversible jump (one can view an animated version here). It is clear that the properties assigned to Expert 2 (green) tend to have higher rental prices on average, followed by Expert 1 (pink) and Expert 3 (blue), which is also consistent with the observations made in the exploratory stage of the analysis. It is interesting to note that Expert 1 properties that are located in between Expert 1 and Expert 2 have lower rental prices than the other properties in the group. In general, the rental prices appear to increase towards the center point for pink and blue points and increase even further as the latitude increases for the green points.

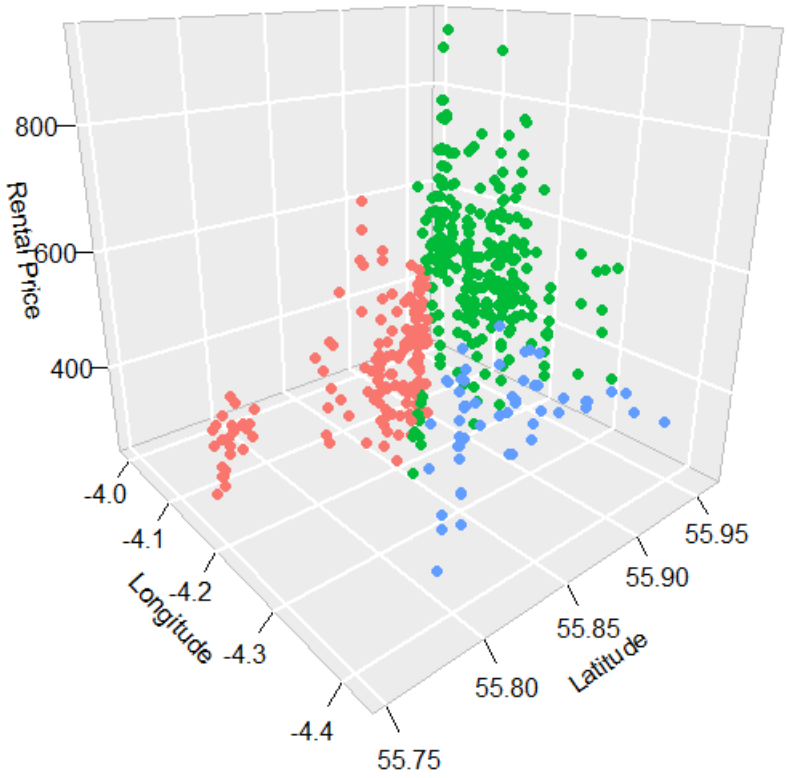


FIGURE 9.8: Three-dimensional view of the geographic locations and recorded rental prices for the Glasgow rental prices training data. The depicted allocations correspond to which expert each of the properties has been allocated the majority of the time. Expert 1 - pink, Expert 2 - green, Expert 3 - blue. Animated version available here.

It is crucial to remember that HME models provide soft probabilistic splits, which means that the division between experts is not as clear-cut as it appears when solely looking at the allocation variables. This means that some points might actually have similar probabilities of being assigned to two or even all three of the experts in the tree. Thus, let us refer to Figure 9.9 and Figure 9.10 to assess the abruptness of the splits between experts. The figures represent the average path probabilities and responsibilities (path probabilities multiplied by the expert densities), respectively, associated with the 3 experts in the tree after the 50-th jump as pie chart slices. It is shortly illustrated why it is of interest to investigate both path probabilities and responsibilities. To further investigate the behavior of rental prices within each expert, the radius of the pie charts in the figures illustrates the rental price of the property, i.e., the larger the radius, the higher the monthly rental price. Figures 9.11 and 9.12 provide a closer view of the soft splits in the areas of borders between the experts.

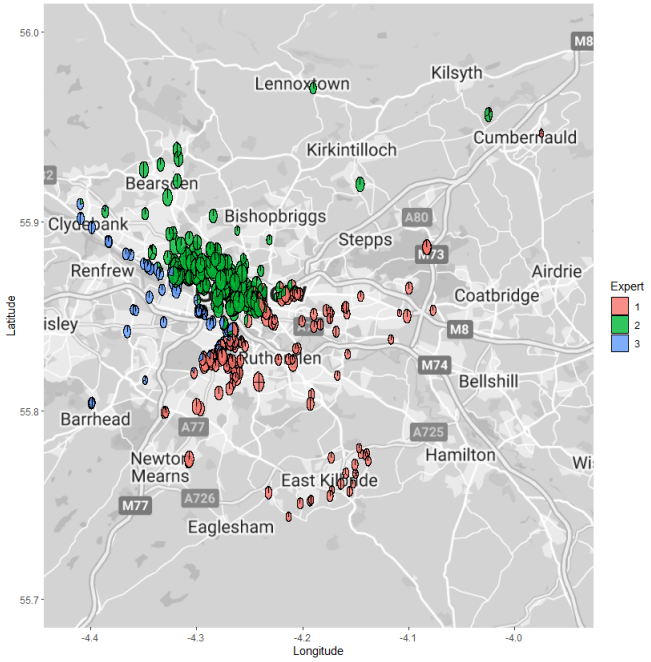


FIGURE 9.9: Pie charts colored by the average path probabilities associated with each of the 3 experts in the tree for the Glasgow rental prices training data. Individual pie chart radiuses illustrate the rental price of the property - the larger the radius, the higher the monthly rental price.

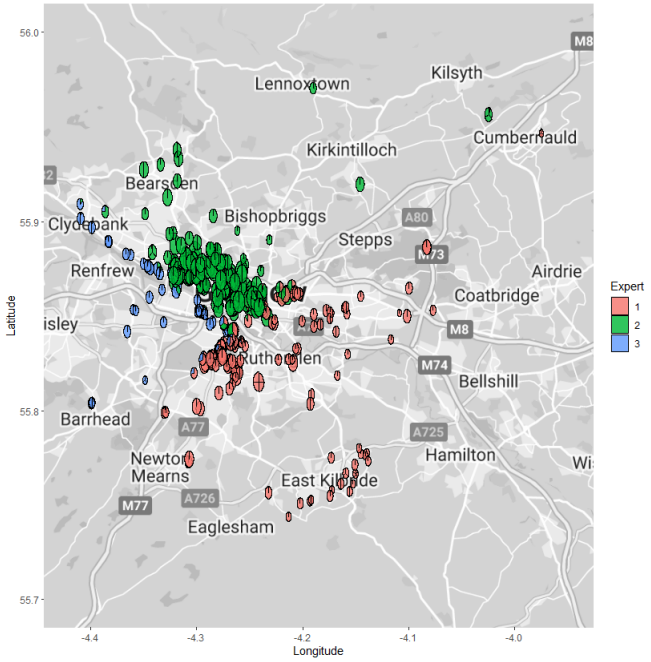


FIGURE 9.10: Pie charts colored by the average responsibilities associated with each of the 3 experts in the tree for the Glasgow rental prices training data. Individual pie chart radiuses illustrate the rental price of the property - the larger the radius, the higher the monthly rental price.

Recall that path probabilities, $\pi_i^{(E)}$, are obtained based on values of explanatory variables, in this case, longitude and latitude, and values of gating parameters corresponding to the nodes of the tree. On the other hand, responsibilities also take into account the expert densities, which depend on the parameters of experts present in the tree, i.e., $\upsilon_i^{(E)} = \frac{\pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right)}{\sum_{E\in\mathcal{E}} \pi_i^{(E)} f^{(E)}\left(y_i|\mathbf{x}_i,\boldsymbol{\theta}^{(E)}\right)}$. Thus, comparing the two can help one better understand the probabilistic splits.

At a first glance, the experts appear to be well separated. Both path probabilities and responsibilities indicate a smoother transition for some properties located on the borders between the experts. On the other hand, the assignment to the latent groups is rather strict when moving away from the boundaries between experts. It is also evident that Expert 2 contains some of the properties with the highest rental prices since the green points appear to be the largest, on average. Expert 3 points are the smallest and most consistent in size indicating the lowest rental prices on average as well as the least amount of variation. There also seems to be more variation present in the rental prices of the properties in Expert 1 when compared to the other two experts in the tree. In general, for Expert 2, points get slightly smaller towards the north and far west of the city. For Expert 1, the size of the points decreases towards the east of the city with the highest rental prices situated in the center of the south of the city.

Next, let us refer to the zoomed-in versions of Figure 9.9 and Figure 9.10. Consider the path probabilities recorded for a property circled in red in Figure 9.11 and denote it as the m-th property. It is evident that path probabilities are distributed across all three experts with $\pi_m^{(E1)} = 0.327$, $\pi_m^{(E2)} = 0.079$, and $\pi_m^{(E3)} = 0.594$. This means that based on its geographical location, this property is most likely to be assigned to expert $E3$ (blue), followed by $E1$ (red) and $E2$ (green). Information, provided by path probabilities, is next compared to that arising from responsibilities. The same property is now circled in red in Figure 9.12. It is immediately clear that the slice corresponding to $E2$ seems to have disappeared from the pie chart. In fact, the recorded responsibilities are $\upsilon_m^{(E1)} = 0.108$, $\upsilon_m^{(E2)} = 0.001$, $\upsilon_m^{(E3)} = 0.891$. It is thus evident that for this property the separation between experts becomes more confident once the expert densities are taken into account. That is, it is now very unlikely for this property to be allocated to $E2$ (green) with the most likely outcome of it being allocated to $E3$ (blue). A similar pattern can be observed for several properties neighboring the one discussed in here. For this example, it is now clear that the more abrupt separation between $E2$ (green) and the other two experts in this area is stemming from expert densities.

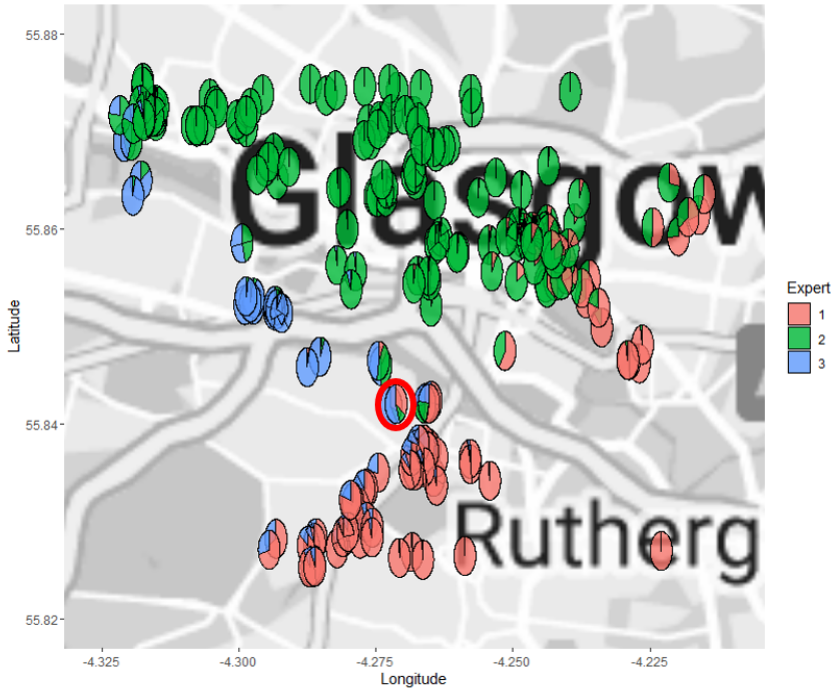


FIGURE 9.11: Zoomed-in version of pie charts colored by the average path probabilities associated with each of the 3 experts in the tree for the Glasgow rental prices training data.

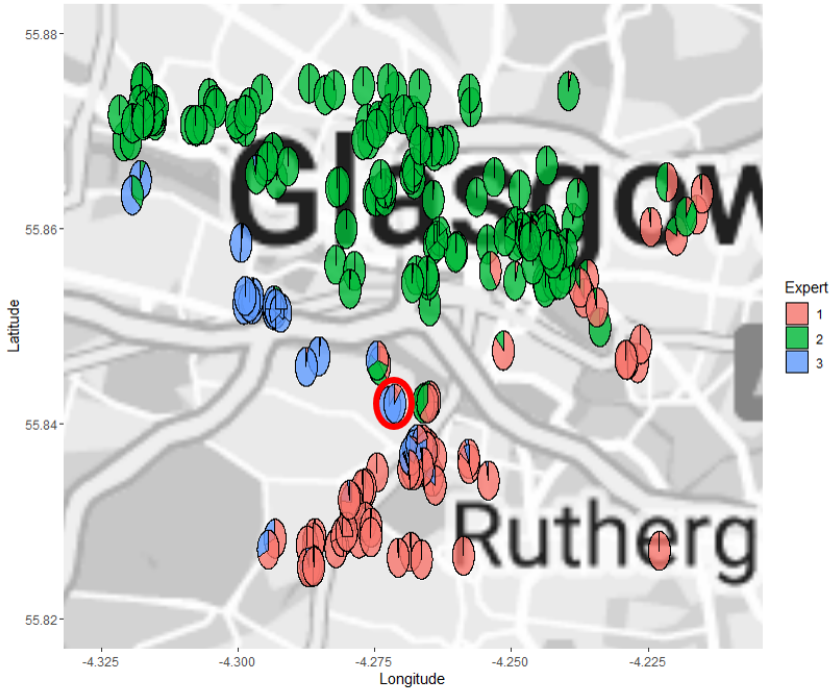


FIGURE 9.12: Zoomed-in version of the pie charts colored by the average responsibilities associated with each of the 3 experts in the tree for the Glasgow rental prices training data.

The impression formed so far is further confirmed by the results presented in Table 9.2. It is evident that the highest average rental price has indeed been recorded for properties allocated to Expert 2, followed by Expert 1 and Expert 3. It is interesting to note that the mean rental prices for Expert 1 and 2 are close to the two modes of the response noted in the explanatory stage of the analysis (see Figure 9.2).

All parameter estimates discussed in relation to Table 9.2 are posterior point estimates obtained as the mean of posterior parameter values and are simply referred to as estimates in the following paragraphs.

TABLE 9.2: For each expert: the average rental price of properties allocated to the particular expert for the Glasgow rental prices training data; the estimated mean posterior parameters for the densities of the normal experts, where $\hat{\beta}_{lonE}$ and $\hat{\beta}_{latE}$ correspond to the coefficients for the longitude and the latitude, respectively, and $\hat{\sigma}^2_E$ corresponds to the mean normal expert variance parameter.

Expert (E)	**1**	**2**	**3**
Mean Rental Price (£)	457.91	606.52	416.26
$\hat{\beta}_{lonE}$	-0.3505	0.0820	-0.0288
$\hat{\beta}_{latE}$	0.1197	-0.4422	0.1869
$\hat{\sigma}^2_E$	0.3537	0.5558	0.1431

It has been noted that there might be different levels of variability, or heteroscedasticity, present in the response across the experts. In agreement with the observations made previously, the highest estimated variability in the response corresponds to Expert 2 while the lowest corresponds to Expert 3.

Looking at the estimated slope coefficients for Expert 1, the negative value for the longitude slope parameter indicates that, keeping the latitude constant, the rental prices of those properties assigned to Expert 1 decrease towards the east of the city. Similarly, the positive value for the estimated latitude slope parameter indicates that, keeping longitude constant, the rental price of properties increases towards the north of the city.

For Expert 2, the value of the longitude slope parameter estimate is positive, however close to zero, indicating that, keeping the latitude constant, the rental prices of those properties assigned to Expert 2 increase slightly towards the east of the city. In contrast to Expert 1, for Expert 2 the negative value for the latitude slope parameter estimate indicates that keeping longitude constant, the rental price of properties decreases towards the north of the city, which is consistent with previously made observations.

Finally, for Expert 3 the value of the longitude slope parameter estimate is negative, however, close to zero, which means that keeping the latitude constant, the rental prices of those properties assigned to Expert 3 decrease slightly towards the east of the city.

Similarly to Expert 1, for Expert 3 the positive value for the latitude slope parameter estimate indicates that keeping longitude constant, the rental price of properties increases towards the north of the city.

Overall, it is evident that rental prices for Expert 1 and 3 behave similarly with respect to longitude and latitude, however, the two experts are the furthest away from each other geographically. Expert 2, on the other hand, exhibits the opposite effects when it comes to the relationship between the rental price and longitude/latitude.

The posterior point estimates of gating slope parameters, obtained as means of posterior parameter values, for two gates in the tree are given in Table 9.3. Let $G1$ denote the root gate node and let $G2$ be a child gate of $G1$.

TABLE 9.3: For each gate: the estimated posterior means for gating parameters, where $\hat{\gamma}_{lonE}$ and $\hat{\gamma}_{latE}$ correspond to the coefficients for the longitude and the latitude, respectively for the Glasgow rental prices training data.

Gate (G)	**1**	**2**
$\hat{\gamma}_{lonG}$	8.5667	-17.9809
$\hat{\gamma}_{latG}$	-11.124984	-20.85077

It is evident that both gates estimate a more abrupt separation between experts with respect to latitude than longitude. It can also be seen that the second split at $G2$ is estimated to be more abrupt than the first one at $G1$. Unfortunately, commenting on gating parameter slope estimates has limited value when it comes to interpretability. An alternative way of investigating the splits involves evaluating the logistic regression function for each gate on a grid, which corresponds to the ranges of explanatory variables, and scaling the result to the range of response variable in order to visualise split planes such as shown in Figure 9.13.

At a first glance, it is clear that, as expected, both gates provide an abrupt separation between experts. It can be seen that two gates split the problem space in different locations. Comparing the location of the first split (at the root gate $G1$) shown in plot (ii) with the corresponding allocations depicted in plot (i), it is evident that the first split separates $E1$ (pink) from the other two experts. Similarly, looking at the second split (at $G2$) shown in plots (iii) and (vi) and comparing them against the allocations as per (i) and (iv), it is evident that the second split defines the boundary between $E2$ (green) and $E3$ (blue). Having assessed the locations of the splits in the context of allocations, the previously noted higher absolute values for $G2$ slope parameters appear consistent with the prior knowledge of rental prices. That is, there is a larger difference between rental prices recorded for properties in the boundaries between experts $E2$ (green) and $E3$ (blue) than that between $E1$ (pink) and the other two experts. Such thorough

investigation of the average HME model fit allows for reconstructing the post reversible jump model architecture, which is shown in Figure 9.14.

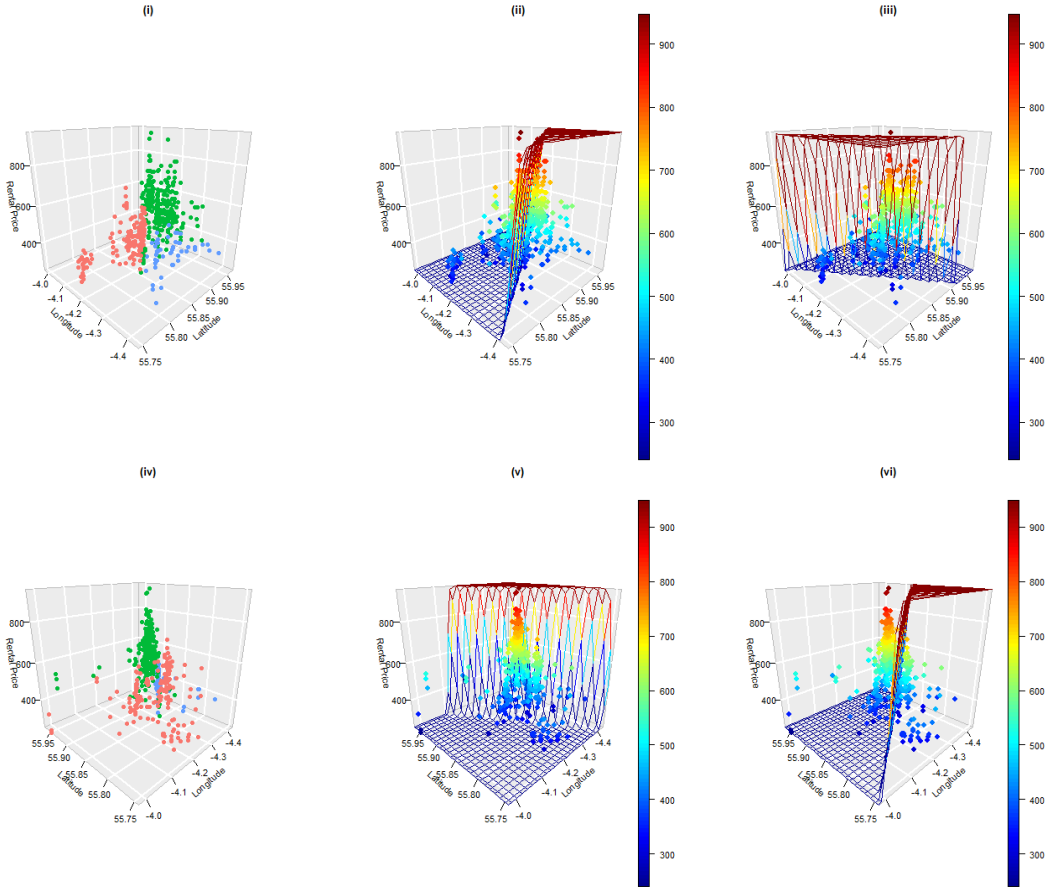


FIGURE 9.13: Visualisation of splits in the fitted HME model for the Glasgow rental price training data. Two angles represented by each row of plots. (i) and (iv) provide a view of average allocations with colors: Expert 1 - pink, Expert 2 - green, Expert 3 - blue.; (ii) and (v) illustrate the split at gate $G1$; (iii) and (vi) illustrate the split at gate $G2$. The planes shown in (ii), (iii), (v), and (vi) correspond to the logistic regression function for each gate evaluated on a grid, which corresponds to the ranges of explanatory variables, and scaled to the range of response variable.

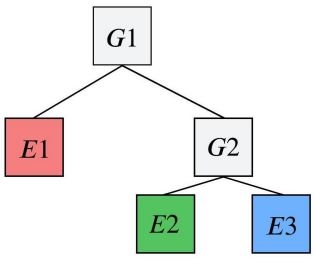


FIGURE 9.14: Architecture of the fitted HME model with 3 experts for the Glasgow rental prices training data.

Finally, the three-dimensional view of the resulting model is represented by the average prediction plane shown in Figures 9.15 and 9.16. The animated version of the figure can be accessed here. The first figure provides the most intuitive view of the fitted model, where a steep change in the prediction plane is observed where the previously seen Expert 1 and Expert 3 meet and the peak corresponding to Expert 2 emerges in the background (see Figure 9.8 for comparison). The main peak thus corresponds to the rental prices in the central and west locations of the city while the smaller peak captures the price drop evident in the central south of Glasgow. Overall, the fitted model appears to represent the relationships noted in the explanatory stage of the analysis as anticipated. Having assessed the performance of RJ MCMC, the addition of gate swaps is investigated next.

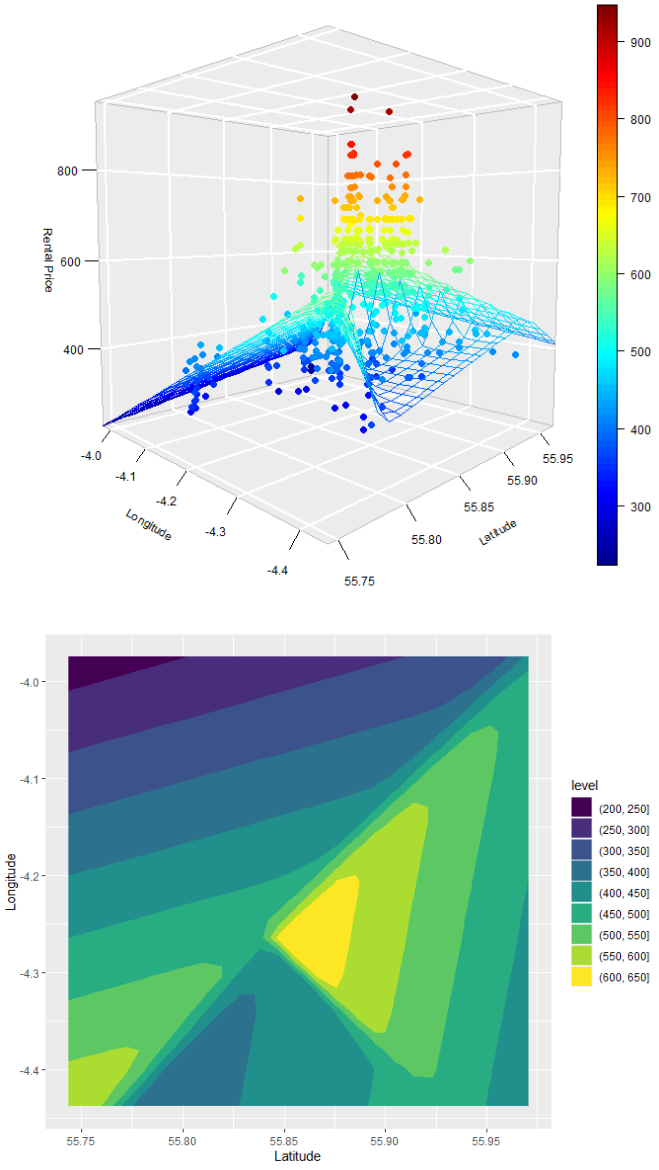


FIGURE 9.15: Average RJ MCMC fitted plane and the corresponding contour plot for the Glasgow rental prices training data. Predictions made on a 25×25 grid. The animated version of the figure can be accessed here.

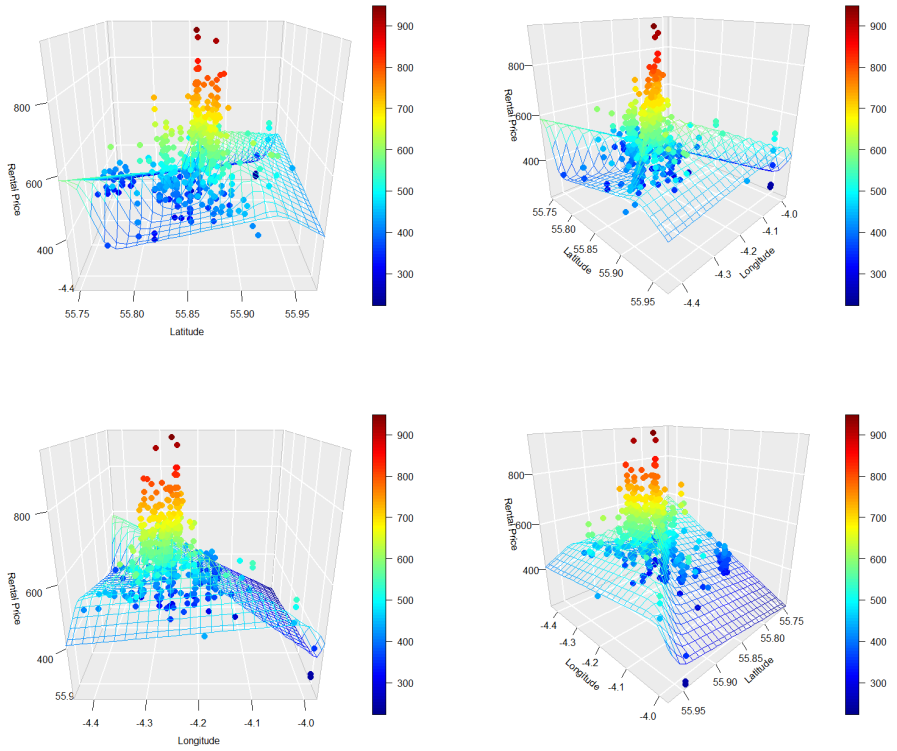


FIGURE 9.16: Average fitted plane for the Glasgow rental prices training data shown from additional angles. Predictions made on a 25×25 grid. The animated version of the figure can be accessed here.

9.5 RJ GS MCMC Results for Glasgow Rental Prices

It has been shown that RJ MCMC settles on a model with 3 experts in it. Given such a small number of experts in the tree, the addition of gate swaps may prove to be excessive, however, it may provide some guidance on when RJ GS MCMC performs best and should be used.

In this application, the frequency of the reversible jumps remains unchanged with one jump proposed every 10th iteration. It has been shown in Section 7.5 that frequent gate swap proposals tend to drive the overall number of experts in the model down. On the other hand, the frequency of swap proposals should be sufficiently high in order to have a notable effect. Thus, gate swaps are proposed every 25th iteration and the results are summarised in Table 9.4.

TABLE 9.4: Acceptance rates for the implementation of reversible jump only (equivalent to Table 6.5) and with the addition of gate swaps algorithm for the Glasgow rental prices training data. Mean squared error obtained on the test data set.

	Splits	**Merges**	**All Jumps**	**Swaps**
RJ MCMC				
Number Proposed	60	50	110	-
Number Accepted	9	7	16	-
Acceptance	15.00%	14.00%	14.55%	-
Mean Squared Error			0.4776	
RJ GS MCMC				
Number Proposed	59	51	110	12
Number Accepted	22	21	43	4
Acceptance	37.29%	41.18%	39.09%	33.33%
Mean Squared Error			0.6096	

Consistent with findings made in Chapter 7, the introduction of gate swaps increases the acceptance rates of the reversible jumps. Even though gate swap proposals do not change the number of experts in the tree, it is evident that successful proposals are followed by an increased reversible jump activity. It can be seen that a third of the proposed swaps have been accepted. Such high acceptance rates might indicate proposals that improve mixing in architecture, however, do not have a large effect on the overall likelihood of the model tree.

Figure 9.17 reveals that the addition of gate swaps encourages exploring trees consisting of up to 5 experts with the most time spent in trees with 2 experts only. The latter finding is expected given the previously established association between the introduction of gate swaps and a decrease in the average number of experts in the tree. Unlike the RJ MCMC, there is no evidence of RJ GS MCMC settling on a certain amount of experts (Figure 9.18). Since reversible jumps continue to be accepted throughout the duration of the MCMC, there is no meaningful way to summarise the posterior model parameters as seen for the RJ MCMC. For the property data application, the gate swaps appear to cause more disturbance than improvement, which is confirmed by the produced mean squared error, that is notably higher than the one achieved without the gate swaps (Table 9.4).

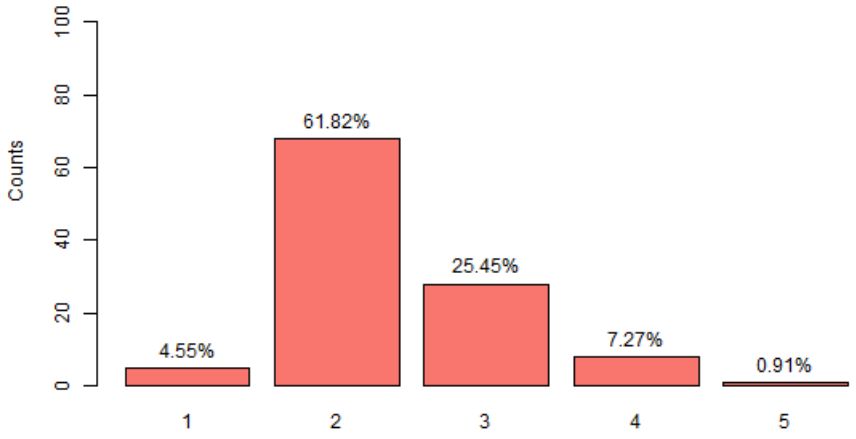


FIGURE 9.17: Distribution of the number of experts in the RJ SG MCMC chain with initial start of 1 expert for the Glasgow rental prices training data.

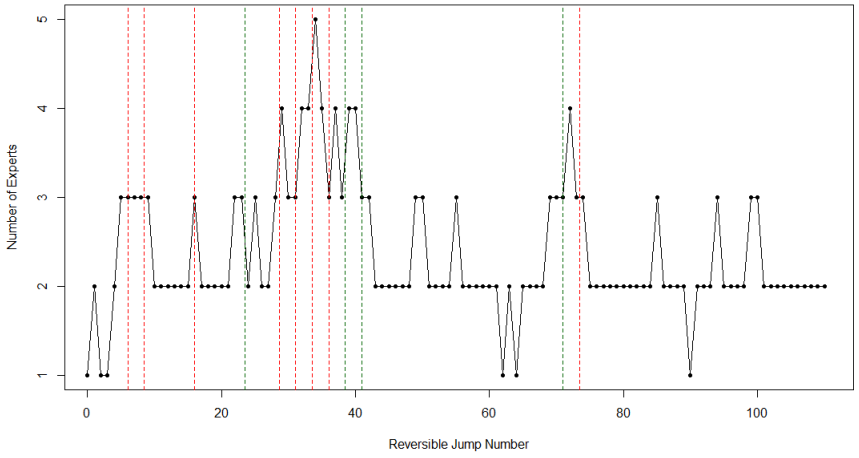


FIGURE 9.18: Number of experts in the tree after each reversible jump proposal for the Glasgow rental prices training data. Gate swap proposals are marked as dashed lines with green indicating an accepted and red indicating a rejected gate swap proposal.

RJ GS MCMC results in the predictions plane shown in Figures 9.19 and 9.20 with the animated version available here. The decrease in the average number of experts across MCMC iterations creates a model fit, in which the rental prices appear to simply increase towards the center of Glasgow. The latter is of course true, however, fails to capture the complex relationships across the east, west, and south areas of Glasgow. An additional RJ GS MCMC run has been created in order to obtain the Gelman-Rubin statistic as per Section 5.5 resulting in the value of 1.07, which indicates that despite the changing architecture, convergence in predictions has been achieved.

Overall, the results obtained in this section are not surprising given the small number of experts in the HME model tree. For this application, the improvement in the exploration of model architecture space does not appear to be worth the loss of modeling and prediction accuracy. The next section evaluates the HME model fitted using the RJ MCMC against two competitors.

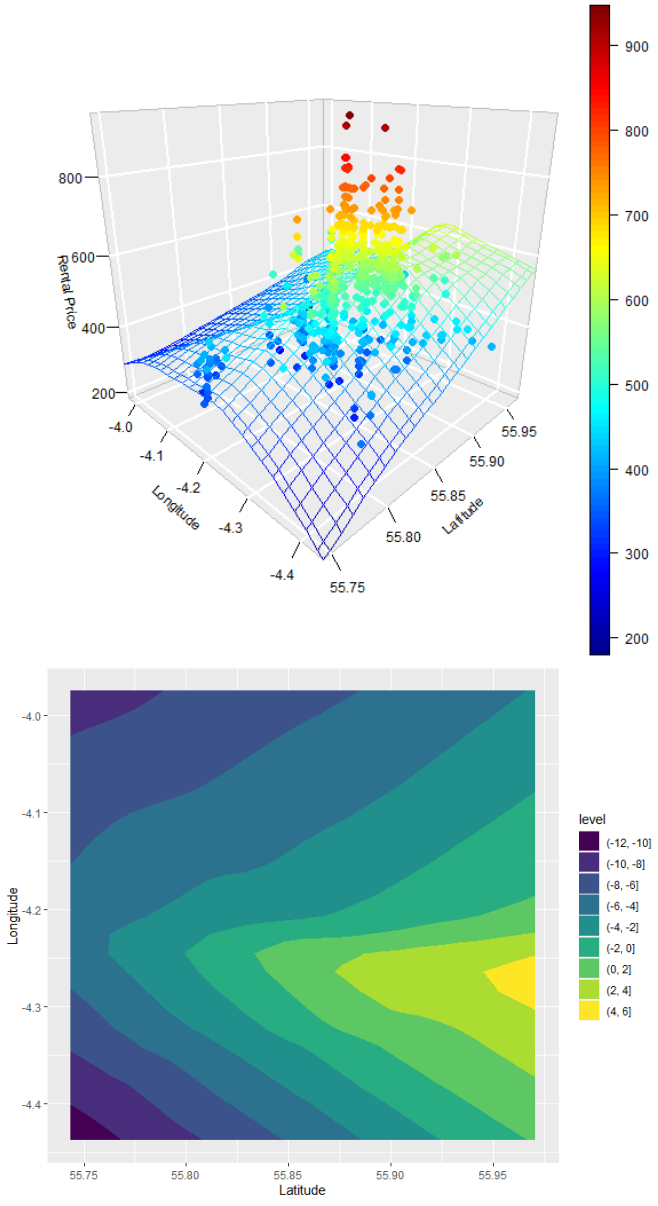


FIGURE 9.19: Average RJ GS MCMC fitted plane and the corresponding contour plot for the Glasgow rental prices training data. Predictions made on a 25×25 grid. The animated version is available here.

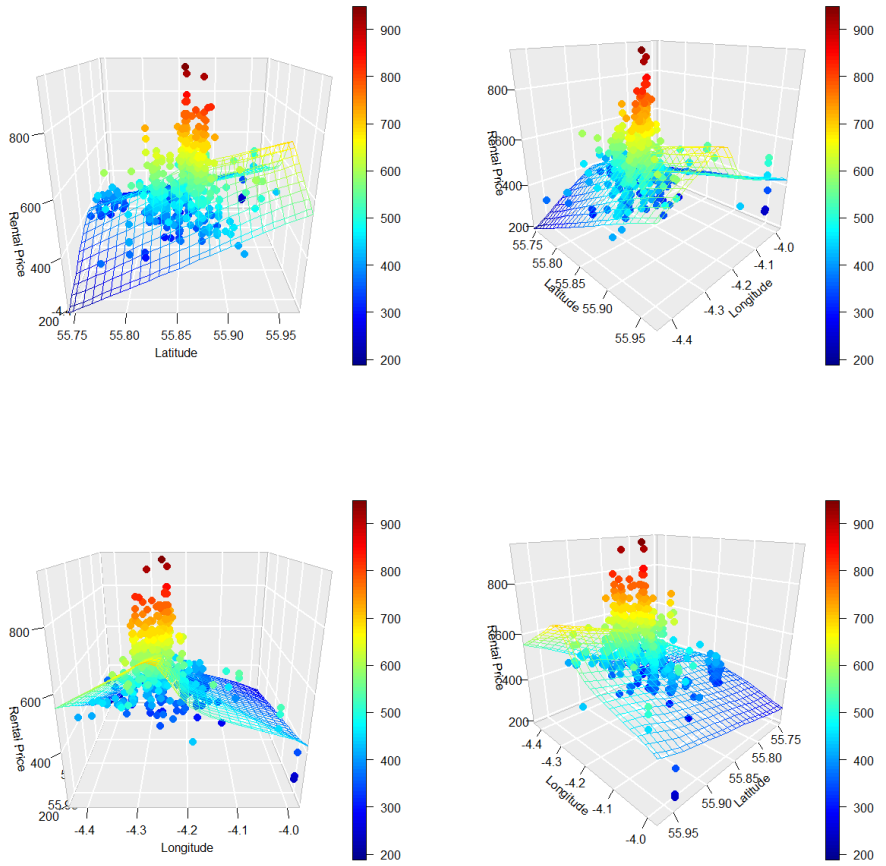


FIGURE 9.20: Average RJ GS MCMC fitted plane the Glasgow rental prices training data view from additional angles. Predictions made on a 25×25 grid. The animated version is available here.

9.6 HME Performance against Competitors for Glasgow Rental Prices

9.6.1 Competitor Model Fitting Details

In this section, the fitted HME model is compared to two competitors - Generalised Additive Models (GAM) and Bayesian Additive Regression Trees (BART) (please see Chapter 8 for details). For both competitors, the models are fitted on the training data set while their predictive performance is evaluated on the test set. As before, the data is standardised with the monthly rental price acting as the response variable, where each y_i corresponds to the monthly rental price of the i-th property. The longitude (x_{1i}) and latitude (x_{2i}) of the property are treated as the explanatory variables forming a vector $\mathbf{x}_i = (x_{1i}, x_{2i})$, which corresponds to the geographical location of the i-th property.

The following GAM model is fitted to the property data in `R` using the default settings of the function `gam` from the package `gam`:

$$Rental\ Price_i = f_1(longitude_i) + f_2(latitude_i) + f_3(longitude_i, latitude_i) + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. For simplicity, the default smoothing splines with 4 degrees of freedom and the default smoothing parameter equal to 1 are used to represent the relationship between the rental price and the longitude and latitude of the property as well the interaction between the two (R Documentation, 2023).

BART is fitted to the property data in `R` using the default settings of the function `bart` from the package `BayesTree`. BART is a sum of trees model, where each tree is constrained by a prior, which is controlled by power and base parameters with the default values of 2 and 0.95, respectively. This prior is recommended to ensure that each individual tree is a weak learner that contributes only a small amount to the overall fit (R Documentation, 2022). As seen for HME, BART model is also run for $1,100$ iterations with the first 100 discounted as burn-in.

9.6.2 Competitor Evaluation against HME

The predictive performance of the competitor models is summarised in Table 9.5. It can be seen that the HME model outperforms GAM and comes close to the BART model (MSE difference of 0.032). The predictive performance results are consistent with those seen for the motorcycle accident data.

TABLE 9.5: Mean squared error obtained from the predictions made on the test data set of the property rental price data for the following models - Hierarchical Mixture of Experts (HME), Generalised Additive Model (GAM) and Bayesian Additive Regression Tree (BART).

	HME	**GAM**	**BART**
Mean Squared Error	0.4776	0.6111	0.4452

For buy-to-let investors in Glasgow, it is crucial to understand how the geographical location of the property affects the rental price. GAM provides a limited level of interpretability that can be achieved by separating the additive smooth components for longitude and latitude and plotting them against the rental price. The visualisation of interaction terms, however, is complicated and less intuitive (see Noam Ross (2022)

for GAM interaction visualisation ideas). Furthermore, GAM does not account for heteroscedasticity present in the response and it has been shown that there is indeed different variability in the rental prices present across distinct areas of Glasgow. Finally, GAM appears to have the largest mean squared error and thus the worst predictive performance out of the models considered.

While BART does have the best predictive performance in terms of the mean squared error, the model does not account for heteroscedasticity and is viewed as more of a *black-box* method. Hence BART does not provide the same level of interpretability as that presented for the HME model (see Section 9.4). The trade-off between prediction accuracy and interpretability is a well-known and discussed topic in machine learning and most of the time the choice comes down to the main goal of the model fitting process (Weller et al., 2021) . In the case of the Glasgow rental property market problem, the additional information provided by the nature and design of HME models is of utmost importance to the first-time buy-to-let investors, who want to develop an understanding of the market as a whole as well as make an accurate prediction. It is thus believed that, given how close the mean squared errors of HME and BART are, such buy-to-let investors would prefer using the HME model.

9.7 Summary

In this chapter, hierarchical mixture of experts model has been fitted to the data set containing rental property prices in Glasgow. An initial investigation of the data revealed complex relationships between the geographical location of the property and its rental price across different areas of the city. An initial impression of such relationships was formed and confirmed by the fitted HME model. It has been shown that a high reversible jump acceptance rate of 14.55% has been achieved demonstrating the success of the devised reversible jump proposals. The automatic architecture selection methods lead to a preferred number of three experts in the tree. Results produced by the MCMC chain during the model fitting process have been presented and discussed to showcase the interpretability of HME model. The latter included looking at the latent assignment variables for the considered properties as well as the mixing proportions associated with the three experts. The soft probabilistic splits, evident for several properties located on the borders between the experts, were showcased and discussed when assessing the abruptness of the tree splits. Next, the estimated posterior mean parameter values were presented and interpreted. The fitted posterior mean variance parameters illustrated HME model's ability to capture heteroscedasticity present in the rental prices across the

three experts. Visualisation of the locations for splits allowed for an in-depth understanding of the model as well as reconstructing the architecture of the fitted HME tree. The gate swaps have, however, proved to be an excessive addition to the model fitting algorithm for this application. Finally, the performance of HME model was compared to GAM and BART models. It has been shown that HME model outperformed GAM and came close to BART in terms of prediction capability as well as demonstrated the added benefits of accounting for heteroscedasticity and a high level of model interpretability.

There is scope for extending the example presented in this chapter to a multivariate scenario. For example, one could investigate rental price changes over time as well as add additional explanatory variables such as which floor the property is located on, if it boasts of having a garden, the year the property was built, and many more.

Overall, the rental prices data helped illustrate that HME models are a powerful tool that offers a high level of model interpretability as well as accurate predictions. Using the fitted HME model, the first-time investor could firstly make a reliable estimate of the expected rental income for a particular property. Secondly, they would be able to interrogate the model output to gain an in-depth understanding of the model representing the rental market in Glasgow. The next chapter offers an overview of the work undertaken throughout this thesis as well as discusses potential future extensions of the work.

Chapter 10

Conclusions, Discussion and Future Work

10.1 Main Goal

From the outset of the work undertaken throughout this thesis, a lack of a flexible and automatic architecture selection technique is identified as the main challenge faced by the HME models. It is demonstrated that pre-setting HME model architecture can lead to poor mixing and issues with convergence. A commonly used approach to architecture selection involves trying several architectures out and using methods such as cross-validation to pick the best one. Such a method is computationally intensive and still requires the user to pick the set of architectures to consider. The process of deciding on how many nodes the model tree should have and how these nodes should be arranged yields an unmanageable number of options. Choosing model architectures in advance also requires setting initial parameter values, which becomes progressively challenging as parameter dimensionality increases. Thus, the work presented in this thesis aims to propose automatic architecture selection methods, which would allow for both adding and removing tree nodes as well as adjusting the order of the existing nodes.

10.2 Reversible Jump Methodology

Growing and pruning the HME model trees results in an ever-changing model parameter dimensionality, which requires a flexible and constructive way to move between the plausible models. The reversible jump algorithm, used for the construction of reversible Markov chains, stands out as a great candidate for the task. It is shown that for HME

models, the naive reversible jump algorithm suffers from low acceptance rates. In general, the cause of low RJ acceptance rates usually lies in uninformed jump proposals (Al-Awadhi et al., 2004; Ehlers and P. Brooks, 2008; Farr et al., 2015; Brooks et al., 2003). To address this issue for HME models, a method for proposing intelligent jumps is developed resulting in the reversible jump acceptance rate increase from 0.6% to 12.2% for one of the applications presented in the thesis. The improvement in the acceptance rates is shown to be consistent across multiple applications and tuning parameter selections. It is also demonstrated that the addition of the reversible jump can improve mixing and convergence as well as allow for escape from an unhelpful initial state. The proposed automatic architecture selection method appears to yield a consistent model fit on average irrespective of the reversible jump frequency and the initial state. The adaptation of the reversible jump methodology with the addition of the reversible jump proposal generation algorithm thus forms the desired solution for automatically growing and pruning the HME model trees.

10.3 Gate Swap Methodology

The reversible jump operates in the leaves of the tree and thus does not allow for sudden dramatic changes inside the model architecture. Such changes are achieved by implementing a novel idea, which consists of swapping the order of existing nodes. The latter is carried out by the addition of the gate swap algorithm to the reversible jump MCMC. It is demonstrated that the gate swaps result in proposing architectures that would not have been considered otherwise. This seems to further improve mixing in architecture with consistently high acceptance rates suggesting the increased exploration of model architecture space. The gate swaps also introduce the option of escaping previously made unbeneficial splits in one step and hence encourage simpler models with fewer experts in the tree. It is observed that gate swaps can cause notable disturbance to the model architectures and are often followed by increased reversible jump activity. Throughout the evaluation of the reversible jump gate swap MCMC, it becomes apparent that gate swaps are better suited for deep trees. Firstly, larger trees are more likely to contain unfortunate splits and hence benefit from the swaps. Secondly, the bigger the tree the more gates there are available for a swap, which creates a larger pool of architectures to be proposed. Given that the addition of gate swaps is not expensive in terms of run time, it is recommended to fit HME models using both the RJ MCMC and the RJ GS MCMC to evaluate which model fitting technique is most suitable for the application in question.

10.4 Performance Against Competitors

In this thesis, HME models are fitted to two real-life applications with both models evaluated against two competitors - GAM and BART. In both cases, the HME model outperforms the GAM and comes close to the BART in terms of its predictive performance. The key differentiating features offered by HME are the ability to account for heteroscedasticity as well as offer a high level of interpretability of the fitted model. As well as creating accurate predictions, HME model allows for an in-depth understanding of the fitted model. The latter includes insight into the distribution of the number of experts across the MCMC runs, the latent assignment variables, the abruptness and location of the separation between the experts as well as access to the normal expert density parameters. The interpretability of the model is showcased with impact when modeling rental prices in Glasgow. The problem is approached from the first-time buy-to-let investor's point of view, which requires developing an understanding of Glasgow rental market as a whole as well as making a reliable prediction. It is shown that the model captures complex relationships between the geographical location and the rental price of properties in Glasgow. The investor can thus identify areas that exhibit a steep price change in close proximity as well as identify general peaks in rental prices. All of the above results in a more desirable outcome than that produced by a prediction-driven black-box method.

10.5 Potential Applications

As seen throughout this thesis, HME models are particularly useful in applications with evident change points, varying degrees of smoothness, and complex relationships between the response and input variables. In addition, HME models are very well-suited for modeling a heteroscedastic response. In practice, the complexity of the relationship between the response and explanatory variables is rarely uniform across the whole problem space. In fact, a complex relationship is often present in a small portion of the problem space with a rather simple relationship evident elsewhere. It is thus preferable to spend relatively more effort sampling in the areas where such relationships are more *interesting*. The structural makeup of HME models allows for doing just that by fitting a simple model in the majority of the problem space and concentrating on the remaining complex subproblem. In addition to applications covered in this thesis, consider an example, tackled by Gramacy (2015), which exhibits the described characteristics. The application involves a new reusable rocket booster, developed by NASA and called the Langley Glide-Back Booster (LGBB). It is of interest to learn about the LGBB response in flight characteristics as a function of a selected number of inputs. In particular, the

relationship between the lift response and speed (Mach) and angle of attack (alpha) with the side-slip angle (beta) fixed at zero is investigated (Figure 10.1).

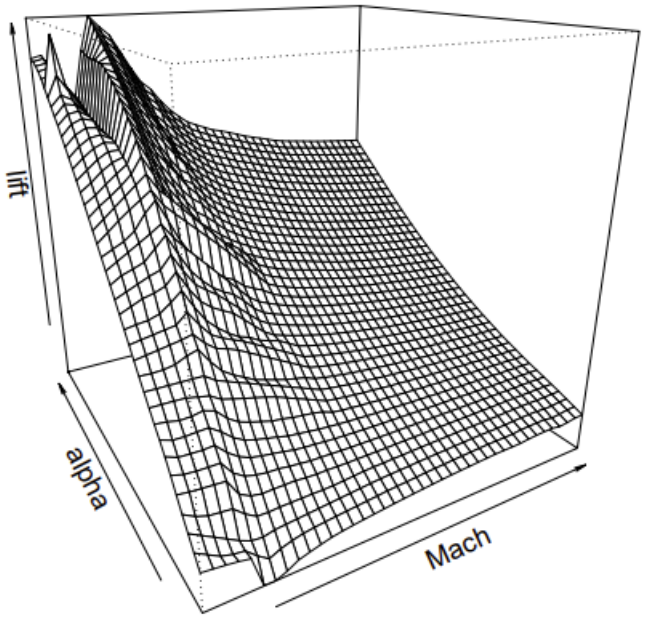


FIGURE 10.1: LGBB in flight Lift plotted as a function of Mach (speed) and alpha (angle of attack) with beta (side-slip angle) fixed to zero. Source: Gramacy (2015).

It is evident that in the majority of the problem space, the relationship between the response variable lift and the two covariates alpha and Mach is simple and could potentially be represented by a smooth plane. On the other hand, there are several peaks and dips present in the lift for smaller Mach values forming a more complex relationship. In addition to the Bayesian Treed Gaussian Process model proposed by the author, such relationships could be represented by an HME model, which would partition the problem space and solve subproblems accordingly.

Further potential HME applications can be observed in the area of geology. For example, surface elevation application discussed by Davis (2002), where varying levels of elevation are modeled as a function of geographical location. HME models could also be used to model complex relationships and varying levels of smoothness often appearing in the area of molecular design. For instance, HME could be considered as an alternative to the currently used polynomials in modeling the relationship between the docking studies and the biological activity of δ-selective enkephalin analogues (Sapundzhi et al., 2015).

10.6 Future Research

Although at this time the focus is placed on showcasing the model interpretability for low dimensional problems, future research could investigate fitting HME models using the RJ GS MCMC on problems of higher dimensionality. Another potential extension of the work presented consists of further development of the reversible jump algorithm, which would allow for merging all sibling nodes as opposed to merging expert siblings only. Such jumps would propose more drastic pruning and growing of the trees. The latter idea, however, also entails a nontrivial task of developing forward jump proposals that would have the functionality to replace any expert with a tree that is also hierarchical.

Although the expert parameter sampling methods proposed throughout this thesis concentrate on Gaussian experts, the methodology could be extended to suit GLM and other experts. The latter would broaden the suitable applications and potentially prove to be a worthwhile addition to the smaller number of available competitor methods. Borrowing some ideas from GAM, new methods for fitting simple smooth functions at the leaves of the tree could also be considered.

Although the individual expert densities in the tree are usually of the same form, they do not need to be. A mixture of different expert densities could potentially improve HME model fit even further. The above, of course, would require developing the methodology for flexible parameter sampling as well as picking the expert density to be used when introducing a new expert.

The HME model architecture also allows for the approach to be used to solve classification problems. A continuous response variable, seen in this thesis, could be replaced by a categorical response variable defining the classes. The reversible jump and gate swap methodology could be used to perform automatic partitioning of the problem space, based on the explanatory variables, in the same way as seen throughout this thesis. On the other hand, the output produced by experts would need to be categorical, which could be achieved by fitting appropriate models at the leaves of the tree, i.e., logistic, multinomial or ordinal regression.

RJ GS MCMC for HME models could be further extended to handle cases with multivariate outcomes. Approaches that allow for modeling multiple responses at the same time are especially required in fields such as medicine, where it is of interest to predict a set of correlated variables (for an application in psychiatry see Teixeira-Pinto et al., 2009). Given that the path probabilities in HME models do not depend on the response variables, only expert densities alongside the corresponding parameter proposals and updates would require adaptation. For example, methods, similar to the latent variable

model for multiple outcomes discussed in Sammel et al. (1999), could be used in the leaves of HME trees.

Developing a methodology for dealing with missing values when fitting HME models is another area of potential future research. Unlike HME, alternative tree-based methods, such as Classification and Regression Trees (CART), use the so-called *surrogate splits* to tackle the problem of missing values (Breiman, 1984). Feelders (2000) describes the function of such surrogate splits as picking in which direction to send an observation with a missing value based on determining to what extent the potential splits resemble the best split in terms of the number of cases that they send the same way. Such an observation is then assigned based on the most resembling surrogate split. In the case of HME, the complication arises from all input variables being used in the calculations performed by both gate and expert nodes.

In general, when considering missing data mechanisms, one is first required to think about the reason behind the missingness (Lai, 2019). For example, the data could be missing completely at random (MCAR), which means that the missingness process is unrelated to the research question, and thus, in general, observations with missing variables can be omitted. Thus, under MCAR, no additional HME-specific methodology is required. Difficulties, however, arise when the probability of missing observations can be explained by the observed data; such a missingness process is known as missing at random (MAR). For MAR, in the cases where missing data is only present in the response, the missing observations can often be omitted, however, the presence of missing data in the input variables requires implementing more advanced methods (Lai, 2019). Finally, utmost care needs to be taken towards the non-ignorable missingness which occurs when conditions of MAR do not hold and thus valid inference can only be obtained by appropriately modeling the missingness mechanism.

A method known as listwise deletion, which uses complete cases only, can be used under MCAR and, in some cases, under MAR. An example of a fully Bayesian approach for missing data entails treating missing data as parameters with some prior information. Alternatively, methods, known as multiple imputation (MI), can be implemented to predict the missing values using the observed data (Buuren, 2012; Demirtas, 2018). In a Bayesian setting, MI methods capture the uncertainty around the produced predictions by imputing multiple data sets and getting the corresponding samples from the posterior distributions of the missing values. Such samples are then used to assign the missing values. Taking into account the nature of missingness processes, future research for HME could thus focus on treating the missing data as parameters, exploring imputation methods as well as developing novel approaches.

10.7 Conclusion

The work carried out throughout this thesis resulted in a functional methodology for automatically selecting and adjusting hierarchical mixture of experts model architecture. The latter included

1. Proposing and evaluating three Bayesian sampling strategies for HME models with Gaussian experts resulting in the recommended approach.
2. Adapting the reversible jump methodology to HME models.
3. Developing methodology for improving on low naive reversible jump acceptance rates for HME models.
4. Proposing algorithm, which outlines forward and backward jump steps for HME models with Gaussian experts.
5. Introducing the gates swaps algorithm for adjusting the existing HME tree architecture.

The above contributions have led to improvement in mixing and convergence as well as sensitivity to starting values for considered HME model applications. It has been demonstrated that the hierarchical mixture of experts can offer accurate predictions, accountability of heteroscedasticity as well a high level of model interpretability.

Appendix A

Iteratively Weighted Least Squares Algorithm

A.1 IWLS Algorithm

An iteratively weighted least squares (IWLS) problem with a solution $\boldsymbol{\alpha}^*$ can be written as

$$\boldsymbol{\alpha}^* = \left(X^T W(\boldsymbol{\alpha}) X\right)^{-1} X^T W(\boldsymbol{\alpha}) Z(\boldsymbol{\alpha}),$$

where X is the design matrix, $W(\boldsymbol{\alpha})$ is the weight matrix and $Z(\boldsymbol{\alpha})$ is the response vector. As per Dutang (2017), the IWLS algorithm can then be written as

1. Initialisation:
   (a) Use original data and add a small shift $y + 0.1$.
   (b) Compute working responses $Z(\boldsymbol{\alpha})^{(0)}$.
   (c) Compute working weights $W(\boldsymbol{\alpha})^{(0)}$.
   (d) Solve this system of linear equations to get $\boldsymbol{\alpha}^{(0)}$

$$X^T W(\boldsymbol{\alpha})^{(0)} X \boldsymbol{\alpha}^{(0)} = X^T W(\boldsymbol{\alpha})^{(0)} Z(\boldsymbol{\alpha})^{(0)}.$$

2. Iteration: for $t = 1, ..., T$:
   (a) Compute working responses $Z(\boldsymbol{\alpha})^{(t)}$.

(b) Compute working weights $W(\boldsymbol{\alpha})^{(t)}$.

(c) Solve this system of linear equations to get $\boldsymbol{\alpha}^{(t+1)}$

$$X^T W(\boldsymbol{\alpha})^{(t)} X\boldsymbol{\alpha}^{(t+1)} = X^T W(\boldsymbol{\alpha})^{(t)} Z(\boldsymbol{\alpha})^{(t)}.$$

A.2 QR Decomposition for IWLS Algorithm

In practice, the linear equations stated in the IWLS algorithm steps 1d) and 2c) are solved using the QR decomposition (Green, 1984). To simplify notation, let $W(\boldsymbol{\alpha}) = W$ and $Z(\boldsymbol{\alpha}) = Z$. The equations can then be simplified as follows

$$\begin{aligned} X^T W X &= X^T W Z \\ X^T W^{\frac{1}{2}} W^{\frac{1}{2}} X &= X^T W^{\frac{1}{2}} W^{\frac{1}{2}} Z \\ \tilde{X}^T \tilde{X} &= \tilde{X}^T \tilde{Z}, \end{aligned}$$

where $\tilde{X} = W^{\frac{1}{2}} X$ and $\tilde{Z} = W^{\frac{1}{2}} Z$. Next, perform a QR decomposition on $\tilde{X}$, i.e. write $\tilde{X} = QR$. Then

$$\begin{aligned} \tilde{X}^T \tilde{X} &= \tilde{X}^T \tilde{Z} \\ (QR)^T QR &= (QR)^T \tilde{Z} \\ R^T Q^T QR &= R^T Q^T \tilde{Z} \\ R^T R &= R^T Q^T \tilde{Z} \\ R &= Q^T \tilde{Z}, \end{aligned}$$

where $Q^T Q = I$, because Q is an orthonormal matrix. The QR decomposition can be exploited further in the calculation of the variance-covariance matrix Σ:

$$\Sigma = (X^T W X)^{-1} = (\tilde{X}^T \tilde{X})^{-1} = ((QR)^T QR)^{-1} = (R^T Q^T QR)^{-1} = (R^T R)^{-1}. \quad \text{(A.1)}$$

Appendix B

Gating Parameter Estimation Details

B.1 Indicator Log-likelihood Function

As seen previously, the indicator likelihood, which is equivalent to the joint density of the allocation variables $\mathbf{z}^{(G)}$ given the parameters $\boldsymbol{\gamma}^{(G)}$, for the points that have reached gate G can be written as

$$L\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) = \prod_{i=1}^{n}\prod_{H}\left(\pi_i^{(G,H)}\right)^{z_i^{(G,H)}},$$

It follows that the corresponding log-likelihood function is

$$\begin{aligned} l\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) &= \sum_{i=1}^{n}\sum_{H} z_i^{(G,H)} \log\left(\pi_i^{(G,H)}\right) \\ &= \sum_{i=1}^{n}\sum_{H} z_i^{(G,H)} \log\left(\frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\sum_{H'}\exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)}\right). \end{aligned}$$

Using the fact that $\sum_H z_i^{(G,H)} = 1$, the above can also be written as

$$\begin{aligned} l\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right) &= \sum_{i=1}^{n}\left[\sum_H z_i^{(G,H)}\left(\log\left(\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)\right) - \log\left(\sum_H \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)\right)\right)\right] \\ &= \sum_{i=1}^{n}\left[\sum_H z_i^{(G,H)}\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i - \log\left(\sum_H \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)\right)\right)\right] \\ &= \sum_{i=1}^{n}\left[\left(\sum_H z_i^{(G,H)}\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right) - \log\left(\sum_H \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)\right)\right]. \end{aligned} \tag{B.1}$$

B.2 Score Function

Let $S\left(\boldsymbol{\gamma}^{(G)}\right)$ denote a score function containing the first derivatives of the indicator log-likelihood function. The score function can be obtained by differentiating (B.1) as follows

$$\begin{aligned} S\left(\boldsymbol{\gamma}^{(G,H)}\right) &= \frac{\partial l\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right)}{\partial \boldsymbol{\gamma}^{(G,H)}} \\ &= \sum_{i=1}^{n}\mathbf{x}_i\left[\sum_H z_i^{(G,H)} - \frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\sum_{H'}\exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)}\right] \\ &= \sum_{i=1}^{n}\mathbf{x}_i\left[\sum_H z_i^{(G,H)} - \pi_i^{(G,H)}\right] \\ &= X^T\left(\mathbf{z}^{(G,H)} - \boldsymbol{\pi}^{(G,H)}\right), \end{aligned}$$

where $\mathbf{z}^{(G,H)} = \left(z_1^{(G,H)}, ..., z_n^{(G,H)}\right)^T$ and $\boldsymbol{\pi}^{(G,H)} = \left(\pi_1^{(G,H)}, ..., \pi_n^{(G,H)}\right)^T$. A vector of first derivatives for all H descending from G is thus obtained as

$$S\left(\boldsymbol{\gamma}^{(G)}\right) = \left(X^T\left(\mathbf{z}^{(G,H)} - \boldsymbol{\pi}^{(G,H)}\right)\right)_H^T = X^{\#T}\left(\mathbf{z}^{(G)} - \boldsymbol{\pi}^{(G)}\right), \tag{B.2}$$

where $X^{\#}$ is the original matrix X repeated as many times as there are splits at gate G, $\mathbf{z}^{(G)} = \left(\mathbf{z}^{(G,H)}\right)_H^T$ and $\boldsymbol{\pi}^{(G)} = \left(\boldsymbol{\pi}^{(G,H)}\right)_H^T$.

B.3 Hessian Matrix

Let $H\left(\boldsymbol{\gamma}^{(G)}\right)$ denote a Hessian matrix containing the second derivatives of the indicator log-likelihood function. The derivation of the Hessian matrix requires obtaining the derivatives of the mixing proportion

$$\pi_i^{(G,H)} = \frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\sum_{H'}\exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)}. \tag{B.3}$$

with respect to the gating parameters. This problem can be further split into two cases. First, differentiating $\pi_i^{(G,H)}$ with respect to $\boldsymbol{\gamma}^{(G,H)}$ (Case 1). Second, differentiating $\pi_i^{(G,H)}$ with respect to $\boldsymbol{\gamma}^{(G,H'')}$, where $H \neq H''$ (Case 2).

Case 1

Let us write the denominator of (B.3) as follows:

$$\sum_{H'}\exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right) = \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right) + \underbrace{\sum_{H'\neq H}\exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)}_{A}.$$

The derivative of $\pi_i^{(G,H)}$ with respect to $\boldsymbol{\gamma}^{(G,H)}$ can then be obtained as follows

$$\begin{aligned}
\frac{\partial \pi_i^{(G,H)}}{\partial \boldsymbol{\gamma}^{(G,H)}} &= \frac{\partial}{\partial \boldsymbol{\gamma}^{(G,H)}}\left[\frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A}\right] \\
&= \frac{\mathbf{x}_i\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)\left(\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A\right) - \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)\mathbf{x}_i\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\left(\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A\right)^2} \\
&= \frac{\mathbf{x}_i\exp\left(2\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right) + A\cdot\mathbf{x}_i\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right) - \mathbf{x}_i\exp\left(2\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\left(\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A\right)^2} \\
&= \frac{A\cdot\mathbf{x}_i\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\left(\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A\right)^2} \\
&= \frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A}\mathbf{x}_i\cdot\frac{A}{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A} \\
&= \pi_i^{(G,H)}\cdot\mathbf{x}_i\cdot\frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A-\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)+A}
\end{aligned}$$

$$= \pi_i^{(G,H)} \cdot \mathbf{x}_i \cdot \left(1 - \frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{1 + \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right) + A} \right)$$
$$= \pi_i^{(G,H)} \left(1 - \pi_i^{(G,H)}\right) \mathbf{x}_i.$$

Case 2

Let H and H'' be two different nodes, then the derivative of $\pi_i^{(G,H)}$ with respect to $\boldsymbol{\gamma}^{(G,H'')}$ is

$$\begin{aligned}
\frac{\partial \pi_i^{(G,H)}}{\partial \boldsymbol{\gamma}^{(G,H'')}} &= \frac{\partial}{\partial \boldsymbol{\gamma}^{(G,H'')}} \left[\frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\sum_{H'} \exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)} \right] \\
&= \frac{0 - \exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right) \mathbf{x}_i \exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G,H'')}\right)}{\left(\sum_{H'} \exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)\right)^2} \\
&= -\frac{\exp\left(\boldsymbol{\gamma}^{(G,H)^T}\mathbf{x}_i\right)}{\sum_{H'} \exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)} \cdot \frac{\exp\left(\mathbf{x}_i^T \boldsymbol{\gamma}^{(G,H'')}\right)}{\sum_{H'} \exp\left(\boldsymbol{\gamma}^{(G,H')^T}\mathbf{x}_i\right)} \mathbf{x}_i \\
&= -\pi_i^{(G,H)} \pi_i^{(G,H'')} \mathbf{x}_i.
\end{aligned}$$

Combining the two cases together, we have

$$\frac{\partial \pi_i^{(G,H)}}{\partial \boldsymbol{\gamma}^{(G,H)}} = \pi_i^{(G,H)} \left(1 - \pi_i^{(G,H)}\right) \mathbf{x}_i, \qquad \text{and} \qquad \frac{\partial \pi_i^{(G,H)}}{\partial \boldsymbol{\gamma}^{(G,H'')}} = -\pi_i^{(G,H)} \pi_i^{(G,H'')} \mathbf{x}_i, \quad \text{(B.4)}$$

with $H \neq H''$. Analogously, having obtained (B.4), the derivation of the second derivatives of the gating parameter log-likelihood function is split into two cases.

Case 1

$$\begin{aligned}
\frac{\partial^2 l\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right)}{\partial\boldsymbol{\gamma}^{(G,H)}\partial\boldsymbol{\gamma}^{(G,H)^T}} &= \frac{\partial}{\partial\boldsymbol{\gamma}^{(G,H)}}\left[\sum_{i=1}^{n}\mathbf{x}_i\left[\sum_{H} z_i^{(G,H)} - \pi_i^{(G,H)}\right]\right] \\
&= \frac{\partial}{\partial\boldsymbol{\gamma}^{(G,H)}}\left[\sum_{i=1}^{n}\mathbf{x}_i\sum_{H} z_i^{(G,H)} - \sum_{i=1}^{n}\mathbf{x}_i\sum_{H}\pi_i^{(G,H)}\right] \\
&= \frac{\partial}{\partial\boldsymbol{\gamma}^{(G,H)}}\left[-\sum_{i=1}^{n}\mathbf{x}_i\sum_{H}\pi_i^{(G,H)}\right] \\
&= -\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T\left(\pi_i^{(G,H)}\left(1-\pi_i^{(G,H)}\right)\right) \\
&= X^T A^{(G,H)} X,
\end{aligned} \tag{B.5}$$

where $A^{(G,H)}$ is a $n\times n$ diagonal matrix with diagonal entries $A_{ii}^{(G,H)} = \pi_i^{(G,H)}\left(1-\pi_i^{(G,H)}\right)$ for $i = 1, ...n$.

Case 2

For some $H'' \neq H$

$$\begin{aligned}
\frac{\partial^2 l\left(\mathbf{z}^{(G)}|\boldsymbol{\gamma}^{(G)}\right)}{\partial\boldsymbol{\gamma}^{(G,H)}\boldsymbol{\gamma}^{(G,H'')^T}} &= \frac{\partial}{\partial\boldsymbol{\gamma}^{(G,H'')^T}}\left[\sum_{i=1}^{n}\mathbf{x}_i\left[\sum_{H} z_i^{(G,H)} - \pi_i^{(G,H)}\right]\right] \\
&= \frac{\partial}{\partial\boldsymbol{\gamma}^{(G,H'')^T}}\left[\sum_{i=1}^{n}\mathbf{x}_i\sum_{H} z_i^{(G,H)} - \sum_{i=1}^{n}\mathbf{x}_i\sum_{H}\pi_i^{(G,H)}\right] \\
&= \frac{\partial}{\partial\boldsymbol{\gamma}^{(G,H'')^T}}\left[-\sum_{i=1}^{n}\mathbf{x}_i\sum_{H}\pi_i^{(G,H)}\right] \\
&= -\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T\left(-\pi_i^{(G,H)}\pi_i^{(G,H'')}\right) \\
&= X^T A^{(G,H,H'')} X,
\end{aligned} \tag{B.6}$$

where $A^{(G,H,H'')}$ is a $n\times n$ diagonal matrix with diagonal entries $A_{ii}^{(G,H,H'')} = -\pi_i^{(G,H)}\pi_i^{(G,H'')}$ for $i = 1, ...n$.

Combining the two results (B.5) and (B.6), the Hessian matrix can then be written as

$$H\left(\boldsymbol{\gamma}^{(G)}\right) = \begin{bmatrix} X^T A^{(G,H_1)} X & \cdots & X^T A^{(G,H_1,H_r)} X \\ \vdots & \ddots & \vdots \\ X^T A^{(G,H_1,H_r)} X & \cdots & X^T A^{(G,H_r)} X \end{bmatrix} = X^{\#T} A^{\#} X^{\#}. \tag{B.7}$$

B.4 IWLS for Gating Parameter Estimation

The iteratively weighted least squares (IWLS) algorithm, outlined in Appendix A, is used to estimate the gating parameters $\boldsymbol{\gamma}^{(G)}$ (Green, 1984). The latter incorporates Newton's method (Fisher, 1925) where the following is iterated until convergence

$$\boldsymbol{\gamma}^{(G)^*} = \boldsymbol{\gamma}^{(G)} + \left(-H\left(\boldsymbol{\gamma}^{(G)}\right)\right)^{-1} S\left(\boldsymbol{\gamma}^{(G)}\right). \tag{B.8}$$

For illustrative purposes, assume that there are p covariates and r splits present at each gate in the model, the dimensions for the elements in (B.8) are then

$$\underbrace{\boldsymbol{\gamma}^{(G)^*}}_{rp\times 1} = \underbrace{\boldsymbol{\gamma}^{(G)}}_{rp\times 1} + \left(-\underbrace{H\left(\boldsymbol{\gamma}^{(G)}\right)}_{rp\times rp}\right)^{-1} \underbrace{S\left(\boldsymbol{\gamma}^{(G)}\right)}_{rp\times 1}.$$

In order to simplify the notation of results obtained for $S\left(\boldsymbol{\gamma}^{(G)}\right)$ and $H\left(\boldsymbol{\gamma}^{(G)}\right)$, let us drop the subscript G as

$$S\left(\boldsymbol{\gamma}\right) = X^{\#T}\left(\mathbf{z} - \boldsymbol{\pi}\right)$$

and

$$H\left(\boldsymbol{\gamma}\right) = X^{\#T} A^{\#} X^{\#},$$

respectively. The problem can then be reformulated as an iterative weighted least squares (IWLS) problem as follows

$$
\begin{aligned}
\boldsymbol{\gamma}^{*} &= \boldsymbol{\gamma} + \left(-H\left(\boldsymbol{\gamma}\right)\right)^{-1} S\left(\boldsymbol{\gamma}\right) \\
&= \underbrace{\left(-H\left(\boldsymbol{\gamma}\right)\right)^{-1}\left(-H\left(\boldsymbol{\gamma}\right)\right)\boldsymbol{\gamma}}_{\boldsymbol{\gamma}} + \left(-H\left(\boldsymbol{\gamma}\right)\right)^{-1} S\left(\boldsymbol{\gamma}\right) \\
&= \left(-H\left(\boldsymbol{\gamma}\right)\right)^{-1}\left[-H\left(\boldsymbol{\gamma}\right)\boldsymbol{\gamma} + S\left(\boldsymbol{\gamma}\right)\right] \\
&= \left(-X^{\#T} A^{\#} X^{\#}\right)^{-1}\left[\left(-X^{\#T} A^{\#} X^{\#}\right)\boldsymbol{\gamma} + X^{\#T}(\mathbf{z}-\boldsymbol{\pi})\right] \\
&= \left(-X^{\#T} A^{\#} X^{\#}\right)^{-1}\left[\left(-X^{\#T} A^{\#} X^{\#}\right)\boldsymbol{\gamma} + X^{\#T} A^{\#} A^{\#-1}(\mathbf{z}-\boldsymbol{\pi})\right] \\
&= \left(-X^{\#T} A^{\#} X^{\#}\right)^{-1}\left[-X^{\#T} A^{\#}\left(X^{\#}\boldsymbol{\gamma} - A^{\#-1}(\mathbf{z}-\boldsymbol{\pi})\right)\right] \\
&= \left(X^{\#T} A^{\#} X^{\#}\right)^{-1}\left[X^{\#T} A^{\#}\left(X^{\#}\boldsymbol{\gamma} - A^{\#-1}(\mathbf{z}-\boldsymbol{\pi})\right)\right] \\
&= \left(X^{\#T} A^{\#} X^{\#}\right)^{-1}\left[X^{\#T} A^{\#}\left(X^{\#}\boldsymbol{\gamma} - A^{\#-1}(\mathbf{z}-\boldsymbol{\pi})\right)\right],
\end{aligned}
$$

where $\boldsymbol{\gamma}^{*}$ is the IWLS problem solution with weights $A^{\#}$, response vector $X^{\#}\boldsymbol{\gamma} - A^{\#-1}(\mathbf{z}-\boldsymbol{\pi})$ and the design matrix $X^{\#}$. From here onwards, the steps of the IWLS algorithm can be implemented as per Appendix A.

Appendix C

Derivation of the Proposed Gating Parameter Density for Forward Jump

Given that $\boldsymbol{\gamma}_1$ and ϵ are independent, we have that

$$\begin{pmatrix} \epsilon \\ \boldsymbol{\gamma}_1 \end{pmatrix} \sim \mathsf{MVN}\left(\begin{pmatrix} 0 \\ \boldsymbol{\mu}_{\boldsymbol{\gamma}_1} \end{pmatrix}, \begin{pmatrix} \sigma_\epsilon^2 & 0 \\ 0 & \Sigma_{\boldsymbol{\gamma}_1} \end{pmatrix} \right).$$

It is evident that $\boldsymbol{\gamma} = (\gamma_0, \boldsymbol{\gamma}_1)^T$ can be written as a linear transformation of $(\epsilon, \boldsymbol{\gamma}_1)^T$ given $\mathbf{x}^*$ as

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_0 \\ \boldsymbol{\gamma}_1 \end{pmatrix} = \begin{pmatrix} 1 & -\mathbf{x}^* \\ 0 & I \end{pmatrix} \begin{pmatrix} \epsilon \\ \boldsymbol{\gamma}_1 \end{pmatrix} = \begin{pmatrix} -\mathbf{x}^{*T}\boldsymbol{\gamma}_1 + \epsilon \\ \boldsymbol{\gamma}_1 \end{pmatrix}.$$

One can work out the mean vector and the variance-covariance matrix of $\boldsymbol{\gamma}$ given $\mathbf{x}^*$ as follows

$$\begin{aligned} \mathbb{E}\left(\boldsymbol{\gamma}|\mathbf{x}^*\right) = \mathbb{E}\left(\begin{pmatrix} \gamma_0 \\ \boldsymbol{\gamma}_1 \end{pmatrix} \middle| \mathbf{x}^* \right) &= \begin{pmatrix} 1 & -\mathbf{x}^* \\ 0 & I \end{pmatrix} \mathbb{E}\left(\begin{pmatrix} \epsilon \\ \boldsymbol{\gamma}_1 \end{pmatrix} \right) \\ &= \begin{pmatrix} 1 & -\mathbf{x}^* \\ 0 & I \end{pmatrix} \begin{pmatrix} 0 \\ \boldsymbol{\mu}_{\boldsymbol{\gamma}_1} \end{pmatrix} \\ &= \begin{pmatrix} -\mathbf{x}^{*T}\boldsymbol{\mu}_{\boldsymbol{\gamma}_1} \\ \boldsymbol{\mu}_{\boldsymbol{\gamma}_1} \end{pmatrix}. \end{aligned}$$

$$\begin{aligned}
\text{Cov}\left(\boldsymbol{\gamma}|\mathbf{x}^*\right) = \text{Cov}\left(\left.\begin{pmatrix}\gamma_0\\ \boldsymbol{\gamma}_1\end{pmatrix}\right| \mathbf{x}^*\right) &= \begin{pmatrix}1 & -\mathbf{x}^*\\ 0 & I\end{pmatrix} \text{Cov}\left(\begin{pmatrix}\epsilon\\ \boldsymbol{\gamma}_1\end{pmatrix}\right) \begin{pmatrix}1 & -\mathbf{x}^*\\ 0 & I\end{pmatrix}^T \\
&= \begin{pmatrix}1 & -\mathbf{x}^*\\ 0 & I\end{pmatrix} \begin{pmatrix}\sigma_\epsilon^2 & 0\\ 0 & \Sigma_{\boldsymbol{\gamma}_1}\end{pmatrix} \begin{pmatrix}1 & 0\\ -\mathbf{x}^* & I\end{pmatrix} \\
&= \begin{pmatrix}\sigma_\epsilon^2 & -\mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1}\\ 0 & \Sigma_{\boldsymbol{\gamma}_1}\end{pmatrix} \begin{pmatrix}1 & 0\\ -\mathbf{x}^* & I\end{pmatrix} \\
&= \begin{pmatrix}\sigma_\epsilon^2 + \mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1}\mathbf{x}^* & -\mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1}\\ -\mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1} & \Sigma_{\boldsymbol{\gamma}_1}\end{pmatrix}.
\end{aligned}$$

Hence

$$\begin{aligned}
\boldsymbol{\gamma}|\mathbf{x}^* &\sim \mathsf{MVN}\left(\begin{pmatrix}-\mathbf{x}^{*T}\boldsymbol{\mu}_{\boldsymbol{\gamma}_1}\\ \boldsymbol{\mu}_{\boldsymbol{\gamma}_1}\end{pmatrix}, \begin{pmatrix}\sigma_\epsilon^2 + \mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1}\mathbf{x}^* & -\mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1}\\ -\mathbf{x}^{*T}\Sigma_{\boldsymbol{\gamma}_1} & \Sigma_{\boldsymbol{\gamma}_1}\end{pmatrix}\right) \\
&\sim \mathsf{MVN}\left(\boldsymbol{\mu}_{\boldsymbol{\gamma}|\mathbf{x}^*}, \Sigma_{\boldsymbol{\gamma}|\mathbf{x}^*}\right).
\end{aligned}$$

Finally,

$$q(\boldsymbol{\gamma}) = \sum_{i=i_1}^{i_{n^*}} \frac{1}{n} \times \phi_{\boldsymbol{\mu}_{\boldsymbol{\gamma}|\mathbf{x}_i}, \Sigma_{\boldsymbol{\gamma}|\mathbf{x}_i}}(\boldsymbol{\gamma}), \tag{C.1}$$

where $\phi_{\boldsymbol{\mu},\Sigma}(\cdot)$ is the multivariate Gaussian density function with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix Σ.

Appendix D

Posterior Means for Parameters of Mixture of Two Gaussian Experts Example from Section 5.4

	Lower	*Estimate*	*Upper*	*Lower*	*Estimate*	*Upper*
		γ_0			γ_1	
Sampler 1	-14.186	-10.416	-6.948	1.451	2.111	2.900
Sampler 2	-13.983	-10.334	-7.05	1.492	2.100	2.853
Sampler 3	-14.102	-10.480	-7.021	1.467	2.126	2.844

TABLE D.1: Posterior means and 95% credible intervals for the gating parameters of the mixture of two Gaussian experts (Section 5.4, data set (i) from Figure 5.1).

	Lower	*Estimate*	*Upper*	*Lower*	*Estimate*	*Upper*
		$\sigma^2_{E^*}$			$\sigma^2_{E^{**}}$	
Sampler 1	0.129	0.198	0.304	0.205	0.309	0.454
Sampler 2	0.129	0.193	0.285	0.233	0.303	0.406

TABLE D.2: Posterior means and 95% credible intervals for the expert variance parameters of the mixture of two Gaussian experts (Section 5.4, data set (i) from Figure 5.1).

	Lower	*Estimate*	*Upper*	*Lower*	*Estimate*	*Upper*
		β_{0E^*}			$\beta_{0E^{**}}$	
Sampler 1	-2.988	-2.011	-1.098	22.541	24.529	26.578
Sampler 2	-2.485	-2.024	-1.611	23.790	24.759	25.716
		β_{1E^*}			$\beta_{1E^{**}}$	
Sampler 1	2.173	2.445	2.735	-3.178	-2.881	-2.588
Sampler 2	2.317	2.449	2.583	-3.056	-2.913	-2.774

TABLE D.3: Posterior means and 95% credible intervals for the slope and intercept parameters of the mixture of two Gaussian experts (Section 5.4, data set (i) from Figure 5.1).

	Lower	*Estimate*	*Upper*	*Lower*	*Estimate*	*Upper*
		γ_0			γ_1	
Sampler 1	-3.226	-1.879	-0.368	0.050	0.302	0.593
Sampler 2	-3.346	-1.747	-0.331	0.034	0.273	0.569
Sampler 3	-2.927	-1.700	-0.299	0.050	0.249	0.500

TABLE D.4: Posterior means and 95% credible intervals for the gating parameters of the mixture of two Gaussian experts (Section 5.4, data set (ii) from Figure 5.1).

	Lower	*Estimate*	*Upper*	*Lower*	*Estimate*	*Upper*
		$\sigma^2_{E^*}$			$\sigma^2_{E^{**}}$	
Sampler 1	0.125	0.192	0.292	0.320	0.527	0.830
Sampler 2	0.117	0.183	0.274	0.332	0.531	0.827

TABLE D.5: Posterior means and 95% credible intervals for the expert variance parameters of the mixture of two Gaussian experts (Section 5.4, data set (i) from Figure 5.1).

	Lower	*Estimate*	*Upper*	*Lower*	*Estimate*	*Upper*
		β_{0E^*}			$\beta_{0E^{**}}$	
Sampler 1	7.610	8.243	8.869	-2.965	-1.826	-0.734
Sampler 2	7.982	8.246	8.533	-2.631	-1.810	-1.047
		β_{1E^*}			$\beta_{1E^{**}}$	
Sampler 1	-0.171	-0.047	0.079	0.904	1.074	1.250
Sampler 2	-0.106	-0.047	0.005	0.949	1.071	1.199

TABLE D.6: Posterior means and 95% credible intervals for the slope and intercept parameters of the mixture of two Gaussian experts (Section 5.4, data set (ii) from Figure 5.1).

Appendix E

Prediction Intervals for HME and BART

HME

For HME, the prediction interval for the i-th point is defined by the 2.5th and 97.5th percentiles of the distribution resulting from all predictions $\hat{\mathbf{y}}_i^* = \left(\hat{y}_i^{*(1)}, ..., \hat{y}_i^{*(T)}\right)$ produced across iterations $t = 1, ..., T$ as follows:

1. Pick expert E^* from all $E \in \mathcal{E}$ with probabilities corresponding to the path probabilities $\left(\pi_i^{(E)(t)}\right)_{E \in \mathcal{E}}$.
2. Draw $u \sim N(0, 1)$.
3. Calculate $\hat{y}_i^{*(t)} = \hat{y}_i^{(t)} + u \times \hat{\sigma}_{E^*}^{(t)}$, where $\hat{y}_i^{(t)}$ denotes the estimated value for y_i at iteration t and $\hat{\sigma}_{E^*}^{(t)}$ corresponds to the posterior standard deviation parameter for the normal expert E^* at iteration t.

The above steps are computed separately for each observation i yielding $\hat{\mathbf{y}}_1^*, ..., \hat{\mathbf{y}}_n^*$, where each $\hat{\mathbf{y}}_i^*$ is used to obtain the corresponding 2.5th and 97.5th percentiles for $i = 1, ..., n$.

BART

For BART, the prediction interval for the i-th point is defined by the 2.5th and 97.5th percentiles of the distribution resulting from all predictions $\hat{\mathbf{y}}_i^* = \left(\hat{y}_i^{*(1)}, ..., \hat{y}_i^{*(T)}\right)$ produced across iterations $t = 1, ..., T$ as follows:

1. Draw $u \sim N(0, 1)$.

2. Calculate $\hat{y}_i^{*(t)} = \hat{y}_i^{(t)} + u \times \hat{\sigma}^{(t)}$, where $\hat{y}_i^{(t)}$denotes the estimated value for y_i at iteration t and $\hat{\sigma}^{(t)}$ corresponds to the posterior standard deviation parameter at iteration t.

The above steps are computed separately for each observation i yielding $\hat{\mathbf{y}}_1^*, ..., \hat{\mathbf{y}}_n^*$, where each $\hat{\mathbf{y}}_i^*$ is used to obtain the corresponding 2.5th and 97.5th percentiles for $i = 1, ..., n$.

GAM

Since GAM is a frequentist approach, the prediction intervals are less straightforward to obtain. The step-by-step approach outlined by Andersen (2019) is thus used.

The corresponding intervals are then obtained as the 2.5th and 97.5th percentiles of the resulting predictions distributions.

Appendix F

Details of GAM fit for Motorcycle Accident Data

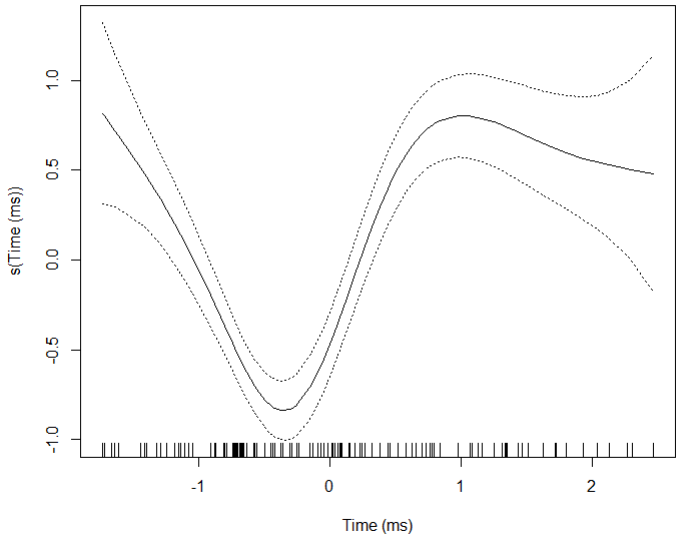


FIGURE F.1: GAM model fit on the motorcycle accident data. The smoothing spline fitted to the explanatory variable (time) is shown as a solid black line alongside the estimated standard errors around the fit shown as dashed lines.

Bibliography

R. P. Adams, Z. Ghahramani, and M. I Jordan. Tree-structured stick breaking processes for hierarchical data. *arXiv preprint arXiv:1006.1062*, 2010.

Admiral. UK supply vs demand analysis for rental requirements., 2022. URL https://www.admiral.com/home-insurance/rental-requirements#supply-vs-demand.

F. Al-Awadhi, M. Hurn, and C. Jennison. Improving the acceptance rate of Reversible Jump MCMC proposals. *Statistics & Probability Letters*, 69:189–198, 08 2004. doi: 10.1016/j.spl.2004.06.025.

M. Aleardi and A. Salusti. Application of Reversible Jump Markov Chain Monte Carlo algorithms to elastic and petrophysical amplitude-versus-angle inversions. *Pure and Applied Geophysics*, 07 2020. doi: 10.1007/s00024-020-02436-w.

M. M. Andersen. Prediction intervals for Generalized Additive Models (GAMs), Aug 2019. URL https://mikl.dk/post/2019-prediction-intervals-for-gam/.

S. Banerjee. Bayesian linear model : Gory details. 2008. URL http://www.biostat.umn.edu/~ph7440/pubh7440/BayesianLinearModelGoryDetails.pdf.

T. Bayes and N. Price. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763. doi: 10.1098/rstl.1763.0053. URL https://royalsocietypublishing.org/doi/abs/10.1098/rstl.1763.0053.

T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young. mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29, 2009. URL http://www.jstatsoft.org/v32/i06/.

P. J. Bickel and E. L. Lehmann. *Frequentist Interpretation of Probability*, pages 1083–1085. Springer US, Boston, MA, 2012. ISBN 978-1-4614-1412-4. doi: 10.1007/978-1-4614-1412-4_92. URL https://doi.org/10.1007/978-1-4614-1412-4_92.

C. Bishop and M. Svensen. Bayesian Hierarchical Mixtures of Experts. 108, 10 2012.

C. M. Bishop and M. Svenskn. Bayesian Hierarchical Mixtures of Experts. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, page 57–64, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 0127056645.

C. Blundell, Y. W. Teh, and K. A. Heller. Bayesian rose trees. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 65–72, Arlington, Virginia, USA, 2010. AUAI Press. ISBN 9780974903965.

L. Breiman. *Classification and regression trees*. Wadsworth International Group, Belmont, Calif, 1984. ISBN 9780534980535;0534980546;9780534980542;0534980538;.

L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, aug 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350. URL https://doi.org/10.1023/A:1018054314350.

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

D. Brigo and F. Mercurio. Lognormal-mixture dynamics and calibration to market volatility smiles. *International Journal of Theoretical and Applied Finance*, 05(04): 427–446, 2002. doi: 10.1142/S0219024902001511. URL https://doi.org/10.1142/S0219024902001511.

S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998. ISSN 10618600. URL http://www.jstor.org/stable/1390675.

S. P. Brooks, P. Giudici, and G. O. Roberts. Efficient construction of Reversible Jump Markov Chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):3–39, 2003. doi: https://doi.org/10.1111/1467-9868.03711. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.03711.

J. Brownlee. A gentle introduction to Markov Chain Monte Carlo for probability, 2019. URL https://machinelearningmastery.com/Markov-chain-monte-carlo-for-probability.

A. Buja and W. Stuetzle. Observations on bagging. *Statistica Sinica*, 16(2):323–351, 2006. ISSN 10170405, 19968507. URL http://www.jstor.org/stable/24307547.

A. Buja, T. Hastie, and R. Tibshirani. Linear Smoothers and Additive Models. *The Annals of Statistics*, 17(2):453 – 510, 1989. doi: 10.1214/aos/1176347115. URL https://doi.org/10.1214/aos/1176347115.

S. v. Buuren. *Flexible imputation of missing data*. CRC Press, London;Boca Raton, FL;, 2012. ISBN 9781439868249;1439868247;.

P. Bühlmann and B. Yu. Analyzing bagging. *Annals of Statistics*, 30, 08 2002. doi: 10.1214/aos/1031689014.

F. Chamroukhi. Non-normal mixtures of experts, 2015. ArXiv preprints 1506.06707, June.

C. Chang, S. Tan, B. Lengerich, A. Goldenberg, and R. Caruana. How interpretable and trustworthy are GAMs?, 2020. URL https://arxiv.org/abs/2006.06466.

L. Chen, M. Cheng, and L. Peng. Conditional variance estimation in heteroscedastic regression models. *Journal of statistical planning and inference*, 139(2):236–245, 2009.

S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995. ISSN 00031305. URL http://www.jstor.org/stable/2684568.

H. Chipman and R. McCulloch. Bayesian Additive Regression Trees. package 'bayestree', 2016. URL https://cran.r-project.org/web/packages/BayesTree/BayesTree.pdf.

H. Chipman, E. George, J. Lemp, and R. McCulloch. Bayesian flexible modeling of trip durations. *Transportation Research Part B: Methodological*, 44:686–698, 06 2010a. doi: 10.1016/j.trb.2010.01.007.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998. ISSN 01621459. URL http://www.jstor.org/stable/2669832.

H. A. Chipman, E. I. George, and R. E. McCulloch. BART: Bayesian Additive Regression Trees. *The Annals of Applied Statistics*, 4(1), mar 2010b. doi: 10.1214/09-aoas285. URL https://doi.org/10.1214%2F09-aoas285.

J. Cook. Effective sample size for MCMC, Jun 2017. URL https://www.johndcook.com/blog/2017/06/27/effective-sample-size-for-{{MCMC}}/.

J. C. Davis. *Statistics and data analysis in geology*. John Wiley & Sons, Chichester;New York, N.Y;, 3rd edition, 2002. ISBN 9780471172758;0471172758;.

C. M. Dayton and G. B. Macready. Concomitant-variable latent-class models. *Journal of the American Statistical Association*, 83(401):173–178, 1998.

H. Demirtas. Flexible imputation of missing data. *Journal of statistical software*, 85 (Book Review 4):1–5, 2018.

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)*, 39(1):1–22, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

T. Denceaux. Splines and Generalized Additive Models. .computational statistics, 2016. URL https://www.hds.utc.fr/~tdenoeux/dokuwiki/_media/en/splines.pdf.

W. S. DeSarbo and W. L. Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, 5:248–282, 1988.

P. Diaconis. The Markov Chain Monte Carlo revolution. *Bulletin of the American Mathematical Society*, 46:179textendash205, 04 2009. doi: 10.1090/S0273-0979-08-01238-X.

J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(2): 363–375, 1994. ISSN 00359246. URL http://www.jstor.org/stable/2345907.

J. Ding, A. Bashashati, A. Roth, A. Oloumi, K. Tse, T. Zeng, G. Haffari, M. Hirst, M. Marra, A. Condon, S. Aparicio, and S. Shah. Feature based classifiers for somatic mutation detection in tumour-normal paired sequencing data. *Bioinformatics (Oxford, England)*, 28:167–75, 11 2011. doi: 10.1093/bioinformatics/btr629.

C. Dutang. Some explanations about the IWLS algorithm to fit generalized linear models. working paper or preprint, August 2017. URL https://hal.archives-ouvertes.fr/hal-01577698.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

R. S. Ehlers and S. P. P. Brooks. Adaptive proposal construction for Reversible Jump MCMC. *Scandinavian Journal of Statistics*, 35(4):677–690, 2008. ISSN 03036898, 14679469. URL http://www.jstor.org/stable/41000296.

ESPC. What was the scottish rental market like in q1 2022?, 2022a. URL https://espc.com/news/post/what-was-the-scottish-rental-market-like-in-q1-2022.

ESPC. Glasgow market rent summary, 2022b. URL https://www.home.co.uk/for_rent/glasgow/current_rents?county=glasgow.

W. M. Farr, I. Mandel, and D. Stevens. An efficient interpolation technique for jump proposals in reversible-jump Markov Chain Monte Carlo calculations. *Royal Society Open Science*, 2(6):150030, Jun 2015. ISSN 2054-5703. doi: 10.1098/rsos.150030. URL http://dx.doi.org/10.1098/rsos.150030.

A. Feelders. Handling missing data in trees: Surrogate splits or statistical imputation? 03 2000. ISBN 978-3-540-66490-1. doi: 10.1007/978-3-540-48247-5_38.

R. A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925. doi: 10.1017/S0305004100009580.

J. Fox and S.F Weisberg. *Appendix: Nonparametric Regression in R" (PDF)*. SAGE, 2018. ISBN 978-1-5443-3645-9.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL https://www.sciencedirect.com/science/article/pii/S002200009791504X.

M. Frigo, C.E. Leiserson, H. Prokop, and S. Ramachandran. Cache-oblivious algorithms. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 285–297, 1999. doi: 10.1109/SFFCS.1999.814600.

J. Fritsch, M. Finke, and A. Waibel. Adaptively growing hierarchicalmixtures of experts. *Advances in Neural Information Processing Systems*, 9:459–465, 1996.

S. Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models*. New York:Springer-Verlag, 2006. ISBN 978-0-387-35768-3.

S. Frühwirth-Schnatter and S. Kaufmann. Model-based clustering of multiple time series. *Journal of Business & Economic Statistics*, 26(1):78–89, 2008. doi: 10.1198/073500107000000106. URL https://doi.org/10.1198/073500107000000106.

S. Frühwirth-Schnatter, C. Pamminger, A.Weber, and R. Winter-Ebmer. Labor market entry and earnings dynamics: Bayesian inference using mixtures-of-experts Markov chain clustering. *Journal of Applied Econometrics*, 27(7):1116–1137, 2012. ISSN 08837252, 10991255. URL http://www.jstor.org/stable/23355927.

A. E. Gelfand. Gibbs sampling. *Journal of the American Statistical Association*, 95(452): 1300–1304, 2000. ISSN 01621459. URL http://www.jstor.org/stable/2669775.

A. Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, 1(3), 2006.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992. ISSN 08834237. URL http://www.jstor.org/stable/2246093.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE transactions on pattern analysis and machine intelligence*,

6(6):721—741, June 1984. ISSN 0162-8828. doi: 10.1109/tpami.1984.4767596. URL https://doi.org/10.1109/tpami.1984.4767596.

N. Gershenfeld. Nonlinear inference and cluster-weighted modeling. *Annals of the New York Academy of Sciences*, 808(1):18–24, 1997.

C. J. Geyer. 1 introduction to Markov Chain Monte Carlo. 2011.

I. S. Gormley and S. Frühwirth-Schnatter. Mixtures of experts models, 2018.

I. S. Gormley and T. B. Murphy. A mixture of experts model for rank data with applications in election studies. *Annals of Applied Statistics*, 2(4):1452–1477, 12 2008. doi: 10.1214/08-AOAS178. URL https://doi.org/10.1214/08-AOAS178.

I. S. Gormley and T. B Murphy. Clustering ranked preference data using sociodemographic covariates. In S. Hess and A. Daly, editors, *Choice Modelling: The State-of-the-Art and the State-of-Practice*, pages 543–569. 2010a. United Kingdom: Emerald.

I. S. Gormley and T. B Murphy. A mixture of experts latent position cluster model for social network data. *Statistical Methodology*, 7(3):385–405, 2010b. doi: 10.1016/j.stamet.2010.01.002.

R. B. Gramacy. PhD Thesis. Bayesian Treed Gaussian PROCESS MODELS, publisher = UNIVERSITY OF CALIFORNIA,, 2015.

P. Green and D. Hastie. Reversible Jump MCMC. *Genetics*, 155, 01 2009.

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)*, 46(2):149–170, 1984. doi: 10.1111/j.2517-6161.1984.tb01288.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1984.tb01288.x.

P. J. Green. Reversible Jump Markov Chain Monte Carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995. ISSN 00063444. URL http://www.jstor.org/stable/2337340.

P. J. Green. *Trans-dimensional Markov Chain Monte Carlo (Chapter 6)*, pages 179 – 198. Oxford University Press, United Kingdom, 2003.

P. J. Green and A. Mira. Delayed rejection in Reversible Jump metropolis-hastings. *Biometrika*, 88(4):1035–1053, 2001. ISSN 00063444. URL http://www.jstor.org/stable/2673700.

P. J. Green and B. W. Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*, volume 58. Chapman & Hall, London;London, Chapman & Hall, 1994;, 1994. ISBN 0412300400;9780412300400;.

D. I. Hastie and Peter J. Green. Model choice using Reversible Jump Markov Chain Monte Carlo. *Statistica Neerlandica*, 66(3):309–338, 2012. doi: https://doi.org/10.1111/j.1467-9574.2012.00516.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9574.2012.00516.x.

T. Hastie and R. Tibshirani. Generalized Additive Models. *Statistical Science*, 1(3): 297 – 310, 1986. doi: 10.1214/ss/1177013604. URL https://doi.org/10.1214/ss/1177013604.

T. Hastie and R. Tibshirani. Bayesian backfitting (with comments and a rejoinder by the authors. *Statistical Science*, 15, 08 2000. doi: 10.1214/ss/1009212815.

T. J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1990. ISBN 9780412343902. URL https://books.google.lt/books?id=qa29r1Ze1coC.

W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. URL http://www.jstor.org/stable/2334940.

K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 297–304, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102389. URL https://doi.org/10.1145/1102351.1102389.

N. E. Helwig. Smoothing spline regression in r, 2021. URL http://users.stat.umn.edu/~helwig/notes/smooth-spline-notes.html.

B. Hernandez, S. Pennington, and A. Parnell. Bayesian methods for proteomic biomarker development. *EuPA Open Proteomics*, 9, 08 2015. doi: 10.1016/j.euprot.2015.08.001.

M. Holmes-Cerfon. Lecture 2: Markov chains (i), 2022. URL https://cims.nyu.edu/~holmes/teaching/asa22/handout-Lecture2_2022.pdf.

Home.co.uk. Current house prices in glasgow, 2022. URL https://www.home.co.uk/guides/house_prices.htm?location=glasgow.

D. Hsu. Tests for variance shift at an unknown time point. *Journal of The Royal Statistical Society Series C-applied Statistics*, 26:279–284, 1977.

Y. Hu, J. L. Ying, H. Daume III, and Z. I. Ying. Binary to bushy: Bayesian hierarchical clustering with the beta coalescent. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 1079–1087. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/e5f6ad6ce374177eef023bf5d0c018b6-Paper.pdf.

G. Huerta, W. Jiang, and M. A. Tanner. Time series modeling via hierarchical mixtures. *Statistica Sinica*, 13(4):1097–1118, 2003. ISSN 10170405, 19968507. URL http://www.jstor.org/stable/24307162.

D. R. Hunter and D. S. Young. Semiparametric mixtures of regressions. *Journal of Nonparametric Statistics*, 24(1):19–38, 2012. doi: 10.1080/10485252.2011.608430. URL https://doi.org/10.1080/10485252.2011.608430.

M. Hurn, A. Justel, and C. P Robert. Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1):55–79, 2003. doi: 10.1198/1061860031329. URL https://doi.org/10.1198/1061860031329.

S. Ingrassia, A. Punzo, G. Vittadini, and S. C. Minotti. The generalized linear mixed cluster-weighted model. *Journal of Classification*, 32(1):85–113, 2016.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

A. Jasra, D. A. Stephens, and C. C. Holmes. Population-Based Reversible Jump Markov Chain Monte Carlo. *Biometrika*, 94(4):787–807, 11 2007. ISSN 0006-3444. doi: 10.1093/biomet/asm069. URL https://doi.org/10.1093/biomet/asm069.

W. Jiang and M. A. Tanner. On the approximation rate of hierarchical mixtures-of-experts for generalized linear models. *Neural Computation*, 11(5):1183–1198, July 1999a. ISSN 0899-7667. doi: 10.1162/089976699300016403.

W. Jiang and M. A. Tanner. On the identifiability of mixtures-of-experts. *Neural networks : the official journal of the International Neural Network Society*, 12(9):1253—1258, November 1999b. ISSN 0893-6080. doi: 10.1016/s0893-6080(99)00066-0. URL https://doi.org/10.1016/s0893-6080(99)00066-0.

W. Jiang and M. A. Tanner. On the asymptotic normality of hierarchical mixtures-of-experts for generalized linear models. *IEEE Transactions on Information Theory*, 46(3):1005–1013, May 2000. ISSN 0018-9448. doi: 10.1109/18.841177.

W. Jiang and M. A. Tanner. Hierarchical mixtures-of-experts for generalized linear models: some results on denseness and consistency. volume 1 of *Proceedings of Machine*

Learning Research, Fort Lauderdale, FL, USA, 03–06 Jan 2020. PMLR. URL http://proceedings.mlr.press/r1/jiang20a.html.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(1):181–214, January 1994.

M. I. Jordan and L. Xu. Convergence results for the em approach to mixtures of experts architectures. *Neural Networks*, 8(9):1409–1431, 1995. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(95)00014-3. URL http://www.sciencedirect.com/science/article/pii/0893608095000143.

D. Karlis and E. Xekalaki. Choosing initial values for the em algorithm for finite mixtures. In *Comput. Stat. Data Anal.*, 2003.

J. M. Keith, D. P. Kroese, and D. Bryant. A generalized Markov sampler. 6:29 – 53, 2004. doi: 10.1023/B:MCAP.0000012414.14405.15. URL https://doi.org/10.1023/B:MCAP.0000012414.14405.15.

C. D. Kemp. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 151 (3):565–567, 1988. ISSN 09641998, 1467985X. URL http://www.jstor.org/stable/2983026.

M. G. Kendall. ON THE RECONCILIATION OF THEORIES OF PROBABILITY. *Biometrika*, 36(1-2):101–116, 06 1949. ISSN 0006-3444. doi: 10.1093/biomet/36.1-2.101. URL https://doi.org/10.1093/biomet/36.1-2.101.

M. Kubsch, I. Stamer, M. Steiner, K. Neumann, and I. Parchmann. Beyond p-values: Using bayesian data analysis in science education research. page 2021, 02 2021.

M. Lai. Course handouts for bayesian data analysis class, 2019. URL https://bookdown.org/marklhc/notes_bookdown/.

K. Larsen. GAM : The predictive modeling silver bullet. 2015.

A. Lawrynowicz and V. Tresp. *Introducing Machine Learning*, volume 18, pages 35–50. 01 2014.

F. Li, M. Villani, and R. Kohn. Modeling conditional densities using finite smooth mixtures. In C. P. Robert Edited by K. L. Mengersen and D. M. Titterington, editors, *Mixtures: Estimation and Applications*, chapter 9, pages 123–144. First edition, 2011. doi: 10.1002/9781119995678.ch6.

A. R. Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636, 2018. doi: 10.1080/01621459.2016.1264957. URL https://doi.org/10.1080/01621459.2016.1264957.

A. R. Linero and Y. Yang. Bayesian regression tree ensembles that adapt to smoothness and sparsity. 2017. doi: 10.48550/ARXIV.1707.09461. URL https://arxiv.org/abs/1707.09461.

J. S. Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427): 958–966, 1994. ISSN 01621459. URL http://www.jstor.org/stable/2290921.

Z. Liu and Z. Zhang. Quantum-inspired hamiltonian monte carlo for bayesian sampling, 2020.

W. Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014. doi: 10.1111/insr.12016. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12016.

D. Lunn. *The BUGS book: a practical introduction to Bayesian analysis*. Chapman & Hall/CRC, London;Boca Raton, Fla;, 2013. ISBN 1584888490;9781584888499;.

S. Masoudnia and R. Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, second edition, 1989. ISBN 978-0412317606.

G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, Inc, 2000. ISBN 978-0-471-00626-8.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley, 2004. ISBN 9780471654063. URL https://books.google.co.uk/books?id=c2_fAox0DQoC.

X. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 06 1993. ISSN 0006-3444. doi: 10.1093/biomet/80.2.267. URL https://doi.org/10.1093/biomet/80.2.267.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

D. L. Miller. Bayesian views of Generalized Additive Modelling, 2019. URL https://arxiv.org/abs/1902.01330.

K. Murphy and T. B. Murphy. Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, 14(2):

293–325, 2020a. doi: 10.1007/s11634-019-00373-8. URL https://doi.org/10.1007/s11634-019-00373-8.

K. Murphy and T. B. Murphy. `MoEClust`*: Gaussian Parsimonious Clustering Models with Covariates and a Noise Component*, 2020b. URL https://cran.r-project.org/package=MoEClust. R package version 1.3.3.

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9 (1):141–142, 1964. doi: 10.1137/1109020. URL https://doi.org/10.1137/1109020.

R. M. Neal. MCMC using hamiltonian dynamics, 2012.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972. ISSN 00359238. URL http://www.jstor.org/stable/2344614.

C. R. Nelson and H. Kang. Pitfalls in the use of time as an explanatory variable in regression. *Journal of business and economic statistics*, 2(1):73–82, 1984.

D. P. Nguyen, L. M. Frank, and E. N. Brown. An application of reversible-jump Markov Chain Monte Carlo to spike classification of multi-unit extracellular recordings. *Network (Bristol, England)*, 14(1):61—82, February 2003. ISSN 0954-898X. doi: 10.1088/0954-898x/14/1/304. URL https://doi.org/10.1088/0954-898x/14/1/304.

N. Noam Ross. Plotting and interpreting GAM interactions, 2022. URL https://campus.datacamp.com/courses/nonlinear-modeling-with-generalized-additive-models-gams-in-r/spatial-gams-and-interactions?ex=4.

J. D. Opsomer and D. Ruppert. Fitting a bivariate additive model by local polynomial regression. *The Annals of Statistics*, 25(1):186 – 211, 1997. doi: 10.1214/aos/1034276626. URL https://doi.org/10.1214/aos/1034276626.

H. Park and T. Ozeki. Singularity and slow convergence of the em algorithm for gaussian mixtures. *Neural Processing Letters*, 29:45–59, 02 2009. doi: 10.1007/s11063-009-9094-4.

K. Pearson. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 186:343–414, 1894.

F. Peng, R. A. Jacobs, and M. A. Tanner. Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition. *Journal of the American Statistical Association*, 91(435):953–960, September 1996. ISSN 0162-1459. doi: 10.1080/01621459.1996.10476965.

H. Permuter, J. Francos, and I. Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2005.10.028. URL https://www.sciencedirect.com/science/article/pii/S0031320305004334. Graph-based Representations.

A. Perperoglou, W. Sauerbrei, M. Abrahamowicz, and M. Schmid. A review of spline function procedures in r. *BMC Medical Research Methodology*, 19, 03 2019. doi: 10.1186/s12874-019-0666-3.

E. B. Prado, R. A. Moral, and A. C. Parnell. Bayesian Additive Regression Trees with model trees. *Statistics and computing*, 31(3), 2021.

M. T. Pratola. Efficient metropolis–hastings proposal mechanisms for bayesian regression tree models. *Bayesian analysis*, 11(3), 2016.

M. T. Pratola, H. A. Chipman, E. I. George, and R. E. McCulloch. Heteroscedastic BART via multiplicative regression trees. *Journal of computational and graphical statistics*, 29(2):405–417, 2020.

R. E. Quandt. A new approach to estimating switching regressions. *Journal of the American Statistical Association*, 67(338):306–310, 1972.

R Documentation. BART: Bayesian Additive Regression Trees, 2022. URL https://www.rdocumentation.org/packages/d{{BART}}s/versions/0.9-22/topics/{{BART}}.

R Documentation. Specify a Smoothing Spline Fit in a GAM Formula, 2023. URL https://search.r-project.org/CRAN/refmans/gam/html/gam.s.html.

C. E. Rasmussen and Z. Ghahramani. The infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 12:554–560, 2002. MIT Press.

D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995. doi: 10.1109/89.365379.

B. D. Ripley. Stochastic simulation (wiley series in probability and statistics). 1987. ISBN 0471818844.

P. Rosopa, M. Schaffer, and A. Schroeder. Managing heteroscedasticity in general linear models. *Psychological methods*, 18:335–51, 09 2013. doi: 10.1037/a0032553.

RW Invest. Property investment glasgow: Your complete 2022 guide, 2022. URL https://www.rw-invest.com/glasgow-property-investment/.

M. Sammel, X. Lin, and L. Ryan. Multivariate linear mixed models for multiple outcomes. *Statistics in medicine*, 18(17-18):2479–2492, 1999.

F. Sapundzhi, T. Dzimbova, N. Pencheva, and P. Milanov. Determination of the relationship between the docking studies and the biological activity of δ-selective enkephalin analogues. 5:98–108, 07 2015.

SAS Help Center. Assessing Markov chain convergence, 2019. URL https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_introbayes_sect025.htm#:~:text=Trace%20plots%20of%20samples%20versus,the%20chain%20is%20mixing%20well.

G. Schmidt, R. Mattern, and F. Schuler. Biomechanical investigation to determine physical and traumatological differentiation criteria for the maximum load capacity of head and vertebral column with and without protective helmet under the effects of impact. *Research Program on Biomechanics of Impacts*, (Final report Phase III): Project G5, 1981.

Scotland's Census. Scotland's census housing report, 2011. URL https://www.scotlandscensus.gov.uk/census-results/at-a-glance/housing/.

S. A. Shaban. Change point problem and two-phase regression: an annotated bibliography. *International statistical review*, 48(1):83, 1980.

J. Sherman and W. J. Morrison. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 21(1):124 – 127, 1950. doi: 10.1214/aoms/1177729893. URL https://doi.org/10.1214/aoms/1177729893.

B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52, 1985. ISSN 00359246. URL http://www.jstor.org/stable/2345542.

S. A. Sisson. Transdimensional markov chains. *Journal of the American Statistical Association*, 100(471):1077–1089, 2005. doi: 10.1198/016214505000000664. URL https://doi.org/10.1198/016214505000000664.

D. R. Smith. The design of divide and conquer algorithms. *Science of Computer Programming*, 5:37–58, 1985. ISSN 0167-6423. doi: https://doi.org/10.1016/0167-6423(85)90003-6. URL https://www.sciencedirect.com/science/article/pii/0167642385900036.

A. Smola. *Introduction to machine learning*. The press syndicate of the university of Cambridge, 2008. ISBN 0521825830.

C. Spearman. "General intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904. ISSN 00029556. URL http://www.jstor.org/stable/1412107.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974. ISSN 00359246. URL http://www.jstor.org/stable/2984809.

Study.eu. Study in glasgow, 2022. URL https://www.study.eu/city/glasgow.

S. Subed, A. Punzo, S. Ingrassia, and P. D. McNicholas. Clustering and classification via cluster-weighted factor analyzers. *Advances in Data Analysis and Classification*, 7 (1):5–40, 2013.

Y. Tan and J. Roy. Bayesian Additive Regression Trees and the general BART model. *Statistics in Medicine*, 38, 08 2019. doi: 10.1002/sim.8347.

B. Tang, M. I. Heywood, and M. Shepherd. Input partitioning to mixture of experts. *IEEE Transactions on Knowledge and Data Engineering*, page 227–23, 2002.

X. Tang and A. Qu. Mixture modeling for longitudinal data. *Journal of Computational and Graphical Statistics*, 25(4):1117–1137, 2016. doi: 10.1080/10618600.2015.1092979. URL https://doi.org/10.1080/10618600.2015.1092979.

N. Tawara, T. Ogawa, S. Watanabe, A. Nakamura, and T. Kobayashi. Blocked gibbs sampling based multi-scale mixture model for speaker clustering on noisy data. 09 2013. doi: 10.1109/MLSP.2013.6661902.

A. Teixeira-Pinto, J. Siddique, R. Gibbons, and S. Normand. Statistical approaches to modeling multiple outcomes in psychiatric studies. *Psychiatric annals*, 39:729–735, 07 2009. doi: 10.3928/00485713-20090625-08.

D. M. Titterington, P. S. D.M. Titterington, S.A.F. M, A.F.M. Smith, U.E. Makov, and John Wiley & Sons. *Statistical Analysis of Finite Mixture Distributions*. Applied section. Wiley, 1985. ISBN 9780471907633. URL https://books.google.co.uk/books?id=hZOQAQAAIAAJ.

N. Toft, G. T. Innocent, G. Gettinby, Stuart W.J., and S. W. J. Reid. Assessing the convergence of Markov Chain Monte Carlo methods: An example from evaluation of diagnostic tests in absence of a gold standard. *Preventive Veterinary Medicine*, 79(2):244–256, 2007. ISSN 0167-5877. doi: https://doi.org/10.1016/j.

prevetmed.2007.01.003. URL https://www.sciencedirect.com/science/article/pii/S0167587707000037.

B. M. Turner, P. B. Sederberg, S. D. Brown, and M. Steyvers. A method for efficiently sampling from distributions with correlated dimensions, Sep 2013.

J. Vallverdu. *Bayesian Versus Frequentist Statistical Reasoning*. 01 2011. doi: 10.1007/978-3-642-04898-2_2.

D. van Ravenzwaaij, P. Casey, and S. D. Brown. A simple introduction to Markov Chain Monte-Carlo sampling. *"Psychonomic Bulletin and Review"*, 25(1):143–154, February 2018. ISSN 1069-9384. doi: 10.3758/s13423-016-1015-8.

B. Verity. Ensuring good mixing, 2019. URL https://bobverity.github.io/MALECOT/articles/ensuring-good-mixing.html#references.

B. Vidakovic. *Bayesian Inference Using Gibbs Sampling – BUGS Project*, pages 733–745. Springer New York, New York, NY, 2011. ISBN 978-1-4614-0394-4. doi: 10.1007/978-1-4614-0394-4_19. URL https://doi.org/10.1007/978-1-4614-0394-4_19.

M. Villani, R. Kohn, and P. Giordani. Regression density estimation using smooth adaptive Gaussian mixtures. *Journal of Econometrics*, 153(2):155–173, 2009. ISSN 0304-4076. doi: https://doi.org/10.1016/j.jeconom.2009.05.004. URL http://www.sciencedirect.com/science/article/pii/S0304407609001419.

R. Waagepetersen and D. Sorensen. A tutorial on Reversible Jump MCMC with a view toward applications in qtl-mapping. *International Statistical Review / Revue Internationale de Statistique*, 69(1):49–61, 2001. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403529.

G. Wahba. *Smoothing Splines*, pages 1349–1353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_527. URL https://doi.org/10.1007/978-3-642-04898-2_527.

M. Wang, W. Fu, X. He, S. Hao, and X. Wu. A survey on large-scale machine learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2574–2594, 2022. doi: 10.1109/TKDE.2020.3015777.

P. Wang, M. L. Puterman, I. Cockburn, and N. Le. Mixed poisson regression models with covariate dependent rates. *Biometrics*, 52:381–400, 1996.

Y. Wang, X. Zhou, H. Wang, K. Li, L. Yao, and S. T. C. Wong. Reversible Jump MCMC approach for peak identification for stroke SELDI mass spectrometry using mixture model. *Bioinformatics*, 24(13):i407–i413, 07 2008. ISSN 1367-4803. doi: 10.1093/bioinformatics/btn143. URL https://doi.org/10.1093/bioinformatics/btn143.

L. Wasserman. *All of statistics : a concise course in statistical inference*. Springer, New York, 2010. ISBN 9781441923226 1441923225. URL http://www.amazon.de/All-Statistics-Statistical-Inference-Springer/dp/1441923225/ref=sr_1_2?ie=UTF8&qid=1356099149&sr=8-2.

S. R. Waterhouse and A. J. Robinson. Classification using hierarchical mixtures of experts. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 177–186, 1994. doi: 10.1109/NNSP.1994.366050.

S. R. Waterhouse, D. MacKay, and A. J. Robinson. Bayesian methods for mixtures of experts. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 351–357. MIT Press, 1996. URL http://papers.nips.cc/paper/1167-bayesian-methods-for-mixtures-of-experts.pdf.

D. L. Weller, T. M. T. Love, and M. Wiedmann. Interpretability versus accuracy: A comparison of machine learning models built using different algorithms, performance measures, and features to predict e. coli levels in agricultural water. *Frontiers in Artificial Intelligence*, 4, 2021. ISSN 2624-8212. doi: 10.3389/frai.2021.628441. URL https://www.frontiersin.org/articles/10.3389/frai.2021.628441.

A. White and T. B. Murphy. Mixed-membership of experts stochastic blockmodel. *Network Science*, 4(1):48–80, 2016. doi: 10.1017/nws.2015.29.

S. N. Wood. Fast stable direct fitting and smoothness selection for Generalized Additive Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):495–518, 2008. doi: https://doi.org/10.1111/j.1467-9868.2007.00646.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2007.00646.x.

M.A. Woodbury. *Inverting Modified Matrices*. Memorandum Report / Statistical Research Group, Princeton. Department of Statistics, Princeton University, 1950. URL https://books.google.co.uk/books?id=_zAnzgEACAAJ.

Y. Xia. A note on the backfitting estimation of additive models. *Bernoulli*, 15(4):1148 – 1153, 2009. doi: 10.3150/09-BEJ183. URL https://doi.org/10.3150/09-BEJ183.

S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.

W. Zhao, Y. Gao, S. A. Memon, B. Raj, and R. Singh. Hierarchical routing mixture of experts. *CoRR*, abs/1903.07756, 2019. URL http://arxiv.org/abs/1903.07756.

A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 1st edition, 2018. ISBN 1491953241.

Zoopla Limited. Zoopla limited. economic and social research council. Zoopla property data, 2022, 2022.