Title
Topics in Evidence Synthesis

Author
Pozzi, Luca

Publication Date
2014

Peer reviewed|Thesis/dissertation

Topics in Evidence Synthesis

by

Luca Pozzi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor in Philosophy

in

Biostatistics

and the Designated Emphasis

in

Computational Sciences and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alan E. Hubbard, Chair
Professor Nicholas P. Jewell
Professor John F. Canny

Spring 2014

Topics in Evidence Synthesis

A M E (N)

Contents

List of Figures

List of Tables

Acknowledgements

There are many important people I would like to thank for the support and encouragement they have given me over the years, some of whom may not realize the impact they have had on completing this dissertation.

This dissertation would not have been possible without the help of my advisers Professors Nicholas P. Jewell and Alan E. Hubbard, with whom it has been an honor to work. I also wish to thank the other member of my dissertation committee, Professor John F. Canny, for his insightful advice and for the very stimulating conversations that helped me open my mind to many interesting aspects of Computer Science. A great deal of help came from Robin Mejia's copious advice on writing, traded with more modest advice on coding. I would also like to thank my dear friends Grazia and Graziella Alcasso for equally supporting me while writing this dissertation.

A lot of people were with me constantly during this journey, their contribution has been crucial in inspiring me and keeping me committed and sane. I'll name too few of them, but remember all their names and faces. My parents Laura Caccianotti and Kiki Pozzi who I have to thank for their patience and for planting in me the seed of everything I grew up to be. Vittoria Boella and Giacomo Monti, for being real family and helping that seed grow with their stories, love and such.

Professor Marco Carone for being both a source of intellectual growth, with new and challenging problems, and a precious friend who supported me in endless ways. Marco Garieri, for being a dear friend and colleague, always there to support and encourage me and challenging me to live up to his admiration.

Cora Repetti and Dr. Karen McKeown for their wise advice, constant support and for bringing me back to reality when I needed it the most. Dr. Dan Brown dear friend and colleague, for being the best American Representative an immigrant can hope for. My old time friends that stayed with me through all these years: Gadano, Aldo and Mari coming to the Far West to celebrate with me and to someone who is not and that I am too stubborn to name here.

Michael Patrick Hughes for being my Best Friend in any moment and for all his most precious support and teaching. I owe him more than I can express.

Team FUEGO and Rob Craven for getting me back in the water where I belong after many years. Sharon Norris, because I would have drowned in the paperwork without her neverlasting patience and dedication. Professor Mauro Gasparini and Professor Vincenzo Arnone for making of me the man I am today.

Frida for being and having always been on my mind and in my heart.

Overview

This dissertation treats three different methods to extract information from the etherogeneous data sources. This is a common problem in applications, as motivated by the examples presented in each Chapter. This sources can be different in nature.

In the field of design of experiment is important to integrate external source of evidence as well as data from previous experiments. Each piece of evidence we collect can help us, if appropriately used, to shed light on the phenomenon of interest. This matter is discussed in the first Chapter were a new method to to construct adaptive experiments is presented. Adaptive trial designs can considerably improve upon traditional designs, by modifying design aspects of the ongoing trial, like early stopping, adding, or dropping doses, or changing the sample size. In the present work, we propose a two-stage Bayesian adaptive design for a Phase IIb study aimed at selecting the lowest effective dose for Phase III. In this setting, efcacy has been proved for a high dose in a Phase IIa proof-of-concept study, but the existence of a lower but still effective dose is investigated before the scheduled Phase III starts. In the rst stage, we randomize patients to placebo, maximal tolerated dose, and one or more additional doses within the dose range. Based on an interim analysis, we either stop the study for futility or success or continue the study to the second stage, where newly recruited patients are allocated to placebo, some fairly high dose, and one additional dose chosen based on interim data. At the interim analysis, we use the criteria based on the predictive probability of success to decide on whether to stop or to continue the trial and, in the latter case, which dose to select for the second stage. Finally, we will select a dose as lowest effective dose for Phase III either at the end of the rst stage or at the end of the second stage. We evaluate the operating characteristics of the procedure via simulations and present the results for several scenarios, comparing the performance of the proposed procedure to those of the non-adaptive design.

The context of the second Chapter deals with the common setting in which the outcome of interest is too hard or expensive to measure or too slow to observe. In this case we might be able to measure another variable which can then be used as a surrogate for the outcome of interest. The development of novel therapies in multiple sclerosis (MS) is one area where a range of surrogate outcomes are used in various stages of clinical research. While the aim of treatments in MS is to prevent disability, a clinical trial for evaluating a drugs effect on disability progression would require a large sample of patients with many years of follow-up. The early stage of MS is characterized by relapses. To reduce study size and duration,

clinical relapses are accepted as primary endpoints in phase III trials. For phase II studies, the primary outcomes are typically lesion counts based on Magnetic Resonance Imaging (MRI), as these are considerably more sensitive than clinical measures for detecting MS activity. Recently, Sormani and colleagues in "Surrogate endpoints for EDSS worsening in multiple sclerosis" provided a systematic review, and used weighted regression analyses to examine the role of either MRI lesions or relapses as trial level surrogate outcomes for disability. We build on this work by developing a Bayesian three-level model, accommodating the two surrogates and the disability endpoint, and properly taking into account that treatment effects are estimated with errors. Specifically, a combination of treatment effects based on MRI lesion count outcomes and clinical relapse, both expressed on the log risk ratio scale, were used to develop a study level surrogate outcome model for the corresponding treatment effects based on disability progression. While the primary aim for developing this model was to support decision making in drug development, the proposed model may also be considered for future validation.

The third Chapter deals with the problem of finding interpretable patterns in the massive amount of data collected in fields such as genomics and epidemiology is increasingly becoming a challenge. We present a new method of histogram regression ideally suitable for high dimensional data. Being a form of histogram regression, the algorithm produces partitions of the covariate space that are selected to predict an outcome minimizing some loss function. The partitions are constructed using a clustering algorithm that groups together observations and constructs a hierarchical partition of the space. The hierarchy is then explored to find a tradeoff between prediction accuracy and complexity, as in common tree methods. This results in sometimes relatively interpretable groups of observations defined by having similar values of covariates. The class of learners proposed in this paper extends the class of tree based methods. This procedure is illustrated by simulations from which we can see how this method compares to CART. A feature selection heuristic is also developed to improve the performances and reduce the noise. Software is made available in the R package HOPSLAM (HOpach-Pam Supervised Learning AlgorithM) to make this methodology easily accessible.

Further detail about the algorithms, the implementation of the methods and further results and data are included in the Appendices.

Chapter 1

A Bayesian Adaptive Dose Selection Procedure with an Overdispersed Count Endpoint

1.1 Introduction

In clinical drug development, a sequence of studies is carried out to identify an efficacious and safe dose of the investigational drug. The different phases of clinical development have a focus on either learning or confirming Sheiner (1997), and the development process in a general medicine (non-oncology) setting can be described as follows. Clinical development typically starts in Phase I with studies to find the maximal tolerated dose in healthy volunteers. In Phase IIa, a Proof-of-Concept (PoC) study aims at confirming efficacy in patients, by comparing some fairly high dose against placebo or active comparator (but placebo will be assumed for now on for the sake of simplicity). If the PoC is successful, full development starts with Phase IIb, where several doses of the investigational drug are evaluated, with the aim to select the lowest effective dose for the confirmatory Phase III. In Phase III, the selected dose is then compared against placebo, confirming its efficacy and safety.

We focus here on the design of Phase IIb studies, and propose a two-stage Bayesian adaptive design to select the lowest effective dose for Phase III, using the results of the previous PoC study. A key assumption we will make is that, in our setting, efficacy has been proved for a high dose, but the existence of a lower but still effective dose has to be investigated before the scheduled Phase III starts.

Usual Phase IIb studies are parallel group studies, where patients are randomized to placebo and three to five doses of the investigational drug. These traditional designs do not

make the best use of patients. For example, some of the doses included in the trial may be equivalent to placebo, and it would be better to allocate patients to other doses. As an alternative, adaptive trial designs can considerably improve upon traditional designs, by modifying design aspects of the ongoing trial, such as early stopping, adding or dropping doses, or changing the sample size Bornkamp et al. (2007) Berry et al. (2010), Berry (2006). We consider here a two-stage adaptive design where in the first stage patients are randomized to placebo, some fairly high dose, and one or more additional doses within the dose range. Based on an interim analysis, the study is either stopped for futility, stopped for success, or enters the second stage. In the second stage, newly recruited patients are allocated to placebo, the highest dose considered, and one additional dose chosen on the basis of interim data. At the end of the trial, all available data are used to select the lowest effective dose.

In confirmatory adaptive Phase III trials, regulatory constraints such as strict control of type I error limit the possibility for adaptations Hung et al. (2011), U.S. Food and Drug Administration (2010), CHMP (2007). Hence in confirmatory adaptive trials, frequentist procedures are typically used for the final statistical analysis Schmidli et al. (2006), Bretz et al. (2009), and Bayesian methods are mainly used to support interim decision making Schmidli et al. (2007) Brannath et al. (2009). In adaptive phase IIb trials, Bayesian methods are commonly used for the final evaluation as well Berry et al. (2002) Grieve and Krams (2005). In our two-stage Bayesian adaptive design, we assume a monotone dose-response relationship, use informative priors for the placebo and the highest dose investigated based on the results of the PoC trial, and derive the posterior distribution for the model parameters at interim.

At the interim analysis, criteria based on the predictive distribution are used to decide whether to stop or to continue the trial, and, in the latter case, which dose to select for the second stage.

In all cases, enough evidence about the efficacy of the highest dose of the drug has been accrued in the previous phase IIA proof-of-concept study and a Phase III is certainly going to be carried out. However the scope of the study is to investigate the possibility of sending a better dose to Phase III. If this second Phase IIb does not give a clear result the highest dose will typically be used in Phase III.

In Section 2, the adaptive trial is presented. First, a negative binomial model for the count endpoint is introduced, and the main features of the adaptive design are briefly discussed. Then a semi-parametric model used to describe the monotone dose-response relationship is presented, and the posterior distribution of the model parameters is derived. Finally, the criteria for the Least Effective Dose (LED) are defined, and the adaptation criteria are discussed. In Section 3, the operating characteristics of the adaptive design are evaluated by simulations. The Chapter closes with a brief discussion on implications and possible extensions.

1.2 A Phase II trial with a Count Endpoint

1.2.1 Overdispersed Count Endpoint

In several disease areas, the primary endpoint in phase II trials is a count endpoint. For example, in the development of investigational drugs against multiple sclerosis, the primary endpoint in phase II is the number of lesions Friede and Schmidli (2010a). Other diseases where count endpoints are used include gout (number of flares) Akacha and Benda (2010) epilepsy (number of seizures) Molenberghs et al. (2007) and COPD Friede and Schmidli (2010c).

A Poisson distribution may not fit count endpoints, as the variance of the counts is sometimes substantially larger than the mean. A commonly used model to describe overdispersed count data is the negative binomial model. This model can often adequately describe count data observed in clinical trials (Keene et al. (2007a), van den Elskamp et al. (2009a), Cook et al. (2009)).

The motivating example for this work comes from research in Multiple sclerosis (MS), a chronic autoimmune disease of the central nervous system that causes inflammation and neurodegeneration. About 2.5 million people worldwide are affected by MS World Health Organization (2004), with 50% of patients severely disabled and not ambulatory 20 years after onset.

Phase II trials in MS patients with new investigational drugs are typically randomized placebo-controlled parallel group studies with a duration of around 6 months Friede and Schmidli (2010a). The primary endpoint is the number of lesions in the brain, measured by Magnetic Resonance Imaging (MRI). The distribution of this count endpoint can not be described by a Poisson distribution, as the variance is typically far larger than the mean. For example, in a recent Phase II trial Kappos et al. (2006), the average number of lesions in the placebo group was 15 and the variance was 500. However, a negative binomial distribution can adequately describe the distribution of the number of lesions Sormani et al. (2009) Sormani et al. (2001b) van den Elskamp et al. (2009a) Mercier et al. (2009).

The negative binomial distribution can be defined as a mixture of Poisson distributions, where the means are from a gamma distribution with shape parameter α and inverse scale parameter β. Denoting the number of events by Y, we write $Y \sim \text{dnbinom}(\alpha, \beta)$.

The expected value of Y is $\mu = \alpha/\beta$ and its variance is $\mu \cdot (1 + \mu/\alpha)$. As α and β increase, with μ being constant, the overdispersion factor $(1 + \mu/\alpha)$ goes to 1, and the negative binomial distribution approaches the Poisson distribution.

The negative binomial model will be used in the following to describe the distribution of the number of events for a given dose. Following a Bayesian approach α and β will be given

prior distributions.

1.2.2 Adaptive Trial Design

We consider the design of a phase IIb clinical trial with the aim to select the lowest effective dose (LED) for phase III. Such a clinical trial typically includes a control group (placebo), the maximum tolerated dose, and several intermediate doses. For ease of presentation, we discuss here the situation where three intermediate doses are available, although the methodology can easily be generalized.

There are various ways to define what constitutes the lowest effective dose, and this requires medical expertise. We consider here the lowest effective dose as the dose which is, in terms of mean number of lesions, better than placebo by at least 50%, and at most 20% worse than the highest dose, for which efficacy has already been proven in the previous phase IIa study. The percentages used here were considered to be appropriate for our case, but will need to be changed in other situations. We emphasize again that our discussion will be conducted in general terms, but using specific details of our case study.

Traditionally, dose selection trials use a fixed design, i.e. patients are allocated to the different treatment groups and at the end of the trial a dose is selected. As an alternative, we propose here a two-stage adaptive clinical trial where in the first stage patients are allocated to placebo (dose 1 from now on), the highest dose under investigation (dose 5 from now on), and 1 to 3 intermediate doses (doses 2 to 4). At the interim analysis, a decision is made to either stop the study for success/futility, or to continue. If the trial continues, then additional patients are randomized to dose 1, dose 5, and an additional intermediate dose chosen at the interim analysis. The interim decision making will be based on Bayesian criteria, as discussed in detail below (Section 2.5). The evaluation of the criteria will be based on a semi-parametric dose-response model described in Section 2.3. Prior information from the previous phase IIa trial is also used (Section 2.4).

1.2.3 Bayesian Model Averaging and Monotone Dose-Response Model

Based on biological and pharmacological knowledge, one can often assume that the dose-response relationship is monotone. A commonly used parametric model to describe monotone dose-response curves is the sigmoid E_{max} model Tan et al. (2011). However, when only few doses (3-5) are used, such a model is often difficult to fit. We propose instead a semiparametric model based on the following monotonicity constraints between the mean effects of the dose groups:

$$\mu_1 \geq \mu_2 \geq \mu_3 \geq \mu_4 \geq \mu_5$$

where $\mu_j = \mathbb{E}[Y_{i,j}]$ and $Y_{i,j}$ is the number of events for patient i in dose group j. We assume that at least one inequality is strict (i.e. there is at least one $>$ sign in place of $\geq$) and we adopt a Bayesian Model Averaging (BMA) approach (see, for example, Hoeting et al. (1999)). Within the BMA approach we put a prior over each model, represented by a different combination of equalities and inequalities, and use Markov Chain Monte Carlo (MCMC) methods to simulate from the posterior distributions of interest.

Formally, BMA is based on three components:

1. A set of M mutually exclusive models $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_M\}$. In each model $\mathcal{M}_m,\ m = 1, ..., M$ a probability mass function $f(y|\theta^{(m)}\mathcal{M}_m)$ is specified for Y, up to the unknown parameter $\theta^{(m)}$;
2. a prior density on $\theta^{(m)}$ for each model $\mathcal{M}_m$, denoted by $g(\theta^{(m)}|\mathcal{M}_m)$;
3. a vector of prior model probabilities $\pi = (\pi_1, ..., \pi_M)$, assigned to the collection $\mathcal{M}$, with $\pi_m = \mathbb{P}(\mathcal{M}_m),\ m = 1, ..., n$. As a default, a uniform discrete distribution $\pi_m = 1/n$ for all m may be assumed, and we will do so.

Combining the first two components, the marginal likelihood for each model is given by

$$p(y|\mathcal{M}_m) = \int f\big(y|\theta^{(m)}, \mathcal{M}_m\big) g\big(\theta^{(m)}|\mathcal{M}_m\big) \mathrm{d}\theta^{(m)}.$$

The posterior probability of the model is then given by:

$$\mathbb{P}(\mathcal{M}_m|y) = \frac{p(y|\mathcal{M}_m)\mathbb{P}(\mathcal{M}_m)}{\sum_k p(y|\mathcal{M}_k)\mathbb{P}(\mathcal{M}_k)}$$

The posterior model probabilities can be combined to produce model averaged predictions for any event of interest:

$$\mathbb{P}(\text{event}|y) = \sum_{m=1}^{M} \mathbb{P}(\text{event}|\mathcal{M}_m, y)\mathbb{P}(\mathcal{M}_m|y)$$

where y are the observed data. For a more complete introduction of this technique and for some bibliographical references see Hoeting et al. (1999) and Ohlssen and Racine (sion), where this approach is applied to continuous and binary outcomes for model selection in the framework of studies in early clinical development.

Notice how we use BMA as a semi-parametric modeling tool to base our curve-free inference upon, not unlike Yin and Yuan (2009). We are therefore not interested in the posterior model probabilities as we would be if we were dealing with a model finding question. The choice of the ensemble of models we are averaging over is also functional only to this purpose.

1.2.4 Case Study: BMA in MS practice

The choice of the model space has to be wide enough to provide generality and strength to the inference, but has to exclude those models which, being biologically unlikely, may misguide our analysis. Pharmacological knowledge may help to rule out some models. For example, in the case study considered here the following models

- $\mu_1 \geq \mu_2 \geq \mu_3 = \mu_4 \geq \mu_5$
- $\mu_1 = \mu_2 \geq \mu_3 = \mu_4 \geq \mu_5$
- $\mu_1 \geq \mu_2 = \mu_3 \geq \mu_4 \geq \mu_5$
- $\mu_1 \geq \mu_2 = \mu_3 = \mu_4 \geq \mu_5$

i.e. the ones which imply a flat region in the middle of the dose-response curve, are to be excluded as being biologically unlikely. The null model

- $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$

is ruled out as the previous Phase IIa study was successful. The models assumed to make up the model space are the remaining ones:

$\mathcal{M}_1$: $\mu_1 > \mu_2 > \mu_3 > \mu_4 > \mu_5$

$\mathcal{M}_2$: $\mu_1 = \mu_2 > \mu_3 > \mu_4 > \mu_5$

$\mathcal{M}_3$: $\mu_1 > \mu_2 > \mu_3 > \mu_4 = \mu_5$

$\mathcal{M}_4$: $\mu_1 > \mu_2 > \mu_3 = \mu_4 = \mu_5$

$\mathcal{M}_5$: $\mu_1 = \mu_2 = \mu_3 > \mu_4 > \mu_5$

$\mathcal{M}_6$: $\mu_1 = \mu_2 > \mu_3 > \mu_4 = \mu_5$

$\mathcal{M}_7$: $\mu_1 = \mu_2 = \mu_3 = \mu_4 > \mu_5$

$\mathcal{M}_8$: $\mu_1 = \mu_2 > \mu_3 = \mu_4 = \mu_5$

$\mathcal{M}_9$: $\mu_1 = \mu_2 = \mu_3 > \mu_4 = \mu_5$

$\mathcal{M}_{10}$: $\mu_1 > \mu_2 = \mu_3 = \mu_4 = \mu_5$

Since BMA is used here as a tool and not as an end, we focus on uniform prior assessments. This working choice seems to go well with our intention to smooth over all plausible models. The Bayesian model according to the BMA approach can now be fully specified as follows, for $i = 1, ..., n_j$, $j = 1, ..., 5$ and $m = 1, ..., 10$:

- Y_{ij} given λ_{ij} is Poisson distributed: $Y_{ij} \sim \text{dpois}(\lambda_{ij})$;
- λ_{ij} is gamma distributed: $\lambda_{ij} \sim \text{dgamma}(\alpha_j, \beta)$, so that $\mu_j = \alpha_j/\beta$ and Y_{ij} is negative-binomial distributed. The scale parameter β is assumed identical across dose groups to reach identifiability;
- $\log(\beta) \sim \mathcal{N}(0, \sigma_\beta^2)$, which corresponds to β around one on average. The scale parameter β is assumed identical across dose groups to reach identifiability; the αs are then proportional to the mean number of lesions;
- the prior distribution of the α's is given in a series of steps, starting from $\log(\alpha_1) \sim \mathcal{N}(\nu_\alpha, \sigma_\alpha^2)$ and proceding as described below.

To model the monotonicity constraint under the m-th model and to complete the specification of the distribution of the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$ we introduce the jump variables $\delta_{k,m} = \log(\alpha_k) - \log(\alpha_{k+1})$ and we put a truncated normal prior on

$$\delta_{sum} = \sum_{k=1}^{4} \delta_{k,m} = \log(\alpha_1) - \log(\alpha_5) \sim \mathcal{TN}(\mu_{sum}, \sigma_{sum}^2),\ m = 1, ..., 10.$$

the maximum differential effect compared to Dose 1. $\mathcal{TN}$ means truncated normal, a normal distribution folded around its mean. More precisely, if $Z \sim \mathcal{N}(0, 1)$ then $X \sim \mathcal{TN}(\nu, \tau^2)$ iff $X = \nu + \tau|Z|$. The different models are subject to the following constraint:

$$\log(\alpha_1) \underbrace{\geq}_{\delta_{1,m}} \log(\alpha_2) \underbrace{\geq}_{\delta_{2,m}} \log(\alpha_3) \underbrace{\geq}_{\delta_{3,m}} \log(\alpha_4) \underbrace{\geq}_{\delta_{4,m}} \log(\alpha_5).$$

which allows us, in cases in which $\alpha_k > \alpha_{k+1}$, to define $\delta_{k,m} = \log(\alpha_{k+1}) - \log(\alpha_k) > 0$

This means that the number of δ's depends on the model: for a given model, the $\boldsymbol{\alpha}$ vector follows a continuous multivariate distribution the support of which is a subspace of dimension less than or equal to five, depending on the chosen model.

The different configurations of the δ's are summarized in the following jump×model matrix:

$$\Delta = \begin{pmatrix} \delta_{1,1} & 0 & \delta_{1,3} & \delta_{1,4} & 0 & 0 & 0 & 0 & 0 & \delta_{1,10} \\ \delta_{2,1} & \delta_{2,2} & \delta_{2,3} & \delta_{2,4} & 0 & \delta_{2,6} & 0 & \delta_{2,8} & 0 & 0 \\ \delta_{3,1} & \delta_{3,2} & \delta_{3,3} & 0 & \delta_{3,5} & \delta_{3,6} & 0 & 0 & \delta_{3,9} & 0 \\ \delta_{4,1} & \delta_{4,2} & 0 & 0 & \delta_{4,5} & 0 & \delta_{4,7} & 0 & 0 & 0 \end{pmatrix} \tag{1.1}$$

where each column represents a different model, for example

$$\mathcal{M}_5 : \mu_1 = \mu_2 = \mu_3 > \mu_4 > \mu_5 \Rightarrow \log(\alpha_1) = \log(\alpha_2) = \log(\alpha_3) \underbrace{>}_{\delta_{3,5}} \log(\alpha_4) \underbrace{>}_{\delta_{4,5}} \log(\alpha_5).$$

For each model (i.e. for each column) the overall treatment effect δ_{sum} is partitioned along each column using a "stick-breaking" construction involving independent uniform random variables. For example, consider model 2, let $s_{k,2} \sim U(0,1),\ k = 1,2$ and let $s_{(1),2} \leq s_{(2),2}$ be their order statistics: the prior for the first change point is $\delta_{2,2} = s_{(1),2} \cdot \delta_{sum}$, while for the second one we have $\delta_{3,2} = (s_{(2),2} - s_{(1),2}) \cdot \delta_{sum}$ and for the third and last one we have $\delta_{4,2} = (1 - s_{(2),2}) \cdot \delta_{sum}$.

After putting a prior on the collection of models, the stick breaking construction allows us to average over all of them without any parametric assumption on the dose-response curve.

1.2.5 Prior Information Based on Historical Trials

Before starting the phase IIb dose finding study, typically a phase IIa study comparing a very high tolerated dose with placebo is available. Such a study can provide relevant prior information on the effect of the drug at the maximal tolerated dose. The study also provides information on the placebo effect, and there may be further historical information on the placebo effect from other clinical trials. The historical data can then be used to derive prior information for the phase IIb studies, for example using the meta-analytic predictive approach Neuenschwander et al. (2010) Gsteiger et al. (2013).

Based on the previous phase IIa trial, the prior was defined by chosing the hyperparameters as follows:

$$\nu_\alpha = 0.1 \quad \sigma_\alpha^2 = 0.49 \quad \sigma_\beta^2 = 0.64 \quad \mu_{sum} = 1.2 \quad \sigma_{sum}^2 = 0.81$$

Putting a prior on the collection of models allows us to average the dose-response curve over all models without strong parametric assumptions. Notice how the variances are given on the logarithmic scale and become then much larger when exponentiated.

1.2.6 Design Issues and Decision Rules

The planned study is parallel-arm, placebo-controlled, highest dose-controlled with one interim look after all enrolled patients complete three months treatment. Based on interim data, a decision will be made whether to stop for success, to stop for futility, or to proceed further. The decision will depend on the Predictive Probabilities for the several alternative post-interim dose selections.

Define the following criteria of acceptability of the dose d, to be used both at interim and at the end of the study:

Exclusion criterion $\mathbb{P}\{\mu_d/\mu_1 \geq 0.7|\text{data}\} \geq 50\%$, i.e. the efficacy of dose d is not better than placebo to a clinically relevant extent.

Efficacy criterion is the intersection of the following events:

(i) $\mathbb{P}\{\mu_d/\mu_1 < 1|\text{data}\} \geq 95\%$, i.e. with very high probability, we require dose d to be superior to dose 1.

(ii) $\max\left\{\mathbb{P}\{\mu_d/\mu_1 \leq 0.5|\text{data}\}, \mathbb{P}\{\mu_d/\mu_5 \leq 1.2|\text{data}\}\right\} \geq 50\%$, i.e., we require the dose to be either at least 50% better than Dose 1, or at most 20% worse than Dose 5.

The reason for setting the second condition is to take into account the possibility that even the effect of Dose 5, due to between-study variation, is close to Dose 1, hence reducing the expected effects of the three doses.

We want to explore the different strategies and prune out those we know to be not-optimal, as illustrated in Figure 1.1. After an initial allocation ratio $1 : a : b : c : 1$ to the five doses, during the interim evaluation, doses 2, 3 and 4 are tested for futility according to the Exclusion criterion. This allows us to prune out those branches that will not lead to a candidate dose and allows us to reduce the space of decision. After dropping the futile doses we proceed as follows:

- if dose 4 has been dropped, stop the trial for futility since, due to the monotonicity constraints, also doses 2 and 3 have to be dropped and we recommend dose 5 for Phase III. We know in fact that the effect of dose 4 is an upper bound for the effects of the other doses, so, if this branch has to be pruned because it is sub optimal, so will be the others.

- if dose 2 meets the Efficacy criterion, stop for success: dose 2 is declared to be the LED to be recommended for Phase III, in fact the effect of dose 2 is a lower bound for the effects of the other doses, hence the other branches will not be superior when penalizing for the increase in the dosage;

- otherwise we will apply the following branch and bound procedure: let j be the lowest dose that has not been dropped, let A_j denote the allocation which contemplates Dose j for second stage in addition to Dose 1 and Dose 5. Then, either the Predictive Probability of a dose to be effective, is greater or equal than a prespecified threshold t, or we prune this branch out of the decision tree and we repeat the procedure with

Figure 1.1: Decision Tree for the LED. Starting from the initial node we move to the Interim Decisions of either stopping the study for Futility or Success or allocating to the most promising dose, from there we proceed to the final analysis which is either going to claim one of the doses is LED, to be used in Phase III, or to assess that the LED does not exist ($\hat{LED}\nexists$), in which case d_5 is going to be used in Phase III. Selecting a dose at Interim does not exclude the other doses to be claimed LED. For example a possible path is highlighted in the above tree in which we select dose 3 for the Interim phase, dose 2 being borderline, but enough evidence is accrued by the end of the study to pronounce the latter LED.

dose $j+1$, if this is different from 5, otherwise the study stops and we recommend Dose 5 for Phase III. If the study continues the same criteria will be used in the end for the final analysis: Doses 2, 3, 4 are tested for futility and the lowest not dropped dose which meets the Efficacy criterion is chosen as the LED.

At the end of the study the same criteria will be used for the final analysis: those among Doses 2, 3 and 4 not discarded at interim will be tested for futility and the lowest not dropped dose which will meet the Efficacy criterion will be chosen as the LED. Notice how the successful dose in the end can be different from the selected dose at interim.

1.2.7 Predictive Probability

For the ease of notation let us define:

$Y = \{\textbf{Past Data}\}$ (in our case, data available *at interim*);

$Y^* = \{\textbf{Future Data}\}$ (in our case, *post interim data*).

A general definition of Predictive Probability (PP) of success is

$$PP = \mathbb{P}\{\text{Success}|Y\} = \mathbb{P}\{Y^* \in \mathcal{Y}_S|Y\} = \int_{\mathcal{Y}_S} p(Y^*|Y)\mathrm{d}Y^* \tag{1.2}$$

for some region $\mathcal{Y}_S$ which defines success. For example, "Success" can be defined as follows:

$$\{\text{Success}\} = \mathcal{Y}_S = \{Y^* : \mathbb{P}\{\theta \in \Theta_E|Y, Y^*\} > c\}$$

for some efficacy domain Θ_E and some threshold c, where θ has the familiar meaning of parameter vector. The predictive quantity (1.2) then becomes

$$PP = \mathbb{P}\{\text{Success}|Y\} = \int_{\mathcal{Y}_S} p(Y^*|Y)\mathrm{d}Y^* = \int \mathbb{1}\{Y^* : \mathbb{P}\{\theta \in \Theta_E|Y, Y^*\} > c\}p(Y^*|Y)\mathrm{d}Y^* \tag{1.3}$$

where $\mathbb{P}\{\theta \in \Theta_E | Y, Y^*\} = \int_{\Theta_E} p(\theta|Y, Y^*)\mathrm{d}\theta$.

Now, consider our setup described in the previous sections and suppose that, at interim, early success has not been claimed and we want to evaluate the decision to allocate patients to three doses: 1, 5 and d, where d is to be chosen among doses not yet discarded for futility (if any) at interim. We base our decision by comparing different PPs over all possible allocations. For each $j \in \{2, 3, 4\}$ not yet futile, let A_j denote post-interim allocation to 1, j and 5. The $j-$th PP, corresponding to allocation A_j, then becomes

$$\mathbb{P}\left\{\left\{\mathbb{P}\{\mu_j/\mu_1 < 1|A_j, Y, Y^*\} > 95\%\right\} \cap \left\{\max\left\{\begin{array}{c}\mathbb{P}\{\mu_j/\mu_1 \leq 0.5|A_j, Y, Y^*\}, \\ \mathbb{P}\{\mu_j/\mu_5 \leq 1.2|A_j, Y, Y^*\}\end{array}\right\} > 50\%\right\}\middle| Y\right\} \tag{1.4}$$

which corresponds to a formal representation of the Efficacy Criterion. Let d the smallest j for which (1.4), representing the Predictive Probability of a dose to be effective, is greater or equal than a prespecified threshold t. Then choose A_d, i.e. continue post-interim with placebo (dose 1), dose d and dose 5.

In practice, Predictive Probabilities are approximated by sums over simulated Y^*, while for the computation of the posterior distribution of the parameters conditioned on the predicted post interim data, i.e. $\mathbb{P}\{\theta \in \Theta_E | Y, Y^*\}$, we rely on the Sampling-Importance Resampling Algorithm (SIR Smith and Gelfand (1992)). Theoretical and computational details are provided in Appendix A.1.

The definition of Success we are using at interim is the event in which the dose gets selected as the LED, called "Selection" in the following. Simulations show how this definition allows us to often select the "right" dose.

1.3 Operating Characteristics of the Adaptive Design

1.3.1 Optimizing the Case Study

The case study concerns the planning of a dose selection trial after a Phase IIa study has shown that the experimental therapy is worth pursuing. The challenge is to optimize the choice of the working parameters a, b, c, i.e. the patient allocations at stage 1 (being the allocation ratio $1 : a : b : c : 1$) and t, the threshold for the Predictive Probability decision. An extensive simulation study is used in order to tune those parameters.

1.3.2 Scenarios

Although we are proposing a Bayesian adaptive design, frequentist operating characteristics are important. Berry et al. (2010). We conduct simulations to explore the behavior of our adaptive design for different scenarios. The superscript (0) is used to identify values of the parameters of the distribution we will simulate our data from, in this sense the Scenario(s) are defined in terms of the "true" value of the unknown parameters we are trying to estimate.

The following features, dose-response, initial allocation, Predictive Probability threshold and number of patients, are to be explored, from which we obtain 54 different combinations of design features and scenarios. To mantain the narration straightforward and not to miss the main picture only one scenario is presented here, while the others are summarized in Appendix C.1.

The **Moderate** Scenario is given by $\boldsymbol{\alpha}^{(0)}_{\text{Mod}} = [1.1, 0.7, 0.1, 0.06, 0.05]$ which implies:

1. $\mu_2^{(0)}/\mu_1^{(0)} = 0.6$ and $\mu_2^{(0)}/\mu_5^{(0)} = 14$, so dose 2 is futile and meets none of the **(ii)** conditions;
2. $\mu_3^{(0)}/\mu_1^{(0)} = 0.09$ and $\mu_3^{(0)}/\mu_5^{(0)} = 2$, so dose 3 meets only the first of the **(ii)** conditions;
3. $\mu_4^{(0)}/\mu_1^{(0)} = 0.04$ and $\mu_4^{(0)}/\mu_5^{(0)} = 1$, so dose 4 meets both of the **(ii)** conditions.

We keep β fixed across dose groups for identifiability reasons; then the αs have the straightforward interpretation of mean number of lesions.

We can see in Figure 1.2 a representation of this dose-response scheme, the light grey bar being the effect under the LED and any dose below the light grey line satisfying the first of the (ii) conditions.

whose scale parameter $\beta^{(0)}$ equals one, the shape parameter $\alpha_j^{(0)} = \mu_j^{(0)}$ for every Dose j.

For what concerns the optimization of the design we will explore different decisions:

- we explore the effects of the initial allocation $a = 1,\ b = 1,\ c = 1$, i.e. $1:1:1:1:1$ and $a = 1,\ b = 2,\ c = 1$, i.e. $1:1:2:1:1$.
- we explore the effects of the adoption of different thresholds for the predictive decision at interim, i.e. $t = 0.4$, $t = 0.5$, $t = 0.6$.
- we explore the impact of using different proportions of the 250 patients for the first and the second stage. The configurations to be considered are 30% for the interim - 70% for the stage 2 - and 50% for the interim - 50% for the stage 2.

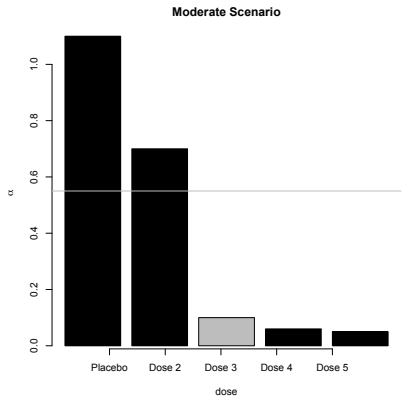

Figure 1.2: True dose-response relationship under the Moderate Scenario

The performance indicators to investigate with our simulations are $\mathbb{P}\{$"right" dose$\}$ and $\mathbb{P}\{$Select a LED$\}$.

Due to the computation burden of the program, we started our sensitivity analysis with 500 studies and $\mathtt{n} = 500$ simulated predictive data set for each dose for the interim decision; the size N of the posterior sample to apply SIR over is chosen to be around 10000; the number of iterations for WinBUGS is fixed to 15000, 500 discarded in the burn-in phase. For each study we simulated all the 50 patients per dose group necessary for the non-adaptive scenario, then we used the same dataset for both the non adaptive and the adaptive scheme in order to compare the performance of the two designs with identical data sets.

Notice how the computations, despite being quite cumbersome, are not problematic for a single study, most of the computational troubles in the present work came from the simulation study which required the use of a proprietary version of the R-WinBUGS interface. The simple implementation will be more than appropriate in a real world application.

1.3.3 Simulations Results

In the following paragraphs, we will call "Right dose" the event of declaring the Right dose the LED, "Selection" the event of declaring some dose the LED. Furthermore FP will mean False Positive and likewise FN False Negative. By Interim allocation we mean the initial allocation scheme used at the beginning of the interim phase, specifically we will indicate by 1:1:1 the Interim allocation scheme 1:1:1:1:1 and by 1:2:1 the Interim allocation scheme 1:1:2:1:1. 30% and 50% are percentage of patients used before the interim look.

Tables 1, 2 and 3 show how the the 30%-70% sample size allocation - for stage 1 and 2 respectively - has a very low probability of early stopping, which is due to the low amount

of information obtained about the dose-response relationship. It is also clear the predictive probability threshold $t = 0.6$ may be too conservative. The performances of the non-adaptive and those of the adaptive design in terms of detecting the LED seems to be almost the same, but the adaptive approach allows us to use a lower number of patients. In fact, while the expected sample size for the non-adaptive analysis is always equal to the total number of patients (in our simulations 250), the expected number of patients needed by the adaptive design is always lower, as illustrated in Table 1.1.

	1:1:1		1:2:1	
	$50\% - 50\%$	$30\% - 70\%$	$50\% - 50\%$	$30\% - 70\%$
Moderate	232	237	238	239

Table 1.1: Expected sample size for the adaptive design (Moderate Scenario), where $\mathbb{E}[\#\text{ of patients}] = (\text{interim sample size}) + (\text{post interim sample size}) \cdot \mathbb{P}\{\text{not stopping at interim}\}$

Another good characteristic of the adaptive dose selection procedure is the possibility of correcting a sub-optimal decision made at interim: in fact, if a"wrong" dose is selected at interim, the BMA and the interim data in the final analysis lead anyway to the detection of the "right" dose.

From the clinical prospective, a small study is preferable to a large study, which means that in comparing the adaptive and the non-adaptive approach the early stopping probability has a positive weight in the evaluation of the method. So, even if the two approaches are equivalent in performances, the adaptive design, requiring a lower number of patients, is to be preferred.

The Non-Adaptive Design, when simulating from the above presented Moderate Scenario, detected the right (dose 3 in this example) dose with probability $\mathbb{P}\{\text{Right dose}\} = 0.84$ and called some dose LED in all cases (i.e. $\mathbb{P}\{\text{Selection}\} = 1.000$).

In addition to the Moderate scenario, four other scenarios for the true dose-response curve were also considered (Optimistic, Intermediate 1, Intermediate 2, Pessimistic), and the corresponding operating characteristics were again obtained by simulation; see Appendix C.1 for details. For some of the scenarios, the expected sample size was considerably reduced in the adaptive design, as compared to the non-adaptive design, e.g. by 50% in the Optimistic scenario, with an initial allocation of 1:1:1, and 30% of patients used in the first stage. The probabilities of selecting the Right dose, and the probability of success where comparable to the non-adaptive design.

	1:1:1-30%	1:1:1-50%	1:2:1-30%	1:2:1-50%
$\mathbb{P}$\{False Positive\}	0.08	0.13	0.07	0.1
$\mathbb{P}$\{False Negative\}	0.00	0.00	0.00	0.00

Table 1.2: Moderate Scenario: Interim Decision

	1:1:1-0.4	1:1:1-0.5	1:1:1-0.6	1:2:1-0.4	1:2:1-0.5	1:2:1-0.6
$\mathbb{P}$\{Right dose|30%,70%\}	0.82	0.86	0.81	0.73	0.78	0.86
$\mathbb{P}$\{Selection|30%,70%\}	1.00	1.00	1.00	1.00	1.00	1.00
$\mathbb{P}$\{False Negative|30%,30%\}	0.00	0.00	0.00	0.00	0.00	0.00
$\mathbb{P}$\{Right dose|50%,50%\}	0.77	0.82	0.77	0.73	0.79	0.85
$\mathbb{P}$\{Selection|50%,50%\}	1.00	1.00	1.00	1.00	1.00	1.00
$\mathbb{P}$\{False Negative|50%,50%\}	0.00	0.00	0.00	0.00	0.00	0.00

Table 1.3: Moderate Scenario: different allocations and different predictive probability thresholds

1.4 Conclusions and Discussion

Ideally, dose finding studies should include a wide range of doses to allow for more accurate dose-response modeling. However, in many situations, due to availability of formulation, tablet strength, and patient number constraints, one can only study a limited number of doses and regimens. We proposed the evaluation of a dose-response under monotone constraints which performed well even when the doses studied were too few for the estimation of a parametric model. To optimize the study, an adaptive design was proposed based on Predictive Probability. To evaluate the operating characteristics, devoted methods were required, such as Sampling Importance Resampling Smith and Gelfand (1992). There are no sufficient statistics available for over-dispersed count data, resulting in more demanding computing.The SIR algorithm was therefore implemented for calculating the predictive probability of success at the interim analysis; this reduced the computing burden of the simulations to an acceptable level.

The case study considers a POC trial designed to look for a dose which can improve, in terms of safety, upon a therapy whose efficacy has already been proved. Even if in many settings POC studies are small and don't accrue enough evidence to motivate a full phase III trial, in the considered example a phase III has already been planned and the study is run to refine the characteristic of such a trial. In other situations, the decision rules may be modified to include futility rules such that if results are unfavorable for all doses, no phase III would be started. As such futility rules can also be expressed as criteria on the posterior distribution, the same framework can be considered as presented in this article.

The equal allocation 1:1:1:1:1 gives the best results, allowing for the exploration of a wider range of doses. Its performances were similar to those of the non-adaptive design,but the number of patients required much lower. The adaptive design proposed here is conservative, one can only stop if either the lowest active dose is LED or the Dose 4 is futile. This are two very extreme but meaningful events. From the observation of the frequency with which, for example, Dose 2 gets selected one might want to taylor them by specifying a more complicated loss function linked, for example, to the joint examination of the chances of success of all the intermediate doses.

If one is willing to stop at interim when there is sufficient evidence that Dose 3 is LED, that will reduce the number of patients. Often, there are logistical reasons and recruitment constraints when choosing one design over the other. The implementation of these simulations assumes that recruitment will be stopped once sufficient patients are recruited for the interim; this may not be realistic. One needs to consider other important factors of a study in addition to the operating characteristics for the design. The proposed approach is, however, flexible enough to be adapted to most of these settings.

In this article, we described a Bayesian adaptive dose selection procedure for overdispersed count data, motivated by clinical trials in MS, where counts are the primary endpoint. Both the adaptive design and the semi-parametric approach can be easily modified for use with other endpoints.

Chapter 2

A Bayesian hierarchical surrogate outcome model for multiple sclerosis

2.1 Introduction

Within a clinical trial setting, a surrogate endpoint is a biomarker that can be used instead of the main outcome of interest in the evaluation of an experimental treatment. While the treatment effect associated with a surrogate endpoint might not be of any direct value to a patient, it can be used to predict the corresponding effect that would have been achieved in key clinical outcomes. For example, based on experience from numerous long-term outcome studies of diverse antihypertensive drugs showing a clear effect on stroke and at least favorable trends on cardiovascular events, blood pressure has become an established surrogate outcome for such therapies and can form the basis for regulatory approval Temple (1999).

In situations where the evidence linking the surrogate to the main clinical outcome is not deemed sufficient for regulatory acceptance, it could still be very valuable in non-confirmatory phases of drug development. This is particularly the case, when the primary clinical endpoint is costly to obtain or requiring a substantial amount of time to observe. However, due to the increased uncertainty associated with use of a surrogate, additional risks are inherent in development decision making Weir and Walley (2006). Therefore, it is essential to gain a quantitative understanding of the evidence linking the surrogate to the main outcome of interest that can then be utilized in the design of clinical studies and the formulation of development decision criteria.

The development of novel therapies for the treatment of multiple sclerosis (MS) is one area where a range surrogate outcomes are used in various stages of clinical research. While the aim of treatments in MS is to prevent long term irreversible disability, a clinical trial for

evaluating a drugs effect on disability progression would require a large sample of patients with many years of follow-up. In early stages, the vast majority of MS cases typically present as relapsing disease (Relapsing Remitting MS, RRMS) where patients suffer from bouts of neurological deficits, affecting the motor, sensory and vegetative nervous system from which they usually recover with little or no sequelae. As such, to reduce study size and duration, clinical relapses are usually used as primary endpoints in phase III trials. In the case of phase II studies, the primary outcomes are typically based on Magnetic Resonance Imaging (MRI), usually in the form of lesion counts, that are around 5 to 10 times more sensitive than clinical measures for detecting MS activity. Although MRI imaging is not accepted for regulatory approval, it is typically one of the most important factors in decision making in drug development.

Recent work by Sormani and colleagues provided a systematic review and meta-analysis examining the role of both MRI and relapse as trial level surrogate outcomes for disability Sormani et al. (2010). Trial level surrogacy is an approach that links the potential surrogate outcome with main outcome of interest by using summary data from each of the trials identified in a systematic review. Typically, the summary information consists of the estimated treatment effect and standard error associated with both the main outcome and surrogate.

In the analysis presented in Sormani et al. (2010), two separate weighted least-squares regressions were fitted. The first relating the observed treatment effect on MRI lesions to the observed treatment effect on disability (adjusted $R^2 = 0.57$), and the second relating the observed treatment effect on relapses to the observed treatment effect on disability (adjusted $R^2 = 0.71$). This approach has two main drawbacks: First, it ignores that the independent variable, which in this case is the observed treatment effect on the potential surrogate, is measured with error Qin (2011). Second, a biomarker combining information from both MRI lesions and relapses may be a better surrogate for disability than each of them alone Sormani et al. (2011). To address these two drawbacks, we propose a joint analysis, linking both MRI and relapses to disability, and also take into account the measurement errors in the biomarkers. To this purpose, we expand the Bayesian trial level surrogate outcome meta-analysis model introduced by Daniels and Hughes (1997) to a three level structure. This model is then used to re-analyze the data provided in the systematic review by Sormani and colleagues.

The remainder of the Chapter is structured as follows: In section 2.2 we describe some background on statistical approaches to examining surrogate outcomes and outline the methodological developments. Subsequently, section 2.3 provides further background on the running example and describes how the key covariance estimates are extracted. Section 2.4 applies the methods developed in section 2.2. Finally, a discussion of the application and methods is presented in section 2.5.

2.2 Methods

The seminal work by Prentice Prentice (1989) subsequently lead to large literature examining statistical evaluation of surrogate outcomes (see Molenberghs et al. (2002) and Weir and Walley (2006) for an extensive overview). Within this area statistical methods have been divided into three groups: Those focusing on data from a single trial such as the Prentice criteria; those which use meta-analysis to combine summary information from multiple trials Daniels and Hughes (1997), Gail et al. (2000); and techniques that use a combination of trial-level and individual-level data Molenberghs et al. (2010). As this Chapter provides an analysis of summary data from multiple clinical trials, with the aim of understanding uncertainty associated with drug development decisions rather than formal validation, we shall solely focus on the Bayesian meta-analytic approach that was developed by Daniels and Hughes (1997).

2.2.1 Two level Surrogate outcome Bayesian meta-analysis model

To provide a framework and corresponding notation, which shall be adopted throughout, a data structure is assumed where there are $i = 1, \ldots, n$ trials and $k = 1, 2, \ldots, K$ possible treatments. In the case of two treatments ($K = 2$), we assume the following sampling model applies to trial i:

$$\begin{pmatrix} \hat{\theta}_i \\ \hat{\gamma}_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \theta_i \\ \gamma_i \end{pmatrix}, \begin{pmatrix} \sigma^2_{\theta i} & \rho_i \sigma_{\theta i} \sigma_{\gamma i} \\ .. & \sigma^2_{\gamma i} \end{pmatrix} \right) \tag{2.1}$$

where $\hat{\theta}_i$ represents the estimated treatment effect on the main outcome of interest, $\hat{\gamma}_i$ denotes the the estimated treatment effect associated with the surrogate outcome, $\sigma^2_{\theta i}$ and $\sigma^2_{\gamma i}$ are the variances that reflect the sampling uncertainty and ρ_i is the correlation between the estimated treatment differences conditional on the true differences θ_i and γ_i.

The simplest trial-level model to describe the relationship between θ_i and γ_i is linear:

$$\theta_i = \alpha_1 + \beta_1 \gamma_i + \epsilon_i \tag{2.2}$$
$$\epsilon_i \sim \mathcal{N}\left(0, \tau^2_\epsilon\right) \tag{2.3}$$

In this context, γ_i are usually assumed to be a set of separate fixed effects.

The combination of the sampling model given in (2.1) with the linear model given in (2.2) -(2.3) leads to a bivariate normal distribution, conditional on γ_i, α_1 and β_1:

$$\begin{pmatrix} \hat{\theta}_i \\ \hat{\gamma}_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \alpha_1 + \beta_1\gamma_i \\ \gamma_i \end{pmatrix}, \begin{pmatrix} \sigma^2_{\theta i} + \tau^2_\epsilon & \rho_i\sigma_{\theta i}\sigma_{\gamma i} \\ .. & \sigma^2_{\gamma i} \end{pmatrix} \right) \tag{2.4}$$

When fitting the model in the Bayesian framework prior distributions must be assumed. In the multiple sclerosis case-study, $\mathcal{N}(0, 10^6)$ priors will be assumed for all fixed effects (α_1, β_1 and $\gamma_i\ i = 1, \ldots, n$). This choice, or more generally a normal distribution with a large variance relative to the scale of the outcome variable, is often a reasonable non-informative prior for fixed effects. In terms of the variance component τ^2_ϵ, it is well known that results can be sensitive to the choice of prior distribution Gelman (2006). Following the suggestions of Spiegelhalter et al Spiegelhalter et al. (2003), a standard half normal prior, denoted by Half-Normal(1) (i.e. with scale parameter set to 1), will be adopted, providing a weakly informative prior over a range of plausible values on the log relative risk scale. The parameters of the covariance matrix σ_{θ_i}, τ_ϵ and σ_{θ_i} are typically assumed to be known in the meta-analytic literature.

If studies have more than two arms the analysis must account for the correlation between the multiple treatment effects within a study. This can be accomplished by adjusting the sampling model, e.g. to a 4-dimensional multivariate normal model in the case of three treatment arms to:

$$\begin{pmatrix} \hat{\theta}_{1\mathrm{i}} \\ \hat{\theta}_{2\mathrm{i}} \\ \hat{\gamma}_{1\mathrm{i}} \\ \hat{\gamma}_{2\mathrm{i}} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \theta_{1i} \\ \theta_{2i} \\ \gamma_{1i} \\ \gamma_{2i} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\theta_1,i} & \rho_{\theta,i}\sigma_{\theta_1,i}\sigma_{\theta_2,i} & \rho_{1,1,i}\sigma_{\theta_1,i}\sigma_{\gamma_1 i} & \rho_{1,2,i}\sigma_{\theta_1,i}\sigma_{\gamma_2,i} \\ .. & \sigma^2_{\theta_2,i} & \rho_{2,1,i}\sigma_{\theta_{1,2},i}\sigma_{\gamma_1,i} & \rho_{2,2,i}\sigma_{\theta_{1,2},i}\sigma_{\gamma_2,i} \\ .. & .. & \sigma^2_{\gamma_1,i} & \rho_{\gamma,i}\sigma_{\gamma_1,i}\sigma_{\gamma_2,i} \\ .. & .. & .. & \sigma^2_{\gamma_2,i} \end{pmatrix} \right). \tag{2.5}$$

2.2.2 Extension to a three level surrogate outcome Bayesian meta-analysis model

If two surrogates are considered, as in the case of relapsing remitting MS case study, the sampling model for study i becomes

$$\begin{pmatrix} \hat{\theta}_i \\ \hat{\gamma}_i \\ \hat{\psi}_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \theta_i \\ \gamma_i \\ \psi_i \end{pmatrix}, \begin{pmatrix} \sigma^2_{\theta i} & \rho_{\theta,\gamma,i}\sigma_{\theta i}\sigma_{\gamma i} & \rho_{\theta,\psi,i}\sigma_{\theta i}\sigma_{\psi i} \\ .. & \sigma^2_{\gamma i} & \rho_{\gamma,\psi,i}\sigma_{\gamma i}\sigma_{\psi i} \\ .. & .. & \sigma^2_{\psi i} \end{pmatrix} \right), \tag{2.6}$$

where $\hat{\psi}_i$ represents the estimated treatment effect from the second surrogate, $\sigma^2_{\psi i}$ is the corresponding variance representing sampling uncertainty and the set of correlations between the estimated treatment effects are denoted by $\rho_{\theta,\gamma,i}$, $\rho_{\theta,\psi,i}$, $\rho_{\gamma,\psi,i}$.

When formulating the trial level model, a natural extension of the Daniels and Hughes model would involve two levels of linear models:

$$\begin{aligned}
\theta_i &= \alpha_1 + \beta_1\gamma_i + \epsilon_i && (2.7)\\
\gamma_i &= \alpha_2 + \beta_2\psi_i + \delta_i && (2.8)\\
\epsilon_i &\sim \mathcal{N}\left(0, \tau_\epsilon^2\right) && (2.9)\\
\delta_i &\sim \mathcal{N}\left(0, \tau_\delta^2\right). && (2.10)
\end{aligned}$$

Here, γ_i are assumed to be a set of separate fixed effects, as usually considered in the meta-analytical context.

Various alternative structures could be applied when formulating the three level model. Here, we shall consider two alternative models: Firstly, equation (2.7) could be modified so that both surrogates contribute directly in the linear predictor:

$$\theta_i = \alpha_1 + \beta_1\gamma_i + \beta_3\psi_i + \epsilon_i. \tag{2.11}$$

Secondly, instead of formulating a conditional structure, the second level of the model could be formulated as a multivariate normal random effects meta-analysis model (see van Houwelingen et al. (2002) for further background):

$$\begin{pmatrix} \theta_i \\ \gamma_i \\ \psi_i \end{pmatrix} \sim \mathrm{MVN}\Big((\mu_1, \mu_2, \mu_3)', \mathbf{\Sigma}\Big). \tag{2.12}$$

For each of the possible models prior distributions must be chosen. Following the suggestions made in section 2.2.1, in the case-study $\mathcal{N}(0, 10^6)$ priors will be assumed for all fixed effects ($\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \mu_1, \mu_2, \mu_3$ and $\psi_i\, i = 1, \ldots, n$) and Half-Normal(1) (i.e. with scale parameter set to 1), will be adopted for the variance components $(\tau_\delta, \tau_\epsilon)$. In the case of the multivariate meta-analysis, Half-Normal(1) priors will also be adopted for the between study standard deviations (i.e. the square root of the diagonal elements of Σ) and uniform priors will be used over the 3-dimensional set of admissible correlations, which were formed using Cholesky-decomposition Turner et al. (2006).

With a number of models under consideration, some form of model assessment and comparison should be implemented. Here, we suggest examining the deviance information criterion (DIC) Spiegelhalter et al. (2002) and graphics that compare predicted outcomes, based on the predictive distribution from the model, with the observed values. Both methods will be applied in the case-study.

2.2.3 Practical issues when extracting trial level data

For each of the studies included in the surrogate outcome meta-analysis, an estimated variance covariance matrix must be formed. In the case of sampling variances, $\sigma^2_{\theta i}, \sigma^2_{\gamma i}$ and $\sigma^2_{\psi i}$, estimates can usually be drawn directly from summary data included in a publication, such as a standard deviation, or by applying standard approximate variance formula associated to an assumed likelihood. However, direct estimation of the correlations among the treatment effect estimates either requires the specification of a joint model for the relationship among each of the treatment differences, which is complex if the outcomes are non-commensurate, or application of a non-parametric bootstrapping technique (see Daniels and Hughes (1997), Normand et al. (2007), Riley et al. (2007)). In either case, estimation requires the availability of the individual patient data from each clinical trial which is not available in a systematic review based primarily on summary information available in publications.

If only summary data is available the challenge of dealing with missing sampling covariances or correlations must be addressed. The problem has been discussed in the context of multivariate meta-analysis Riley (2009), where the author suggests: using a range of plausible values; sensitivity analysis over the entire correlation range and finally the use of an alternative model, described in Riley et al. (2007), that does not depend on within study correlation. Because we wish to focus on the modeling developments outlined in sections 2.2.1 and 2.2.2, here sensitivity analysis across a range of reasonable values shall later be utilized in the case-study. This choice of values shall be based on some limited proprietary individual patient data.

2.3 Background to the MS case-study and data extraction

2.3.1 Background to application

Due to the advancement in therapies for Multiple Sclerosis, for ethical reasons, fewer placebo controlled trials will be launched Polman et al. (2008). Therefore, future studies will typically have lower event rates and as a consequence require a combination of more patients and longer duration. Surrogate endpoints in MS have a long history Wolinsky and Beck (2011) and are well motivated given the aforementioned design challenges.

As stated in section 2.1, we focus on the recent work by Sormani and colleagues who provided a systematic review and meta-analysis examining the role of both MRI and relapse as trial level surrogate outcomes for disability progression in trials examining therapies for relapsing-remitting multiple sclerosis (RRMS). Full background details are provided in Sor-

mani et al. (2010), below a brief overview is provided together with a summary of the key findings.

During the systematic review a set of 19 randomized double-blind controlled trials in RRMS were identified, with a total of 44 arms, 25 contrasts, and 10,009 patients. The data extraction, which was conducted independently by 3 reviewers, involved gathering information from each of the studies on the following outcome measures: First, the treatment effect expressed as a risk ratio based on MRI lesion counts. Second, the corresponding risk ratio associated annualized relapse rate. Finally, the risk ratio based on the proportion of patients with a disability progression as assessed by the Expanded Disability Status Scale (EDSS) score.

In the analysis presented in Sormani et al. (2010), set of treatment contrasts within a trial, weights w were specified according the study duration and number of completers $\left(w = n_{complete}\sqrt{\frac{\text{follow up in months}}{12}}\right)$. In the trials with more than two treatments, each of the treatment contrasts were assigned an adjusted weight, accounting for the correlation among the contrasts. Specifically, when forming an adjusted weight, the number of patients associated with the control group was distributed equally among each of the treatment contrasts.

Having derived the set of weights, the following linear weighted regression models were applied:

$$\theta_i = \alpha_1 + \beta_1\gamma_i \; i \in J$$

$$\theta_k = \alpha_2 + \beta_2\psi_k \; i \in K$$

where $\theta_i = \log(\text{EDSS risk ratio})$, $\gamma_i = \log(\text{Relapse risk ratio})$, $\psi_i = \log(\text{MRI risk ratio})$, J is the set of EDSS and relapse treatment effect pairs and K is the set of EDSS and MRI treatment effect pairs. A strong association between the disability treatment effect and the potential surrogate was observed in both cases (relapses - adjusted $R^2 = 0.71$; MRI - adjusted $R^2 = 0.57$). As noted in the introduction, this approach ignores that the independent variable is measured with error Qin (2011). To address these two drawbacks, we propose an analysis based on the methods described in section 2.2. However, as emphasized in section 2.2.3, implementation requires an appropriate set of estimated covariance matrices, which shall be discussed in the next section.

2.3.2 Covariance matrix formulation

The Bayesian surrogate outcome models outlined in section 2.2 are based around forming an approximate normal likelihood from the estimated treatment effects and corresponding estimated variance-covariance matrix. In the multiple sclerosis case study, the treatment effects can be taken directly from the summary data provided in Sormani et al. (2010). However, the corresponding standard errors, which form the diagonal elements of the estimated variance-covariance matrix must be approximated from the available summary information by using a combination of sampling model assumptions, assumptions about drop-out and incomplete follow-up as described in Appendix A.2. Further, the covariance parameters cannot typically be obtained from summary data. In this case, results from the analysis of proprietary data are used to form the basis of a sensitivity analysis.

The final two elements required for the Bayesian analysis are the covariance parameters to deal with multiple arm studies and the covariance parameters among the different types of outcomes. In the case of the former, this was achieved by using the same set of sampling models described above to calculate the squared standard error associated with placebo arm rates, which is equal to the required covariance. While in the case of the latter, a initial value of 0.05 shall be used for each of the missing correlations. The selection of this value was in part informed by exploratory analysis of proprietary data. To assess robustness of the results to this choice, values of 0.1 and 0.01 shall also be used in a sensitivity analysis.

2.4 Analysis of the MS data using Bayesian surrogate outcome models

As an initial step, the two classical weighted regression models, described in Sormani et al. (2010) were applied. Recall that for the set of studies, these models regressed the response variable $\hat{\theta}_i$ ($\log$(EDSS risk ratio)) against either $\hat{\gamma}_i$, ($\log$(Relapse risk ratio)) or $\hat{\psi}_i$ ($\log$(MRI risk ratio)). In addition, an approach that used identical regression models but with the weights based on the inverse variances associated with $\hat{\theta}_i$, which were derived in Section 2.3.2, was implemented. For both sets of weights, the resulting estimated intercepts, slopes and adjusted R^2 values are displayed in Table 2.1. The two approaches produce similar results. However, the adjusted R^2 values associated with the inverse variance weights were slightly lower suggesting a slightly weaker relationship.

Next, the Bayesian surrogate outcome models, described in Section 2.2, were fitted. The two level model was based on regressing θ_i against γ_i, while the three level approach, used the chain of regression models described in (2.7)-(2.10). When fitting these models, and in all subsequent models presented in this section, WinBUGS 1.4.3 Lunn et al. (2000) was used to draw 50000 MCMC samples from each of three parallel sampling chains following

a 50000-iteration burn-in period. Convergence was assessed using the Brooks Gelman and Rubin diagnostic.

The resulting regression models are displayed in Figure 2.1. The display shows the posterior distribution for the regression line $\theta_i = \alpha_1 + \beta_1\gamma_i$, together with a 95% credible interval and the corresponding 95% predictive interval, based on $\theta_i = \alpha_1 + \beta_1\gamma_i + \epsilon_i$,which accounts for the between study variability $\epsilon_i \sim \mathcal{N}(0, \tau_\epsilon^2)$. It is noted, when comparing the 2-level and 3-level models, the regression lines provide similar point-estimates. However, the 3-level model provides a narrower predictive distribution, suggesting a benefit of including both MRI and relapse jointly in a model. The predictive distribution could be particularly valuable when assessing the probability of success of a future phase III study based on phase II study, which would typically have a reasonable amount of MRI data but limited relapse data.

Sensitivity analysis to the choice of sample correlation ρ is examined in Table 2.3. The table displays the posterior means and standard deviations for each of the model parameters resulting from fitting the 2-level and 3-level Bayesian surrogate outcome models, with three different assumed values of ρ. The results suggest robustness over a range of correlations.

The final part of the analysis involved examining alternative modeling structures for the 3-level model based on equations (2.11) and (2.12). In Table 2.2, the deviance information criterion (DIC) is shown for each of the models. The multivariate meta-analysis model and the extended 3-level model both have slightly lower DIC than the initial 3-level model. However, the differences are small suggesting little difference among models in terms of a complexity and fit tradeoff. To investigate further, for each of the models, forest plots examining the study level relapse risk ratios ($\exp(\gamma_i)$)and EDSS risk ratios ($\exp(\theta_i)$) were produced. Figure 2.2 displays posterior means and 95% credible intervals are displayed for the set of risk ratios associated with relapse.

Regressor	Weights	Intercept (s.e.)	Slope (s.e.)	adjusted R^2
$\hat{\gamma}_i$	Sormani	0.1 (0.055)	0.63 (0.087)	0.7
$\hat{\gamma}_i$	Inverse Variance	0.074 (0.05)	0.575 (0.083)	0.676
$\hat{\psi}_i$	Sormani	0.2 (0.097)	0.36 (0.085)	0.56
$\hat{\psi}_i$	Inverse Variance	0.107 (0.091)	0.285 (0.085)	0.465

Table 2.1: The results from weighted linear regression of the response variable $\hat{\theta}_i = \log(\text{EDSS risk ratio})$ against either $\hat{\gamma}_i$, ($\log(\text{Relapse risk ratio})$) or $\hat{\psi}_i$ ($\log(\text{MRI risk ratio})$). Two sets of weights were employed: The first is based on the previous work of Sormani and colleagues. The second is based on inverse variance weights derived from the $\log(\text{EDSS risk ratio})$.

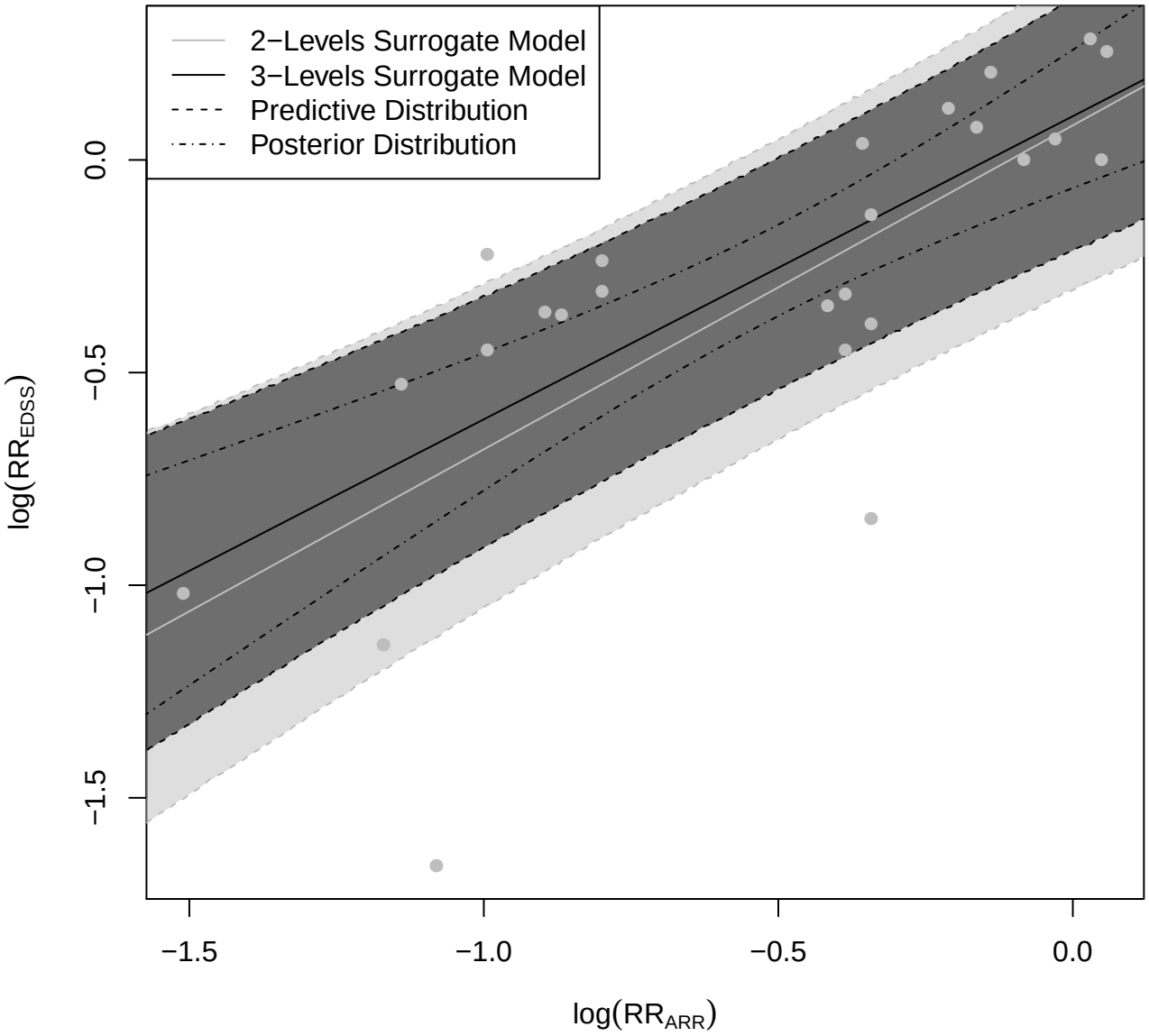

Figure 2.1: The fitted 2-level and 3-level Bayesian surrogate outcome models for $\theta_i = \log(\text{EDSS risk ratio})$ based on γ_i, $(\log(\text{Relapse risk ratio}))$ in the 2-level model or a 3-level model involving γ_i and ψ_i $(\log(\text{MRI risk ratio}))$. The display shows the posterior distribution for the regression line $\theta_i = \alpha_1 + \beta_1\gamma_i$, together with a 95% credible interval and the corresponding predictive interval, based on $\theta_i = \alpha_1 + \beta_1\gamma_i + \epsilon_i$, which accounts for the between study variance component, $\epsilon_i \sim \mathcal{N}(0, \tau_\epsilon^2)$.

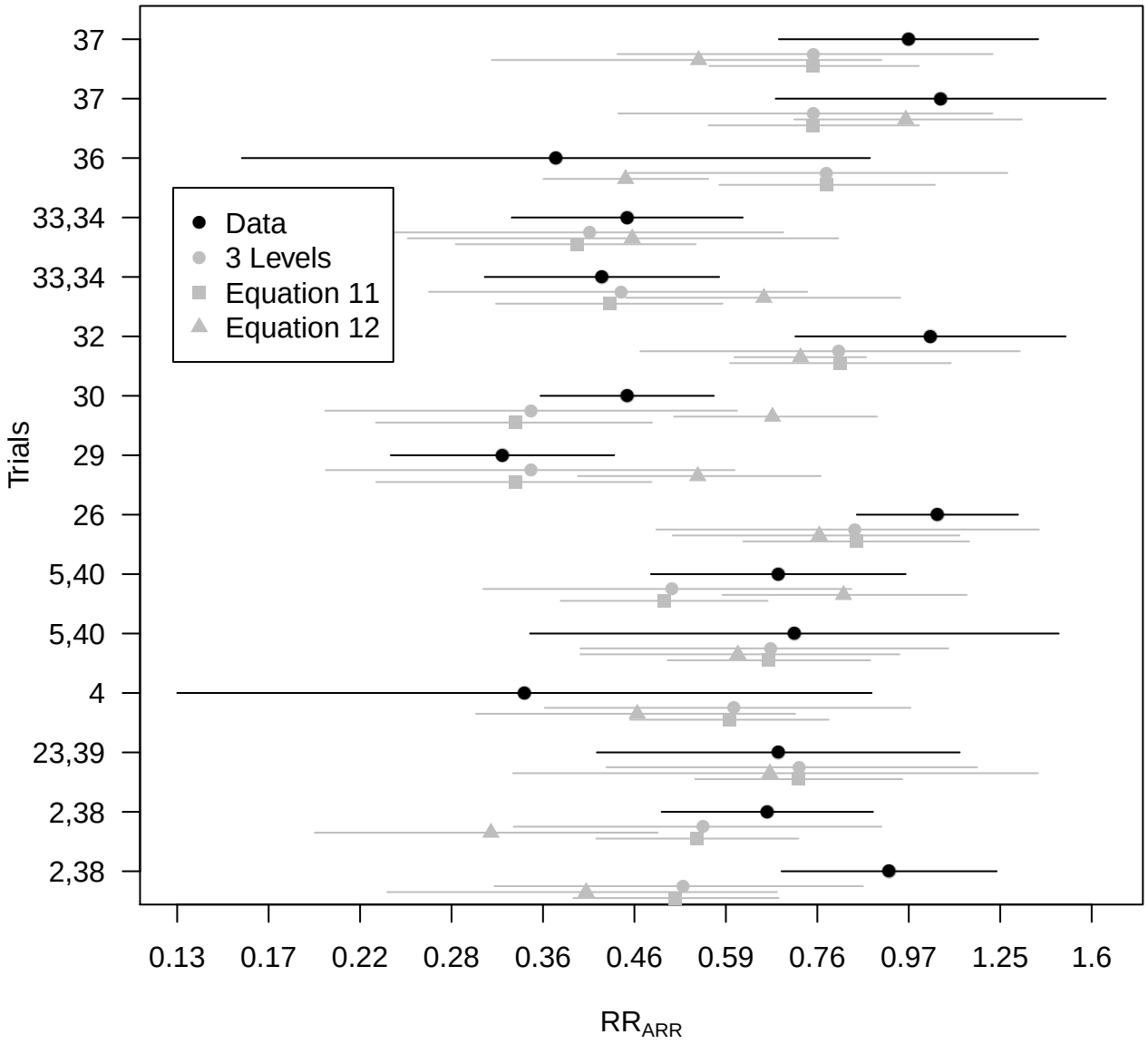

Figure 2.2: Analysis of the multiple sclerosis surrogate outcome data using three alternative surrogate models. Posterior means and 95% credible intervals are displayed for the set of risk ratios associated with relapse. The models are: 3-level model (gray circle)and multivariate meta-analysis (gray triangle). For comparison purposes, the estimates and 95% are displayed from fixed effects model (black circle).

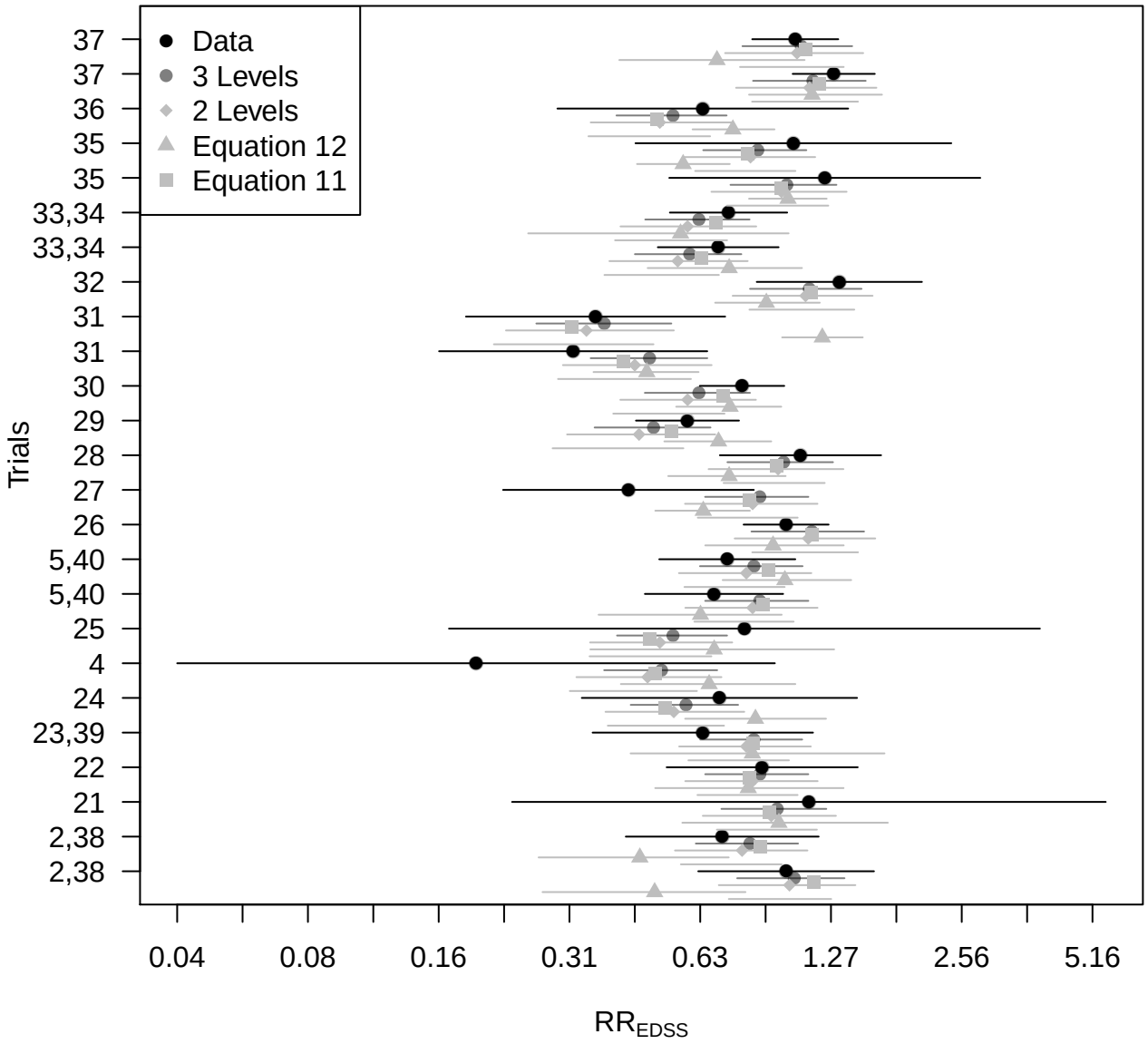

Figure 2.3: Analysis of the multiple sclerosis surrogate outcome data using three alternative surrogate models. Posterior means and 95% credible intervals are displayed for the set of risk ratios associated with EDSS. The models are: 3-level model (gray circle); extended three level model (gray square) and multivariate meta-analysis (gray triangle). For comparison purposes, the estimates and 95% are displayed from fixed effects model (black circle).

Model	$D(\bar{\mu})$	$\overline{D(\mu)}$	pD	DIC
Fixed effects (saturated)	-74.146	-9.162	64.984	55.822
3 level surrogate outcome model	-51.336	-11.006	40.33	29.324
extended 3 level (equation 11)	-52.899	-12.517	40.382	27.865
Multivariate meta-analysis (equation 12)	-59.521	-15.658	43.863	28.205

Table 2.2: Examining model fit and complexity using the Deviance information criteria (DIC). The mean of the posterior deviance is denoted by $\overline{D(\mu)}$, the deviance of the posterior mean is denoted by $D(\bar{\mu})$ and the effective number of parameters is denoted by $pD = \overline{D(\mu)} - D(\bar{\mu})$. Four models are examined: A saturated fixed effects model; 3-level model; the extended 3-level model and multivariate meta-analysis model.

2.5 Discussion

The case-study presented in this Chapter reconsidered a previously published surrogate outcome meta-analysis in the context of multiple sclerosis clinical trials. Specifically, a combination of treatment effects based on MRI lesion count outcomes and clinical relapse, both expressed on the log risk ratio scale, were used in a study level surrogate outcome model for corresponding treatment effects based on disability progression, measured using the expanded expanded disability status scale (EDSS). The work extended a previous Bayesian model described in Daniels and Hughes (1997) to a three level model accommodating the two surrogates.

One of the advantages of the Bayesian model is the ability to provide predictions for the results of future clinical trials that fully account for parameter uncertainty. This can be particularly valuable in a multiple sclerosis drug development setting. For example, a completed phase II study, which would typically have a reasonable amount of MRI data but limited relapse data and no information about disability progression, could be used a the basis for calculating the probability of success of a future phase III program. However, trial durations and population will often change substantially between phase II and phase III. Therefore, assumptions about exchangeability when moving from phase II to phase III must be critically assessed. In some circumstances, when forming a prediction, it would be necessary to discount the value of the historical information.

As noted, when only summary data is available from clinical trial publications, one of the key challenges when developing a study level surrogate outcome meta-analysis model, is dealing with missing sampling covariances or correlations. Here, sensitivity analysis across a range of reasonable correlation values was utilized. Alternative approaches, such as different model formulations, suggested in the multivariate meta-analysis literature Riley (2009), could also be implemented. Nevertheless, access to individual patient data would greatly help with this problem.

Finally, we note, the European Medicines Agency has stated its commitment to publishing clinical trial information on all drugs submitted as part of the licensing process, whether or not they are approved Eichler et al. (2012), Steinbrook (2013). As emphasized above, this information would be invaluable when forming a surrogate outcome model in the multiple sclerosis setting and other disease areas. Hence, combining individual patient data with summary data in a surrogate outcome setting would provide an important area for future research.

Model	Parameter	$\rho = 0.05[95\%CI]$	$\rho = 0.01[95\%CI]$	$\rho = 0.1[95\%CI]$	$\rho = 0[95\%CI]$
2 level	$\hat{\alpha}_1$	0.081 [-0.094, 0.244]	0.08 [-0.097, 0.246]	0.081 [-0.094, 0.246]	0.079 [-0.1, 0.246]
	$\hat{\beta}_1$	0.762 [0.484, 1.021]	0.765 [0.479, 1.028]	0.754 [0.477, 1.016]	0.766 [0.476, 1.033]
	$\hat{\tau}_\epsilon$	0.146 [0.021, 0.287]	0.15 [0.019, 0.295]	0.145 [0.025, 0.284]	0.151 [0.025, 0.294]
3 level	$\hat{\alpha}_1$	0.103 [-0.066, 0.259]	0.104 [-0.066, 0.259]	0.104 [-0.066, 0.259]	0.101 [-0.066, 0.259]
	$\hat{\beta}_1$	0.713 [0.469, 0.954]	0.717 [0.469, 0.954]	0.711 [0.469, 0.954]	0.714 [0.469, 0.954]
	$\hat{\tau}_\epsilon$	0.101 [0.007, 0.239]	0.098 [0.007, 0.239]	0.1 [0.007, 0.239]	0.101 [0.007, 0.239]
	$\hat{\alpha}_2$	-0.114 [-0.372, 0.121]	-0.112 [-0.376, 0.125]	-0.114 [-0.371, 0.112]	-0.108 [-0.366, 0.131]
	$\hat{\beta}_2$	0.535 [0.282, 0.778]	0.537 [0.278, 0.785]	0.537 [0.288, 0.774]	0.54 [0.287, 0.792]
	$\hat{\tau}_\delta$	0.212 [0.057, 0.415]	0.211 [0.056, 0.418]	0.21 [0.06, 0.414]	0.211 [0.058, 0.415]

Table 2.3: The posterior means and standard deviations for each of the model parameters resulting from fitting the 2-level (relapse on disability) and 3-level Bayesian surrogate outcome models with three different assumed values for sampling correlation ρ.

Chapter 3

Non-parametric Regression and Classification via Supervised Hierarchical Clustering

3.1 Introduction

Mechanisms for collecting massive amounts of empirical data are growing more common and sophisticated in many fields such as biology, medicine and public health. Advances in genomics and related high throughput technologies, the advent of electronic medical records and other developments are creating datasets that enable researchers to ask new questions. At the same time analyzing such data presents new challenges. One such challenge common to many data-driven fields involves finding interpretable patterns guided by very limited hypotheses about which mechanisms may be producing such patterns (e.g. which variables are important and how they are related to each other and to some outcome of interest). An example of this kind of problems comes from emergency care units where patients are carried in after all sorts of serious traumas. The conditions of the patients upon arrival in the ER, such as the kind of trauma or the amount of blood lost and their characteristics such as gender or age are all important predictors of their prognosis. The early diagnosis of the patient being at a higher risk of an adverse event such as the need for Massive Blood Transfusion (MBT) or death in the first 24 hours is crucial for decision making. The information available defines a quite complicated picture, often too difficult to decipher even for the physicians. In practice, as described in Nunez et al. (2009), indicators of particular conditions are combined to define scores that summarize the patient's overall status. Those scores provide an easy to compute one number summary, which is very valuable in an emergency setting, but provides a pretty limited description of the patterns that link the patients' conditions to the severity of their status.

The inability to make inference about a particular parameter of the data-generating distribution without strong assumptions on the underlying model is potential consequence of the so-called curse of dimensionality. An ambitious parameter of interest could be the joint distribution of all the variables, estimating the conditional distribution of some outcome given a large list of potential predictors. Depending on the goal of the estimation, there are different approaches to address the curse of dimensionality. One approach is to assume a relatively small statistical model. However these models typically rely on assumptions that are more a function of practicality than of well-established theory.

In the present article we follow the steps of the roadmap laid out in van der Laan and Rose (2011) to construct a flexible tool which provides an interpretable estimate of the conditional distribution of an outcome given high dimension predictors without relying on restrictive parametric assumptions. The roadmap is aimed at clearly defining the research question we are trying to answer by specifying the data structure we are dealing with, the statistical model their distribution belongs to and the parameter of this distribution that we are trying to estimate. As in the example from the intensive care units our data consists in a set of features, continuous, categorical or a mix of the two, and an outcome variable of some kind. This data is fully observed and i.i.d. from a non parametric distribution. Following the roadmap we define the parameter of interest as the expected loss minimizer according to an appropriate loss-function.

The parameter we are interested in is a function that links the joint distribution of the variables to the value of the outcome, so it makes sense to define this function in terms of a regression problem, linking particular regions of the feature space to a common value of the outcome. The estimation of the joint density of the variables is often addressed by constructing a partition of the space, trying to identify regions of high density and to separate different peaks or modes. One way to do this is with unsupervised learning methods such as clustering which group observations based on their similarity to each other with respect to a certain metric. This essentially reduces the potentially large vector of covariates to a single categorical variable, which assigns each point in the space to a group. In addition, if there is an outcome of interest, then a natural second step would be to examine how the distribution of outcomes varies by these this new grouping variable. In our trauma example, one could use the groupings created by clustering clinical and other variables related to patient heath, to predict the relevant outcomes, such as death. With unsupervised learning, groups can be found, but these groups are defined independently from of their ability to predict the outcome. The algorithm proposed here combines unsupervised clustering, used to generate a sequence of potential prediction models, and loss-based estimation (using cross-validation; van der Laan and Dudoit (2005)) to choose which of these candidates are the best at predicting the outcome.

Prediction problems are made harder when high number of features are measured, such as in gene expression studies. This high dimensionality can obscure the relatively small portion of variables that are actually related jointly to the outcome. Addressing this issue

properly might imply the need to define our loss function so that it penalizes models using too many features. We are in fact interested in finding a trade-off between parsimony and prediction error, specifically if a less complicated model results in a better "fit" than a more complicated one (say based on more variables and splits). In other settings we might be interested in a screening that throws out those variables whose inclusion doesn't imply a significant gain. For example in emergency settings such as the ER example introduced above we are interested in the simplest possible model that gives equivalent prediction to the best possible model. A well-known approach to variable selection is to introduce a penalty on the number of predictors in a linear regression problem, like in the well known lasso Tibshirani (1996). Lasso adds a penalty related to the number of features considered in a least squares problem in order to find the solution with the lowest number of non-zero coefficients. The parametric assumptions underlying this approach are a strong limitation. Tree-based methods on the other hand, are a non-parametric tool which addresses variable selection in a more flexible way. CART (Breiman et al. (1984)) iteratively splits the space using one feature at a time to identify a parsimonious set of predictors, making it a very useful approach to simultaneous prediction and dimensionality reduction. This procedure achieves the two-fold goal of constructing a parsimonious learner often well suited to certain application where decision trees are preferred. For instance, decision trees are widely used in medical literature because of their interpretability and parsimony, as described for example in Podgorelec et al. (2002) or in Kotsiantis (2013) and its references. Given our proposal for combining clustering with model selection based on the cross-validated risk estimate is a generalization of classification trees (Breiman et al. (1984)), We propose an approach to variable selection that borrows from procedures used to prune regression trees. We construct cluster-based trees iteratively adding variables to a working set and select the size of the working set which is optimal in terms of cross-validated risk wrt a proper loss function while constructing the candidate splits in terms of the distance between observations in the covariate space.

The paper is structured as follows. In Section 3.2 the general class of estimator defined by our algorithm is presented and framed in the context of histogram regression. In histogram regression the feature space is partitioned into regions and then the outcome is regressed on the indicator functions of those sets. We review a general framework for using loss-based, data-adaptive methods based on cross-validation for choosing candidate trees generated by clustering for use as the basis for histogram regression. We then discuss a specific implementation of this approach based on the hierarchical clustering routine, HOPACH van der Laan and Pollard (2003). In Section 3.3 the proposed methodology is presented and the details of the specific implementation are described. The functionalities of the R package are used to provide use cases and examples of how the method can be readily applicable in real settings. Results of a simulation study are presented in Section 3.4 to compare the performance of the proposed method to that of CART. In Section 3.5, the need for variable selection is discussed and an heuristic to address the problem is added to the algorithm. In Section 3.6, a real data application. We look at Trauma patients from Nunez et al. (2009), and construct a

parsimonious biomarker predicting adverse events. In Section 3.8 results are discussed and we present possible directions for future work.

3.2 Background on Histogram Regression and Loss-Based Estimation

The data are i.i.d. pairs , $\mathbf{O} = (Y, \mathbf{X}) \sim P_0$, where $P_0 \sim \mathcal{M}$ being $\mathcal{M}$ the space of models defineed by partitions of the feature space and $\mathbf{X}$ has dimension p. The components of $\mathbf{X}$ and Y can be continuous, categorical or a mix of the two. The class of parameters we consider is indexed by a partition of the covariate space R_k, $k = 1, ..., K$ which specify the regression function

$$\psi(P_0)(X) = \mathbb{E}[Y|\mathbf{X}] = \sum_{k=1}^{K} c_k \mathbb{1}\{\mathbf{X} \in R_k\} \tag{3.1}$$

The parameter of interest, given some specified loss-function, can be defined as the minimizer over the the specified family of regression functions, the one that minimizes the expected loss, i.e. $\psi_0(X) = \underset{\psi}{\text{argmin}} \mathbb{E}_P \mathcal{L}(Y, \mathbb{E}[Y|\mathbf{X}])$ (discussed in more detail in Section 3.2.1). The structure defined by equation 3.1 implies a dimensionality reduction, specially in cases in which $p \gg n$ and $K \ll p$. The high dimensionality of the data points is collapsed in the information contained in the class label. Many well known methods fall under the label of histogram regression, differing in the way the regions are constructed. For instance classification methods such as Support Vector Machines (SVM) Cortes and Vapnik (1995) or Linear Discriminant Analysis Fisher (1936) aim at splitting the feature space using some discriminant functions based on hyperplanes that separate regions where the outcome takes a different value. Tree based methods such as CART are also a kind of histogram regression which uses recursive binary splits to define the basis for the regression function. In the next Sections we will propose an alternative way to construct the basis for histogram regression based on hierarchically clustering the observations.

3.2.1 Loss-Based Estimation: Conceptual Framework

As opposed to the situation where $\mathcal{M}$ contains the true model, consider that it is a possibly misspecified estimating model. We still want to define a standard by which our procedure will be optimal with regard to competing candidates that are all based on partitions of covariate space. Loss-based estimation provides a framework for defining optimality in this context (van der Laan and Dudoit (2005)).

In order to estimate such a parameter our first step is to define a loss function L, such

that our parameter of interest is the minimizer of its the expectation, i.e. the risk minimizer

$$\psi_0 \equiv \underset{\psi}{\operatorname{argmin}} \mathbb{E}_{P_0} \mathcal{L}(\mathbf{O}, \psi)$$

Some common examples of loss functions are the squared error loss and -log-likelihood loss.

The next step is to define a finite collection of candidate estimators (a "sieve") that approximates the parameter space. Different classes can correspond to different statistical procedures indexed by the values of their tuning parameters, as the case of the Super (Machine) Learning Algorithm described in Van Der Laan et al. (2007). For each candidate estimator $\psi_n^{(k)}$ we can define the conditional risk as the random variable $\int \mathcal{L}(x, \psi_n^{(k)}) dP(x)$.

The risk minimization is translated into the data-driven selection of an optimal estimator, indexed by $\hat{k}$, that minimizes this conditional risk (that is, conditional on a particular fit, $\psi_n^{(k)}$, indexed by some size of model, e.g. the number of partitions of covariate space, k). Let $\tilde{k}$, the theoretical Oracle selector, be the minimizer of the *true* conditional risk over the space of candidates. The class indexed by the oracle selector is the one that minimizes the distance between the conditional risk and the minimum of the true risk (attained by the parameter of interest).

Using results from (ReF - Sandrine and Mark's 2005 paper), we define an estimator of ψ that is asymptotically equivalent (in fit) to the Oracle Selector. To do so, we divide the set of data into V equal folds, then iteratively using $V-1$ folds as a training set, producing estimates of the parameter of interest for each candidate. Those estimates are compared with the reserved data (validation set). The prediction error is averaged to produce an estimate of the conditional risk. Let $\hat{k}$ be the CV-selector, i.e. the procedure that selects the candidate estimator which minimizes the cross-validated risk:

$$\hat{k} = \underset{k}{\operatorname{argmin}} \mathbb{E}_{B_n} \int \mathcal{L}(x, \psi_n^{(k)}(P^0_{n,B_n})) dP^1_{n,B_n} = \mathbb{E}_{B_n} \frac{1}{n_1} \sum_{\{i:B_n(i)=1\}} \mathcal{L}(X_i, \psi_n^{(k)}(P^0_{n,B_n}))$$

being B_n the indicator of the training set. This procedure has a conditional risk asymptotically equivalent to the Oracle risk, and hence asymptotically optimal. The formal oracle result from van der Laan and Dudoit (2005) considers a loss function uniformly bounded and with second moment bounded by the first moment and proves that the CV estimator of the conditional risk wrt such a loss function converges at the Oracle's conditional risk with an error term $\mathcal{O}(\log K(n)/n)$. If the number of candidate estimators $K(n)$ is polynomial in sample size, then the cross-validation selector is asymptotically equivalent to the oracle selector or converges at rate $\mathcal{O}(\log n/n)$ with respect to $P_0\{\mathcal{L}(\psi) - \mathcal{L}(\psi_0)\}$. This means that one gets performance asymptotically equivalent to the Oracle selector, even if the number of candidates grows very large, so that for practical sample sizes, we can use the CV-risk among a large pool of candidates without overfitting.

3.3 Methods

In this section we introduce the general methods we use for building the partitions and splitting the space in Section 3.3.1, while in Section 3.2.1 we present the main ideas behind loss-based estimation. In Section 3.3.2 we fill in the details relative to the proposed algorithm and we present the implementation that makes it available in the R package HOPSLAM.

3.3.1 Grouping Observations and Splitting the Space

The flexibility in identifying convoluted patterns in the data depends strongly on the class of sets or metric adopted. For example the CART procedure splits the feature space based on a single variable at the time. The space is iteratively split in pairs of hyperplanes defined by an inequality based on the selected variable. In a two-dimensional setting this technique defines a set of rectangles in the feature space whose sides are parallel to one of the coordinate axis. Fig. 3.1 shows an example of the resulting partition.

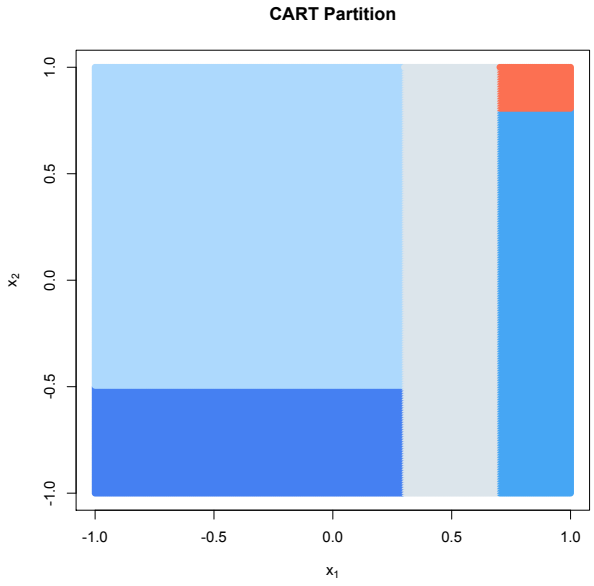

Figure 3.1: CART partitions the space using rectangles whose sides are parallel to the coordinate axes. This is defined by hierarchical sequence of splits involving one variable at the time. This class of sets is not very flexible in capturing elaborate patterns in the data. In this figure the sets are defined by the inequalities $\{X_1 < 0.3\} \cap \{X_2 < -0.5\}$, $\{X_1 < 0.3\} \cap \{X_2 > -0.5\}$, $\{X_1 > 0.3\} \cap \{X_1 < 0.7\}$, $\{X_1 > 0.7\} \cap \{X_2 < 0.8\}$ and $\{X_1 > 0.7\} \cap \{X_2 > 0.8\}$

We consider partitioning methods based on clustering (Kaufman and Rousseeuw (1990)),

and construct a richer class of partitions, which are functions of a distance (or similarity) matrix of all observations. Procedures such as PAM or K-Means define the clusters in terms of centroids and assign an observation to a cluster based on the proximity to the centroid of that cluster. Different kinds of metrics will result in very different shapes for the clusters (and thus ultimately different candidate partitions for our procedure to choose among). For example, Euclidean distance on the raw data makes sense when the variables are measured at the same scale and are (roughly) equivalent variables (e.g., gene expression measurements). However, cosine angle distance will compare the pattern of the features after rescaling them and the correlation distance will go even further by centering the values around their mean. The choice of the metric defines the characteristics of the features that are important in defining the groups (the pattern, scale and size, the scale and pattern only, etc.) and of defining the geometry of the sets in our partition as shown in Fig. 3.2.

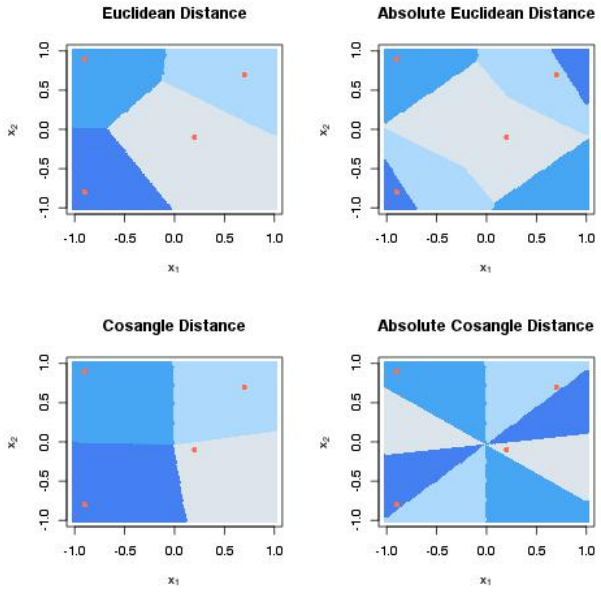

Figure 3.2: Different partitions obtained using different metrics. The partitions are defined as the points that are closer to one medoid than all the others. We pick the medoids to be the points $(-0.9, -0.8)$, $(0.7, 0.7)$, $(0.2, -0.1)$ and $(-0.9, 0.9)$. Different notions of distance define completely different partitions (represented by the color)

Some algorithms, such as PAM or K-means, require the number of groups as an input. There are several hierarchical clustering algorithms, based on divisive or agglomerative approaches. To generate the class of sets to construct our partition with we consider HOPACH, introduced in van der Laan and Pollard (2003). HOPACH is a hierarchical procedure built upon a clustering algorithm that requires a fixed number of clusters. In particular we choose PAM (Partitioning Around Medoids Kaufman and Rousseeuw (1990)) as the underlying clustering algorithm because it can work with any possible definition of distance and is therefore

very flexible in defining the partition. PAM is a clustering algorithm that groups the observations in a specified number of clusters by minimizing the sum of the distance of each observation from its class representative, called a medoid. Clusters are defined by the choice of the medoid, and then observations are assigned to the closest medoid, which defines the cluster. A heuristic search through possible medoids is performed, and the set chosen that result from optimizing a measure of how tight the observations are grouped within the same group and how far apart the clusters are.

One measure of how well cluster an observation is, is the silhouette score. For the i-th observation in the k-th cluster the silhouette contribution is

$$\begin{aligned} S(i) &= \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad i \in C_k \\ a_i &= \operatorname*{avg}_{j:x_j \in C_k} \delta(x_i, x_j) \quad \text{within cluster contribution} \\ b_i &= \min_{h \neq k} \Big\{ \operatorname*{avg}_{j:x_j \in C_h} \delta(x_i, x_j) \Big\} \quad \text{between cluster contribution} \end{aligned}$$

The mean of the silhouette score of each observation is a measure of how well the clusters split the data

HOPACH-PAM can be described as follows:

Initialize: Split the full data, using PAM, in a number of groups between 2 and K (a parameter to be appropriately tuned) and select the number of clusters k which maximizes the mean of the silhouette score. Order the clusters based on the distance between the respective medoids.

Iterate: For each cluster identified at the previous iteration perform the same splitting described above, choosing the optimal k based on the mean silhouette score, order the clusters and collapse some of them together if doing so improves the mean (average) silhouette score.

Main Clusters: through iterations we built a hierarchy of clusters. The final clustering solution is selected as the level that maximizes the mean silhouette score

HOPACH-PAM has three main features that provide more flexibility relative to other hierarchical procedures:

1. It can use any measure of distance: this implies a high flexibility in the way the space is partitioned, as shown in Figure 3.2, but also the possibility for combining heterogeneous features.

2. It doesn't use just binary splits.

3. It can collapse clusters allowing to correct a decision to split at a previous step. Standard hierarchical procedures are greedy algorithms, i.e. they choose the optimal split at the considered level, even though it's not always true that a sequence of local optimizations leads to a global optimum in this case.

A convenient feature of PAM is how the medoids around which the observations are clustered are observations themselves. In addition, being the pairwise distance is the only summary of the observations used by the algorithm, categorical, binary and continuos variables can be easily combined by defining an appropriate dissimilarity measure. For example we can define a distance that sums contributions from different components of a vector. In the case the vector is composed by categorical and continuous components we can use two different kind of dissimilarities and then simply sum their contributions. The same idea can be used to work with features which are on a different scale. We can either re-scale them in a pre-processing step, or define a weighted Manhattan distance that combines different contributions. Let J_{cat} be the set of categorical features in $\mathbf{X}$, and let J_{cont} be the set of continous features in $\mathbf{X}$. Then we can define the distance between $\mathbf{X}_i$ and $\mathbf{X}_{i'}$ as

$$\delta(\mathbf{X}_i, \mathbf{X}_{i'}) = w_{cat} \cdot \delta_{cat}(\mathbf{X}_{i,J_{cat}}\mathbf{X}_{i',J_{cat}}) + w_{cont} \cdot \delta_{cont}(\mathbf{X}_{i,J_{cont}}\mathbf{X}_{i',J_{cont}})$$

where δ_{cont} can be one of the distances shown in Fig. 3.2, δ_{cat} can be, for example, the indicator of $\{X_{i,j} = X_{i',j}\}$ and the w's are two weights that can be specified to give a different importance to the two contributions

A common use of cluster analysis is to group the observations and then interpret the clusters obtained based on subject matter knowledge (see, for example, Golub et al. (1999)). In this case clusters get "validated" by how well they match some grouping of the variables according to some feature not included in the clustering procedure (e.g. disease type, phylogenetic information, etc...). We formalize this idea to create a supervised learner that creates a set of partitions based on how well the observations cluster together and then uses the outcome variable to select the partition of the space which represents it better.

3.3.2 HOPSLAM: HOPACH-Pam Supervised Learning AlgorithM

In the previous sections we described how to construct a set of partitions of the feature space based on a clustering algorithm and how to select among a space of candidate estimators of a quantity of interest. In this section we are going to describe how to use those partitions to build our set of candidate learners and how to explore this set using V-fold cross-validation.

The HOPACH clustering algorithm defines a tree. This tree represents the hierarchy of splits, each node is a cluster and each level of this tree represents a partition of the space. If

we just consider different levels of the full HOPACH tree our candidate estimators are the subtrees with depth $1, 2, ...$ To each of them we associate a predicted value for the parameter of interest $\mathbb{E}[Y|\mathbf{X}]$ as in the histogram regression-classification setting, e.g. the average of a continuous outcome for the observations within each group. The groups defined by these trees are, however, a function of the silhouette score, which might improve down a branch even if splitting does not improve the estimate of $\mathbb{E}[Y|\mathbf{X}]$. We can think, for instance, of a region coherent in terms of Y but in which the distribution of the features has two or more modes. In this case HOPACH would split the region, because doing so would improve the silhouette score, but the splits would make the prediction model more complicated without providing more accurate (and maybe even less accurate) predictions of the outcome.

To make the space of candidate estimators richer without changing the nature of the partitions we want to have the flexibility to split only some regions of the space and we want to let the outcome to determine if a cluster is to be split or not. The levels of the full tree are the clustering solutions found letting the silhouette score determine the need for further splits. Once we have this set of clusters we can examine the outcome and collapse the groups together when doing so improves the fit. This is equivalent to considering as a space of candidate estimators the set of the subtrees in which only some branches are pruned at a varying depth. This can be done by pruning upward the full HOPACH tree along each branch instead of cutting the tree at a defined level. Now the set of possible trees includes the subtrees that we obtain by collapsing together the children of a node. Fig. 3.3 represents the candidate estimators under different definition of the splitting rule.

Pruning a tree is based upon a tradeoff between the parsimony of the model and the prediction error. Making the partition finer decreases the error (bias) in prediciton, but the gain might not be sufficient to justify the increase in complexity and thus variance of the prediction, resulting in an "over-fit" model. A common way to index trees obtained by pruning a larger partition tree (Breiman et al. (1984)) is to index the subtrees by a complexity parameter α that tunes the tradeoff between error and complexity. We define a measure analogous to the AIC, the Split Information Criterion (SIC). This measure, like the AIC, has a term related to the complexity and another linked to the goodness of fit based on the chosen loss function instead of the likelihood. The SIC is defined as

$$\text{SIC(partition)} = [\#\text{ elements in the partition}] + \sum_i \mathcal{L}(Y_i, \text{partition})$$

where $\hat{Y}_i(\text{partition})$ is the fitted value of the outcome associated to the chosen partition of the space. Given a full HOPACH tree we start from the leaves and go up every branch pruning the subtree and keeping the parent cluster every time the following condition holds

$$\text{SIC(parent)} \leq \sum \alpha \text{ SIC(children)}$$

where α is the parameter that indexes the space of trees. For $\alpha = 1$ we split whenever the inequality is strict.

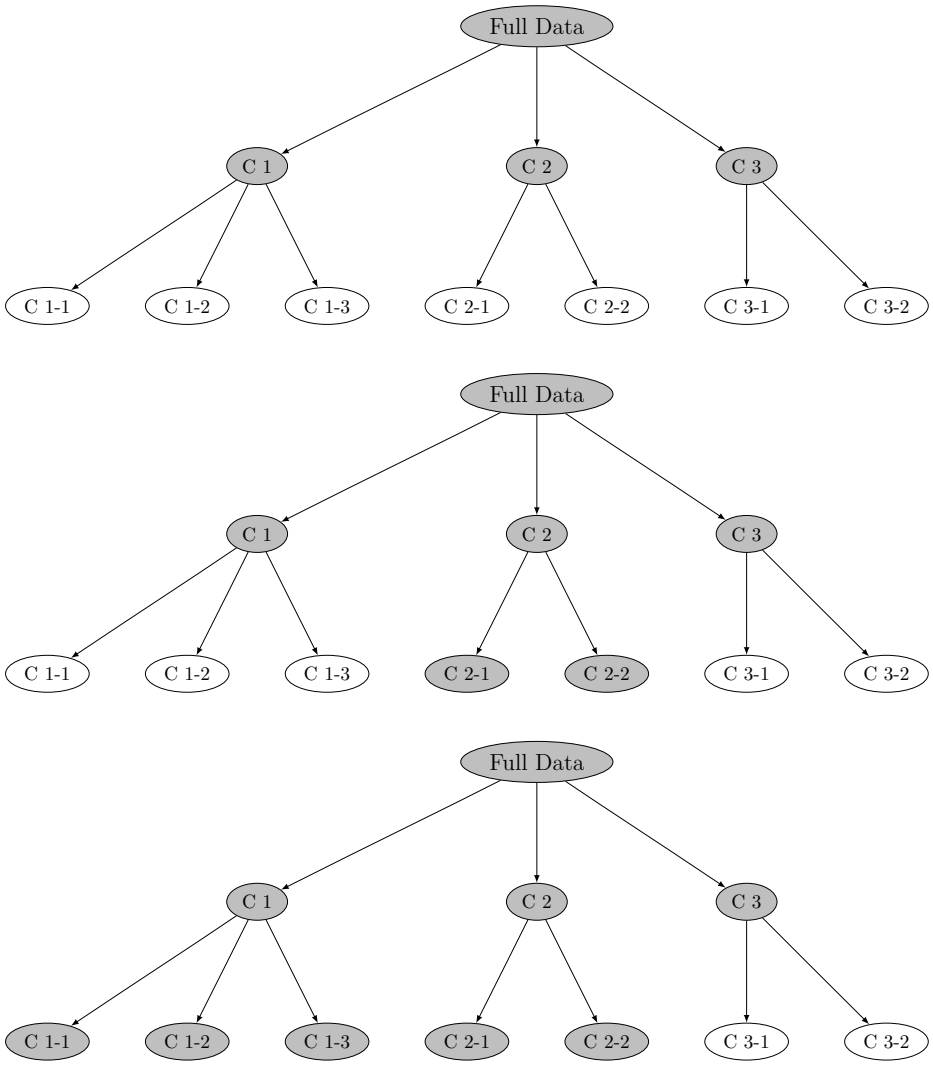

Figure 3.3: Three examples of possible partition subtrees on the same HOPACH tree. Greyed out are the subtrees whose leaves define the final partition. The first example simply considers the second level of the HOPACH tree as the final solution. The second and third trees, on the other hand, represent the case in which we prune along a branch and not across a specific level. In one case only Cluster 2 is split in its two children (Cluster 2 - 1 and Cluster 2 - 2), in the other case only the children of Cluster 3 are collapsed back together.

Now that we indexed with α the set of candidate estimators we can use cross-validation to choose its optimal value. The parameter α defines how aggressive is the pruning, and the goal of cross-validation is to estimate how such strategy generalizes well to new data. We partition the dataset into V (approximately) equal parts, set aside the v-th as a validation subset and use the rest of the data to build a full HOPACH tree. From this tree we can compute the range of α's that define all the possible subtrees. Namely, for each split there is a value of α for which the algorithm would prune the subtree: $\frac{SIC(\text{parent})}{SIC(\text{children})}$. For each value of α we have a different tree that defines a partition of the space to which we associate a prediction of the value of the outcome (e.g. the partition average). We can compare such prediction with the value of the outcome in the fold we left out using the loss function. This defines, for each fold, a piece-wise constant function of α. The CV selector α^* is the minimizer of the point wise average of those V functions. Algorithm 1 is a schematic representation of this procedure.

Input: Features $\mathbf{X}$, Outcome $\mathbf{Y}$, distance d
Output: Partition that minimizes the empirical risk and predicted value of the outcome for each element of the partition
for $v = 1 \to V$ **do**
  $tree \leftarrow \text{HOPACH}(\mathbf{X}_v^0, d)$;
  $\hat{\mathbf{Y}}(\alpha) \leftarrow \text{fit}(\text{prune}(tree, \alpha), \mathbf{Y}_v^0)$;
  $loss[v](\alpha) \leftarrow \mathbb{E}_n \mathcal{L}(\mathbf{Y}_v^1, \hat{\mathbf{Y}}(\alpha))$;
end
$Risk(\alpha) = \frac{1}{V} \sum_{v=1}^{V} loss[v](\alpha)$;
$\alpha^* = \underset{\alpha}{\text{argmin}}\ Risk(\alpha)$;
$Tree \leftarrow \text{prune}(\text{HOPACH}(\mathbf{X}, d), \alpha^*)$;
$\hat{\mathbf{Y}} \leftarrow \text{fit}(Tree, \mathbf{Y})$;

Algorithm 1: HOPSLAM$(d, \mathbf{X}, \mathbf{Y})$. The operator fit associates a common value of the outcome to each partition. X_v^0 is the v-th training set, X_v^1 the v-th validation set. $loss[v](\alpha)$ is the average of the losses, i.e. the empirical risk of the estimator built with the training set and applied to the validation set. The tree defined by the HOPACH clustering is a function of the distance matrix which is a function of the data $\mathbf{X}$ and of the chosen distance d.

The Algorithm we just described constructs a flexible and potentially interpretable partition of the feature space space. It is helpful to define the splits in terms of spacial grouping of the observations because it offers us insight on how similar the observations are to the medoid and between each other. Belonging to a cluster implies being similar to the other observations in the group. An important reason for collapsing adjacent regions when they are associated to a similar value of the outcome is the importance to control the complexity of the estimator. If splitting is not improving the prediction error enough to justify the increase in the number of groups our algorithm will penalize this choice and keep the ad-

jacent regions together. Since groups are defined in terms of proximity to the medoids the algorithm produces a quite interpretable tool. In medical settings such as the one in Nunez et al. (2009) we are interested in making a prognosis of an incoming patient based on initially collected clinical data. However, a procedure that defines too many patient groups carries too much information which becomes impractical for use by medical personnel.

3.4 Simulation Study

In this section the characteristics of the proposed algorithm are studied by simulation. We begin by explaining with simple examples the strengths and weaknesses of the proposed procedure in comparison with a standard recursive partitioning method. Then we evaluate the performance of the method in high dimensional settings with a larger simulation study.

The toy examples are designed to give an intuition of the strengths and differences of our method compared with CART. With the first toy examples we want to show how even a simple geometry is hard to reconstruct with rectangles such as those in Fig. 3.1. These examples help explaining the performance we observe in higher dimensional settings where a characteristic of CART is the choice of more groups than necessary, due to the relative lack of flexibility of the kind of splits used.

3.4.1 Simple setting

As a first example we chose the $[0, 10] \times [0, 10]$ square with a diagonal band. The outcome is set to one on the upper side of the the band delimited by the lines $y = x \pm 5$, to zero on the band and to -1 on the other side i.e. $Y|\mathbf{X} = \mathbb{1}\{X_2 > X_1 + 5\} - \mathbb{1}\{X_2 < X_1 - 5\}$. In the feature space data is simulated around nine points on the equally spaced grid $(\mu_1, \mu_2) \in \{-8, 0, 8\} \times \{-8, 0, 8\}$ with bivariate gaussian distribution (correlation zero and standard deviation 2.5) each one representing a mode in the feature distribution. A sample of 500 observations is generated. We fit HOPSLAM using euclidean distance clustering, 10 fold cross-validation and quadratic loss and CART using function rpart, from the homonymous package Therneau et al. (2012) fitting a CART regression tree. We pick such a geometry for the sake of a clear comparison of the two algorithms. Such a geometry is both simple enough to make clear the behavior of the two methods and quite realistic.

Fig. 3.4 shows a case where CART is not flexible enough to split the covariate space parsimoniously. If we look at the center of the right hand side panel we can see how CART uses small rectangles to partition that area. This area doesn't contain many observations, so those small regions are also fairly empty. This happens because the set of candidates used by CART is limited to orthogonal splits. Such splits do not easily capture linear decision

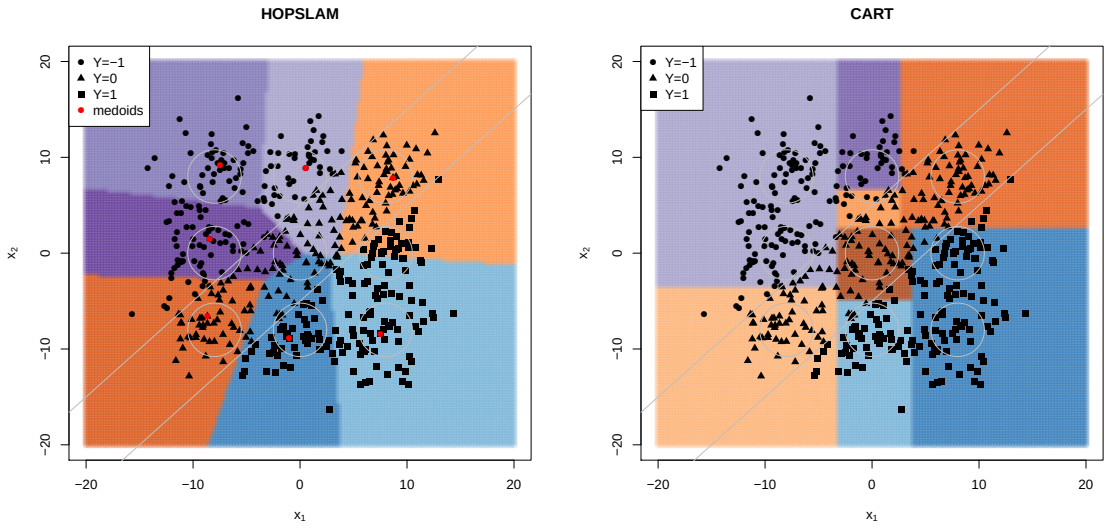

Figure 3.4: Predicted labels for CART and for HOPSLAM. The grey circles are centered around the centers the data is simulated around and the radius is equal to the standard deviation. Different colors identify different labels predicted by the algorithm. The shape of the dot represents the value of the outcome. We also use the color to represent the split of the space.

boundaries, as shown in Fig. 3.5. CART creates many small regions to approximate the boundaries between the groups. The consequence of this is that overfits the observed data and that those small regions will not capture well new observation.

The groups that CART identifies are not strongly related to the underlying spacial distribution of the observation, but just to the outcome. In practice CART uses the outcome variable to define regions in which it is homogeneous, while our procedure first identifies groups in terms of spacial coherency and then merges together the groups found to improve the coherency in terms of the outcome variable. This simple example shows when the HOPSLAM approach has the potential for much better prediction accuracy relative to CART. In this simple example the CV-risk of HOPSLAM is 0.14 while CART has a CV-risk of 0.2. This will be made clearer from the simulations in section 3.4.2.

One final remark can be made on the nature of the regions identified by the two methods. The regions HOPSLAM identifies can be described as the points that are similar to the respective medoid. This helps characterizing the groups and reducing the dimension of the population by picking few representative individuals. The regions defined by CART can be hard to interpret, such as the turquoise region in the middle of the bottom part of 3.4. HOPSLAM uses 7 groups to describe the joint distribution of the variables and predict the outcome, CART uses just one more group, but it's clear from Fig. **?** how they are inadequate in capturing the variability in the data.

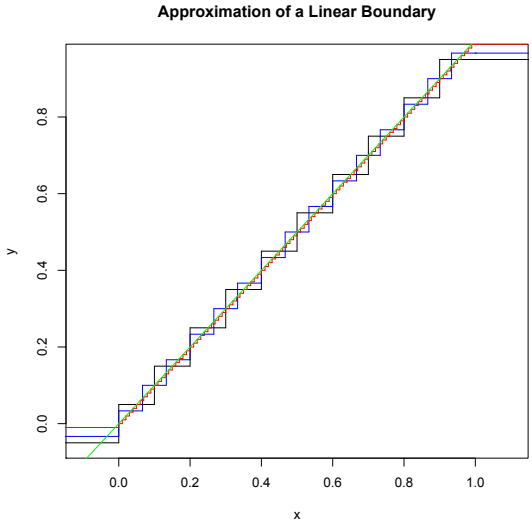

Figure 3.5: An example of stepfunctions approximating a linear decision boundary. A very high number of splits is necessary to achieve a good approximation of the stright line. HOPSLAM decision boundaries are defined as stright lines (hyperplanes in higher dimension) while CART decision boundaries are step functions (lines/hyperplanes parallel to one of the coordinate axis).

3.4.2 Relative Performance

We now look more in depth at the example from Fig. 3.4 and study the average behavior of the algorithm through repeated simulations. From the same distribution we draw 250 simulated datasets and compare HOPSLAM and CART. We simulate $n = 500$ points from a bivariate gaussian mixture with varying standard deviation (but zero correlation) and means on the grid $\{-8, 0, 8\} \times \{-8, 0, 8\}$. The Outcome is set to -1 above the band determined by the lines $x_2 = x_1 \pm 5$, to zero on the band and to one below the band.

In Fig. 3.6 we repeat 250 times the scenario in Fig. 3.4, where the standard deviation of the components of the mixture is set to 1 and 2. We can see how the behavior is very similar to Fig. 3.4, i.e. CART tends to require an higher number of groups and, partitioning in smaller regions, but paying a high price in terms of interpretability and over fitting achieving a higher MSE. In this scenario the true number of groups is 9. CART, in some simulations, uses up to 12 groups. As made clear in Fig. 3.4 most of these groups are tiny and very hardly generalizable to a new dataset. So we can see how CART pays an higher price in terms of MSE due to the high number of small group used to partition the space as shown in Fig. 3.6. This is relevant in applying the regions in a prediction setting: too small and ad hoc regions generalize poorly to a new dataset. Like in Fig. 3.4 many small regions are created and this generalizes very poorly as shown by the higher MSE. HOPSLAM, on the

other hand, has a better MSE and does a good job finding the true number of groups.

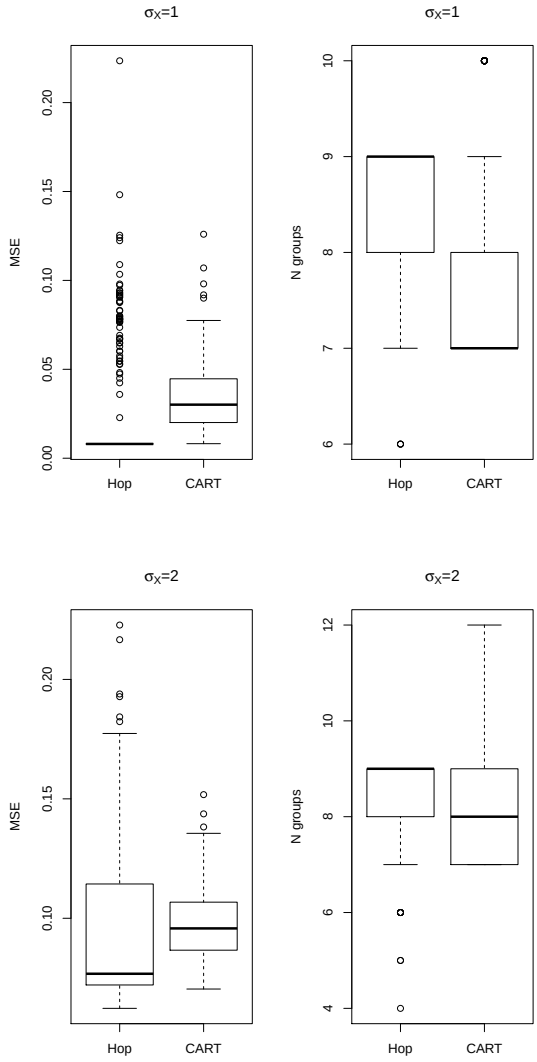

Figure 3.6: MSE and number of groups for the simulation study. The data is simulated from a 9 component gaussian mixture. The elements of the mixture have a diagonal covariance matrix with standard deviation 1 (top) and 2 (bottom) respectively. The Outcome is set to -1 above the band determined by the lines $x_2 = x_1 \pm 5$, to zero on the band and to one below the band.

3.4.3 Simulation under a True CART Model

To show the different use of the spacial distribution we make the setting even simpler. We simulate 1000 from a bivariate normal distribution centered at zero, with diagonal covariance matrix and unitary standard error. We then translate rigidly the observation with positive abscissa by 0.1 and those with negative abscissa by -0.1. The outcome is set to -1 on the left and to $+1$ on the right. In this case the information carried by the spacial density in the $\mathbf{X}$ space has nothing to do with the outcome. This makes this problem very easy for CART since a single vertical line perfectly splits the two classes where HOPSLAM uses three groups that still don't capture the geometry of the problem, for example the upper triangle in the left hand side panel.

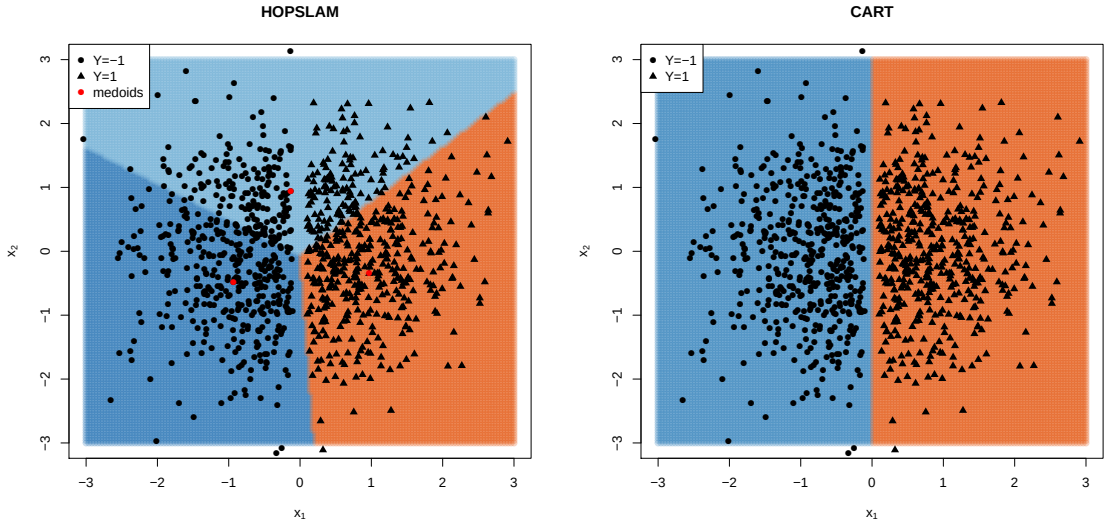

Figure 3.7: Predicted labels for CART and for HOPSLAM in the simple circular example. Data is generated around a bivariate gaussian centered at the origin with standard deviation 1 and no correlation. Points in the right half plane are shifted to the right by 0.1, points in the left half plane are shifted to the left by 0.1. The color of the dots is the predicted label, the shape the value of the outcome. The color is used to represent the split of the space.

The setting in 3.7 is, however, very unrealistic. It implies that a small change in the $\mathbf{X}$'s yields a sudden jump in the Y's. Also here the distribution of the feature has a high density peak exactly in the point of discontinuity for the Y's. Of course CART does better when it contains the true model, but as one can see, a model such as this simulation is not typically realistic.

In this example there is no relation between the density of the variables and the outcome. Most of your population is around zero, but the outcome changes dramatically between

positive and negative ordinate. This is unrealistic because is a very ill conditioned problem in which most of the population is clustered around the origin and a very small fluctuation in their coordinates make them switch condition. This might, in an applied seting, imply that we are not measuring an important variable. To give a simpler practical example, this situation would be anlogous to a setting in which being 1 pound above or below average explains an outcome such as getting cancer or not.

3.5 Variable Selection

Thus far, we have treated the set of predictors used to construct the clusters as fixed. However, in many settings, such as the gene expression studies mentioned in the introduction, we observe a high number of features, e.g. 10^4 or more gene expression values, and there can be significant improvements in both explanatory and predictive power by using only a small subset of the available variables. We propose a heuristic approach to variable selection as an add on to the general version of the algorithm. When considering subsets of variables, we run into the computationally infeasible problem of evaluating the full space of such subsets. A common solution in regression literature (see for example Narendra and Fukunaga (1977)) is to use a branch and bound approach based on the monotonicity of the residual sum of squares. Other very popular dimensionality reduction methods like lasso are also built upon a linear model and extensions. In these settings we know the error to increase when removing variables. It is the goal of such methods to find an appropriate tradeoff between the parsimony gained and the price payed in terms of accuracy. However, in our case a smaller number of variables might support a cleaner clustering solution and hence have a lower error. This means that we have to find an alternative to branch and bound to avoid exploring all the combinations of variables, we adopt one of the following heuristics:

Forward Starting from the leaves, i.e. the singletons with a single variable. Initialize the working set as the empty set.

1. While the cardinality of the working set is smaller than the maximum number of variables needed, do:
2. For each variable j not in the working set build the tree using the union between the working set and the variable j;
3. For each tree at the previous step compute the prediction error associated. Pick the variable whose inclusion yields the lowest error and add it to the working set.

Backward this approach starts from a working set with all the variables and iteratively removes one variable at a time until we removed the maximum number of variables needed.

In the rest of this paper, we will use forward selection.

We address the problem of selecting the optimal number of variables using the loss based estimation roadmap we described in Section 3.2.1. The set of candidate estimators contracted with the forward selection heuristic is indexed by the number of variables used in building the full tree and by the parameter α used to prune it. For each training fold we construct the full HOPACH tree using $k = 1, 2, ...$ variables. For each one of this trees we identify a set of α's as in the simple case described in Algorithm 1. The error is now a function of two variables, α and k. We then select the pair $(\alpha^*_{k^*}, k^*)$ for which the cross-validated risk is minimum. Algorithm 2 from Appendix A.3 describes this in a more schematic and detailed way.

3.5.1 Feature selection: a simulation study in higher dimension

An simulation study has been run for the variable selection. We extend the example presented in Section 3.4.2. Data is simulated by a nine component gaussian mixture in two dimensions and noise is introduced by adding an increasing number of noise features with mean zero. The simulations are ran for a number of noise features equal to 3 and 5. In Fig. 3.8 we can see from some of the results how HOPSLAM performs well in terms of dimensionality reduction, being able to efficiently reduce the dimensionality. We see how in Fig.3.8 we outperform CART in terms of MSE, while capturing the number of groups in the true data generating distribution and reducing the dimensionality considerably. CART's approach to dimensionality reduction makes it a very powerful tool, as shown in Fig.3.8 it is more aggressive in pruning noise features which results in a lower number of groups. Correctly reducing the dimensionality allows CART to work in a smaller space and hence to use less group. This strategy is, however, too aggressive, since CART throws away too many variables and pays a price in terms of MSE. Our method on the other hand keeps the MSE under control, being more conservative with the dimensionality reduction.

3.6 Real Data Application

As a real data application we consider the study from Nunez et al. (2009). The authors address the problem of predicting an adverse event, Massive Blood Transfusion (MBT), defined as requiring more than 10 Packed Red Blood Cells (PRBC) units within the first 24 hours. The authors review existing methods for clinical assessment and propose a score, ABC (Assessment of Blood Consumption), a number between 0 and 4, defined as the sum of four dichotomous components readily available at the patient's bedside: the presence of a penetrating mechanism, systolic blood pressure lower than 90 mmHg, heart rate higher than 120bpm and positive Focused Abdominal Sonography for Trauma (FAST). An ABC score larger than 2 is used as a predictor for an higher risk of adverse events. The authors

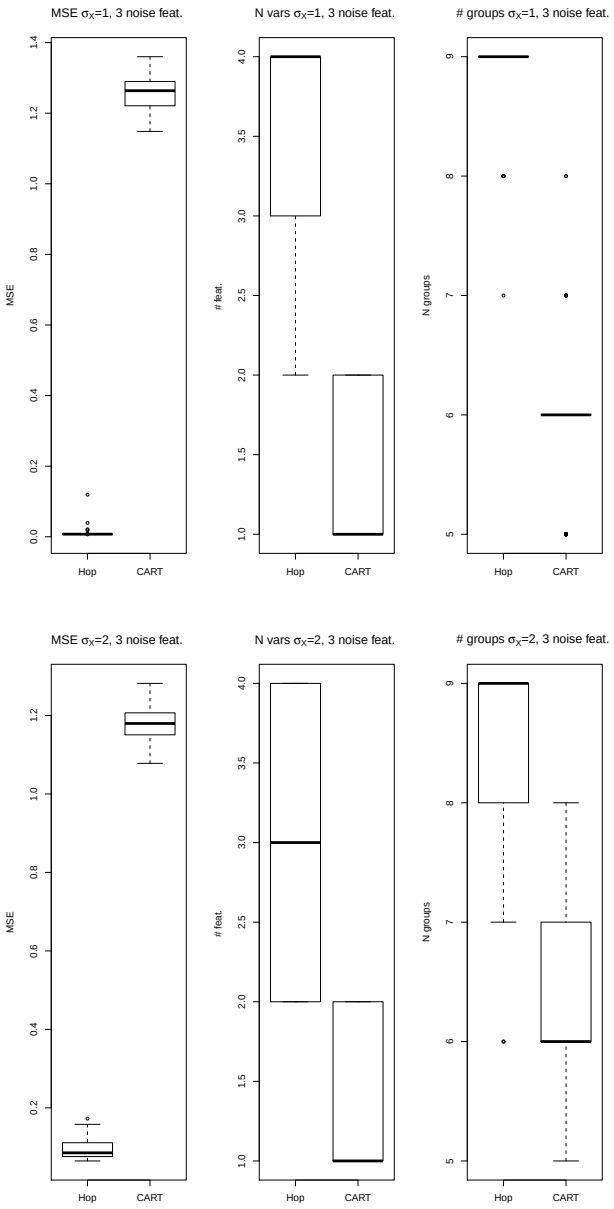

Figure 3.8: MSE, number of variables used and number of groups for the simulation study (100 simulations). The data is simulated from a 9 component gaussian mixture. The elements of the mixture have a diagonal covariance matrix with standard deviation 1 (top) and 2 (bottom) respectively. The Outcome is set to -1 above the band determined by the lines $x_2 = x_1 \pm 5$, to zero on the band and to one below the band. To the relevant variables 3 noise features are added.

explain how this measure can be evaluated very quickly upon patient arrival and how it is

very predictive of the subsequent conditions of the patient.

The data consist in a total of 513 patients with around forty variables describing their condition upon arrival at the Intense Care Unit (ICU). These variables describe the kind of trauma suffered, the conditions in which the patient is upon arrival and his conditions after the very first emergency procedures. The ABC score provides a very useful summary, helping the physicians to make quick decision without measuring too many variables.

In this case the result obtained using all the variables available is worse than the one given by a smaller subsets. In fact, the clustering solution obtained using all the measurements available has no relationship with the outcome variable. The MBT cases were almost evenly split between different clusters so the estimated outcome was, for all the clusters, non-MBT.

Even though we will use the four variables that define the ABC score to cluster the patient, we find useful to display the results in 2D plots representing the projection on the span of the first two principal components components, explaining 60% of the variability. This is a very common visualization technique, implemented, for example, for the PAM object in the R package clustering Maechler et al. (2013), that helps us to give a 2D representation of higher dimensional data.

From a preliminary exploration we can see how there is some area in the feature space where the patients at higher risk of MBT concentrate. Fig. 3.9 shows the projection of the data on the first two principal components of the ABC variables. Most of the observations are concentrated in the center, but we can see how the right hand side is mostly associated with MBT.

Using the sum of the four indicators is a quick solution to get a summary of the most important characteristics of the patient, but identifying specific patterns can add more granularity, differentiating between patients with the same ABC score. In Fig.3.10 we can see the comparison of the supervised clustering of the observations using a cosangle based dissimilarity with the results obtained with a threshold of 2 on the ABC score. We perform the supervised clustering on the same set of variables used to compute the ABC score The supervised partitioning has a cross-validation estimated misclassification rate of 0.11, while the procedure that selects the threshold on the ABC score that minimizes the misclassification rate has a CV estimated misclassification rate of 0.12. Also in this case CART's performances are not comparable with those of the other methods, with a CV estimated misclassification rate of 0.98 , as made clear by Fig. 3.10.

It is clear from the results how HOPSLAM captures the important features of the data and results in a similar classification as produced by experts working in trauma.

One of the streghts of HOPSLAM is the interpretability granted by the underlying PAM clustering. The medoids are the observations that best represent their group. In our case we can see themedoid in Tab. **??**

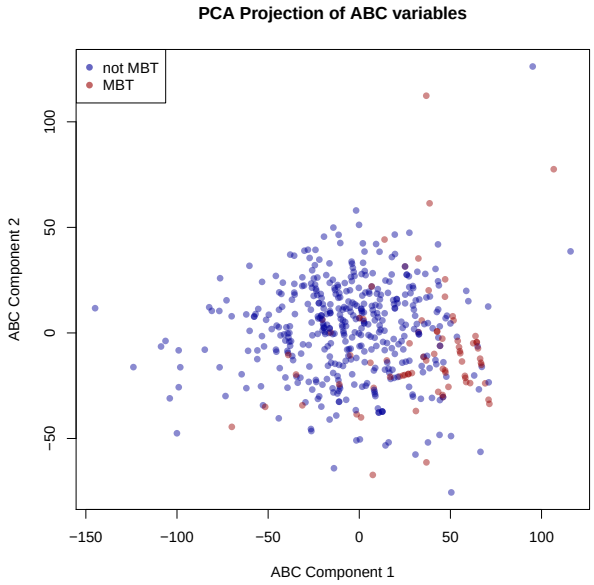

Figure 3.9: Projection on the first two principal components (60% variability explained) of the variables contribution to the ABC score. Red dots are patients who received a massive blood transfusion within the first 24 hours.

$\hat{Y}$	MOI	edbp	edhr	FAST
0	1	122	86	0
0	1	164	92	0
1	1	80	117	0
0	1	130	124	0

As we can see for all of them a penetrating mechanism is present, but what is more interesting is how a lower asystolic blood pressure characterizes the cluster associated with the positive predicted outcome. That cluster also has a higher heart rate than two of the other clusters. We can see in Fig. 3.11 how the variables are distributed across clusters

3.7 Software

We have made the algorithm described above available in the R package HOPSLAM. This provides a prototype implementation that can be applied to real data and provides estimation and visualization tools for fitting supervised partitions of the space. The core function of the package, HOPSLAM is currently based on HOPACH as an underlying clustering algorithm and builds upon the hopach function from Pollard et al. (2010) with similar syntax and

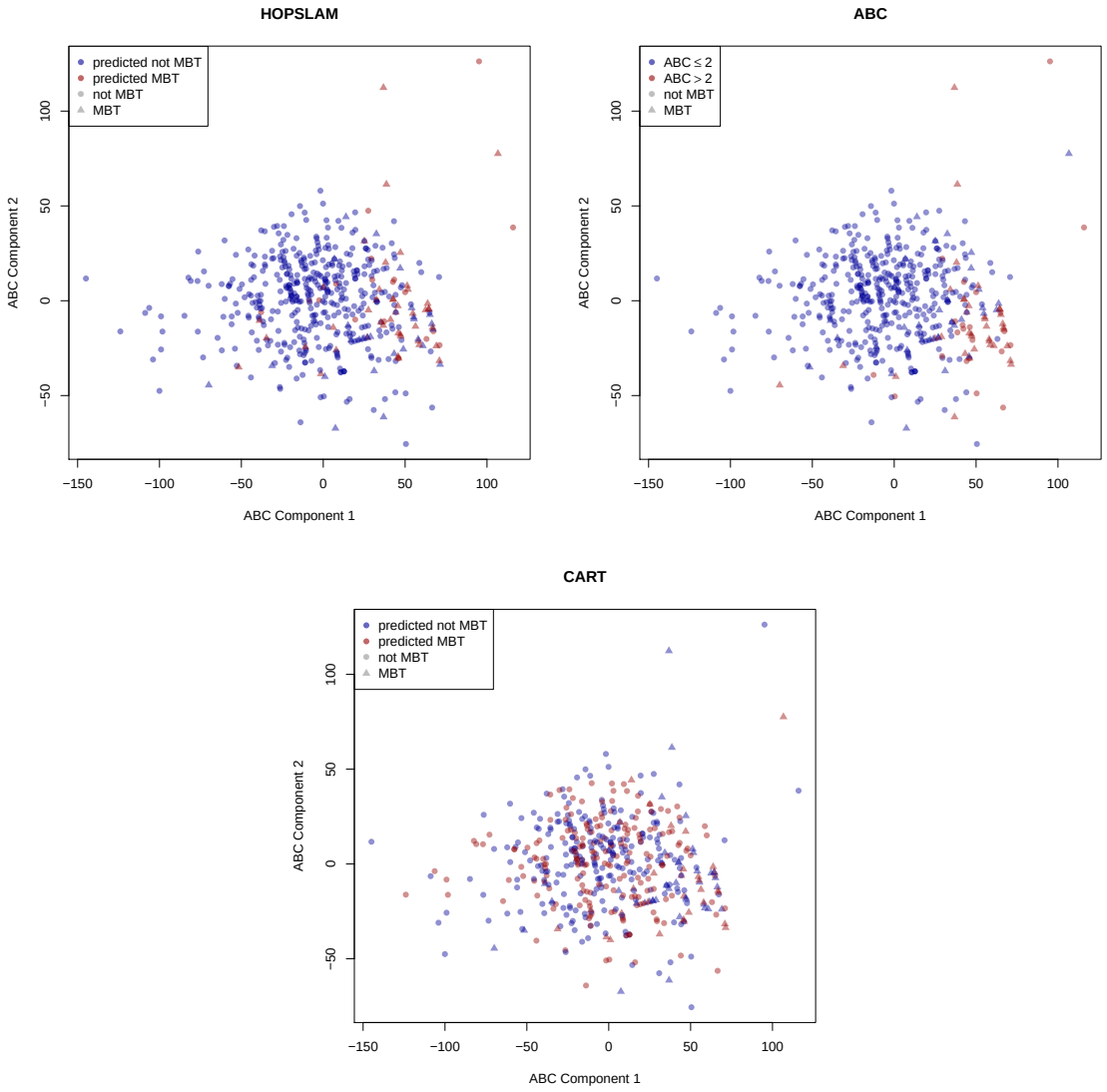

Figure 3.10: Comparison between thresholding the ABC score, classifying based on proximity on the ABC variables and CART. Notice how the variables are the same as in Fig. 3.9 but the color coding now represents the predicted label and the information on the outcome is conveyed by the shape of the dot.

behavior. Main inputs of the functions are the feature matrix X and the outcome vector Y and the number of folds for the cross-validation. All the distances available for hopach are also available for HOPSLAM, but users can specify their own choice by creating a function dissuser and a function vdissuser and specifying the option d="user". The first function will be used to compute the full distance metric, while the second just computes the distance vector of a point from a set of observations. Users also have the option to specify their own

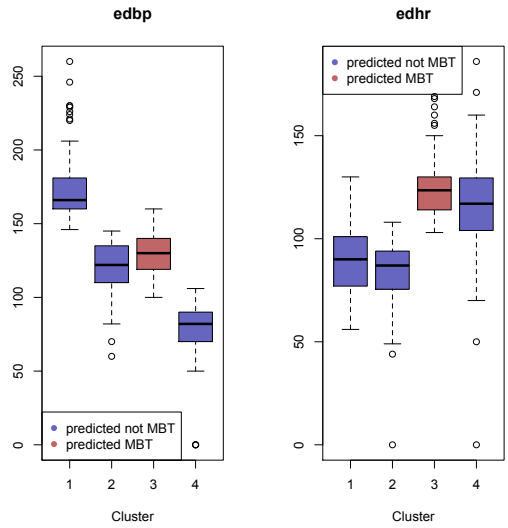

Figure 3.11: Boxplot for systolic blood pressure and heart rate by cluster. In the predicted MBT.

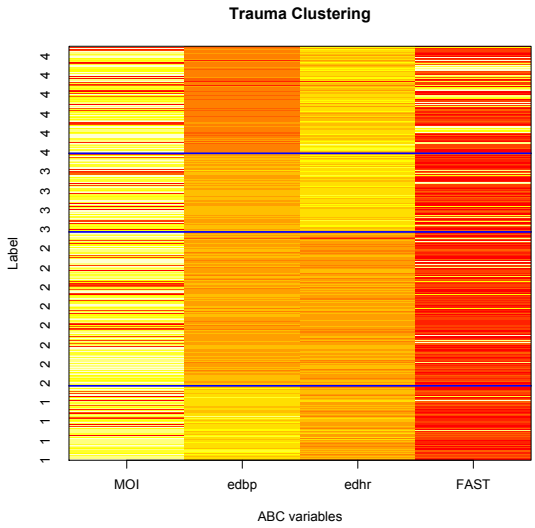

Figure 3.12: Heatmap of the ABC variables divided by cluster. We see a similar pattern to Fig. 3.11.

choice of fit and loss functions. The package includes the most common choices of an average fit, majority vote, squared error loss and misclassification error, but these functions are an example that the user can follow, if need be, to define something more exotic.

3.8 Discussion

In this article we present a novel method to use clustering to construct a space of candidate estimators of a non-parametric regression function. These estimators are constructed by using HOPSLAM clustering to define a hierarchy of splits. The selection of the optimal estimator is supported by the theory of loss-based estimation and by the optimality of the cross-validated selector that allows us to choose a candidate from this space with theoretical guarantees on the risk. The Oracle Inequality for the risk of the cross-validated selector gives an asymptotic bound on the performance of this procedure which, as the sample size increases, converges to the true risk minimizer. We show, via simulations and data analysis that our method performs better than standard CART trees in terms of mean squared error under many scenarios and it is more stable in the number of regions used to partition the space. The groups our method produces generalize well to new samples and are interpretable since they group the observations in terms of their pairwise similarity.

The present work shows how the trees built with the proposed procedure perform better than CART trees. However regression trees have known limitations (see, for example Hastie et al. (2001)) and they are often used as a base for building ensemble estimators. A natural extension of the present work is to study the performance of a forest of such trees, as described in **??**, in order to exploit the better performances on the single tree. This would, however, imply a loss in interpretability. The ideal ensemble method to improve the performance of our estimator would be gradient boosting Friedman (2000), which would allow to combine multiple trees without losing interpretability.

Both these extension rely on being able to fit the model multiple times and very fast. This is problematic since the current implementation has all the limitations that come from building on pre-existing code that has not been developed for being optimally reused many times. There are two layers in the current implementation: HOPSLAM calls hopach which calls pam. The function implementing PAM clustering has been written to be applied to a single dataset at the time, hence performs a series of checks and transformations of the data that get run every time the function is called. The same happens with the current implementation of HOPACH. The goal of both functions is to cluster efficiently a single dataset performing all the operations necessary to make this as user-friendly as possible. Making the current implementation more efficient will require to slim down these operations and to remove all the "bells and whistles". The current implementation is, however, a useful prototype that allows to tackle medium sized datasets and makes the method available to be applied to real problems.

Bibliography

Akacha, M. and Benda, N. (2010). The impact of dropouts on the analysis of dose-finding studies with recurrent event data. *Statistics in Medicine*, 29:1635–1646.

Berry, D. (2006). Bayesian clinical trials. *Nature Reviews Drug Discovery*, 5:27–36.

Berry, D. A., Mueller, P., Grieve, A. P., Smith, M. K., Parke, T., and Krams, M. (2002). Bayesian designs for dose-ranging drug trials. *Gatsonis, C., Kass, R. E., Carlin, B., Carriquiry, A., Gelman, A., Verdinelli, I., West, M., eds. Case Studies in Bayesian Statistics*, 5:99–181.

Berry, S., Carlin, B., Lee, J., and Müller, P. (2010). *Bayesian Adaptive Methods for Clinical Trials*. CRC Press.

Bornkamp, B., Bretz, F., Dmitrienko, A., Enas, G., Gaydos, B., Hsu, CH. König, F., Krams, M., Liu, Q., Neuenschwander, B., Parke, T., Pinheiro, J., Roy, A., Sax, R., and Shen, F. (2007). Innovative approaches for designing and analyzing adaptive dose-ranging trials. *Journal of Biopharmaceutical Statistics*, 17:965–995.

Brannath, W., Zuber, E., Branson, M., Bretz, F., Gallo, P., Posch, M., and Racine-Poon, A. (2009). Confirmatory adaptive designs with Bayesian decision tools for a targeted therapy in oncology. *Statistics in Medicine*, 28:1445–1463.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks/Cole, Monterey, CA.

Bretz, F., Koenig, F., Brannath, W., Glimm, E., and Posch, M. (2009). Adaptive designs for confirmatory clinical trials. *Statistics in Medicine*, 28:1181–1217.

CHMP (2007). Committee for medicinal products for human use: Reflection paper on methodological issues in confirmatory clinical trials planned with an adaptive design.

Cook, R., Bergeron, P., Boher, J., and Liu, Y. (2009). Two-stage design of clinical trials involving recurrent events. *Statistics in Medicine*, 28:2617–2638.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Daniels, M. and Hughes, M. (1997). Meta-analysis for the evaluation of potential surrogate markers. *Statistics in Medicine*, 16(17):1965–1982.

Doucet, A. and Johansen, A. (2008). A tutorial on particle filtering and smoothing: fifteen years later. *Tech. Report U.B.C.*

Eichler, H., Abadie, E., Breckenridge, A., Leufkens, H., and Rasi, G. (2012). Open clinical trial data for all? a view from regulators. *PLoS Medicine*, 9(4):e1001202.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Friede, T. and Schmidli, H. (2010a). Blinded sample size reestimation with count data: methods and applications in multiple sclerosis. *Statistics in Medicine*, 29:1145–1156.

Friede, T. and Schmidli, H. (2010b). Blinded sample size reestimation with count data: methods and applications in multiple sclerosis. *Statistics in Medicine*, 29(10):1145–1156.

Friede, T. and Schmidli, H. (2010c). Blinded sample size reestimation with negative binomial counts in superiority and non-inferiority trials. *Methods of Information in Medicine*, 49:618–624.

Friede, T. and Schmidli, H. (2010d). Blinded sample size reestimation with negative binomial counts in superiority and non-inferiority trials. *Methods Inf Med*, 49(6):618–24.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Gail, M., Pfeiffer, R., van Houwelingen, H., and Carroll, R. (2000). On meta-analytic assessment of surrogate outcomes. *Biostatistics*, 1(3):231–246.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(33):515–533.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537.

Grieve, A. and Krams, M. (2005). ASTIN: a Bayesian adaptive dose-response trial in acute stroke. *Clinical Trials*, 2:340–351.

Gsteiger, S., Neuenschwander, B., Mercier, F., and Schmidli, H. (2013). Using historical control information for the design and analysis of clinical trials with overdispersed count data. *Statistics in Medicine*.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc.

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999). Bayesian Model Averaging: a tutorial (with discussion). *Statistical Science*, 14:382–417.

Hung, H., Wang, S., and O'Neill, R. (2011). Flexible design clinical trial methodology in regulatory applications. *Statistics in Medicine*, 30:1519–1527.

Jiang, H. and Wong, W. (2009). Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics*, 15:1026–1032.

Kappos, L., Antel, J., Comi, G., Montalban, X., O'Connor, P., Polman, C., Haas, T., Korn, A., Karlsson, G., and Radue, E. (2006). FTY720 D2201 study group. oral Fingolimod (FTY720) for relapsing multiple sclerosis. *New England Journal of Medicine*, 355:1124–1140.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

Keene, O., Jones, M., Lane, P., and Anderson, J. (2007a). Analysis of exacerbation rates in asthma and chronic obstructive pulmonary disease: example from the TRISTAN study. *Pharmaceutical Statistics*, 6:89–97.

Keene, O., Jones, M. R., Lane, P., and Anderson, J. (2007b). Analysis of exacerbation rates in asthma and chronic obstructive pulmonary disease: example from the tristan study. *Pharmaceutical Statistics*, 6(2):89–97.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283.

Lunn, D., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS - a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2013). *cluster: Cluster Analysis Basics and Extensions*.

Mercier, F., Schmidli, H., and Kappos, L. (2009). Using the negative binomial model to assess the distribution of mri responses to treatment: modelling results from a phase II study of oral fingolimod (FTY720) in relapsing multiple sclerosis. *Multiple Sclerosis*, 15:S221–S211.

Molenberghs, G., Burzykowski, T., Alonso, A., Assam, P., Tilahun, A., and Buyse, M. (2010). A unified framework for the evaluation of surrogate endpoints in mental-health clinical trials. *Statistical Methods in Medical Research*, 19(3):205–236.

Molenberghs, G., Buyse, M., Geys, H., Renard, D., Burzykowski, T., and Alonso, A. (2002). Statistical challenges in the evaluation of surrogate endpoints in randomized trials. *Controlled Clinical Trials*, 23(6):607–625.

Molenberghs, G., Verbeke, G., and Demétrio, C. (2007). An extended random-effects approach to modeling repeated, overdispersed count data. *Lifetime Data Analysis*, 13:513–531.

Narendra, P. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C–26(9):917–922.

Neuenschwander, B., Capkun-Niggli, G., Branson, M., and Spiegelhalter, D. (2010). Summarizing historical information on controls in clinical trials. *Clinical Trials*, 7:5–18.

Normand, S., Wang, Y., and Krumholz, H. (2007). Assessing surrogacy of data sources for institutional comparisons. *Health Services and Outcomes Research Methodology*, 7(1):79–96.

Nunez, T. C., Voskresensky, I. V., Dossett, L. A., Shinall, R., Dutton, W. D., and Cotton, B. A. (2009). Early prediction of massive transfusion in trauma: Simple as abc (assessment of blood consumption)? *The Journal of TRAUMA Injury, Infection, and Critical Care*, 66(2):346–352.

Ohlssen, D. and Racine, A. ((Under Revision)). A flexible Bayesian approach for modeling monotonic dose-response relationships in clinical trials with applications in drug development.

Podgorelec, V., Kokol, P., Stiglic, B., and Rozman, I. (2002). Decision trees: an overview and their use in medicine. *Journal of Medical Systems*, 26(5):445–463.

Pollard, K. S., van der Laan, M. J., and Wall, G. (2010). *hopach: Hierarchical Ordered Partitioning and Collapsing Hybrid (HOPACH)*. R package version 2.20.0.

Polman, C., Reingold, S., Barkhof, F., Calabresi, P., Clanet, M., Cohen, J., Cutter, G., Freedman, M., Kappos, L., Lublin, F., McFarland, H., Metz, L., Miller, A., Montalban, X., O'Connor, P., Panitch, H., Richert, J., Petkau, J., Schwid, S., Sormani, M., Thompson, A., Weinshenker, B., and Wolinsky, J. (2008). Ethics of placebo-controlled clinical trials in multiple sclerosis: a reassessment. *Neurology*, 25(70):1134–40.

Prentice, R. (1989). Surrogate endpoints in clinical trials: definition and operational criteria. *Statistics in Medicine*, 8(4):431–440.

Qin, L. (2011). Magnetic resonance imaging lesion count as a surrogate endpoint in relapsing-remitting multiple sclerosis clinical trials. Master's thesis, University of British Columbia.

Riley, R. (2009). Multivariate meta-analysis: the effect of ignoring within study correlation. *Journal of the Royal Statistical Society, Series A*, 172(4):789–811.

Riley, R., Abrams, K., Sutton, A., Lambert, P., and Thompson, J. (2007). Bivariate random-effects meta-analysis and the estimation of between-study correlation. *BMC Medical Research Methodology*, 7(1):1–15.

Schmidli, H., Bretz, F., and Racine, A. (2007). Bayesian predictive power for interim adaptation in seamless phase ii/iii trials where the endpoint is survival up to some specified timepoint. *Statistics in Medicine*, 26:4925–4938.

Schmidli, H., Bretz, F., Racine, A., and Maurer, W. (2006). Confirmatory seamless phase II/III clinical trials with hypotheses selection at interim: Applications and practical considerations. *Biometrical Journal*, 48:635–643.

Sheiner, L. (1997). Learning versus confirming in clinical drug development. *Clinical Pharmacological Therapy*, 61:275–291.

Smith, A. and Gelfand, A. (1992). Bayesian statistic without tears: a sampling-resampling perspective. *The American Statistician*, 46:84–88.

Sormani, M., Bonzano, L., Roccatagliata, L., Mancardi, G., Uccelli, A., and Bruzzi, P. (2010). Surrogate endpoints for EDSS worsening in multiple sclerosis. *Neurology*, 75(4):302–309.

Sormani, M., Bruzzi, P. Miller, D., Gasperini, C., Barkhof, F., and Filippi, M. (2009). Modeling MRI enhancing lesion counts in multiple sclerosis using a negative binomial model: implications for clinical trials. *Journal of Neurological Sciences*, 163:382–401.

Sormani, M., Bruzzi, P., Rovaris, M., Barkhof, F., Comi, G., Miller, D., Cutter, G., and Filippi, M. (2001a). Modelling new enhancing MRI lesion counts in multiple sclerosis. *Multiple sclerosis*, 7(5):298–304.

Sormani, M., Li, D., Bruzzi, P., Stubinski, B., Cornelisse, P., Rocak, S., and De Stefano, N. (2011). Combined MRI lesions and relapses as a surrogate for disability in multiple sclerosis. *Neurology*, 77:1684–1690.

Sormani, M., Miller, D., Comi, G., Barkhof, F., Rovaris, M., Bruzzi, P., and M., F. (2001b). Clinical trials of multiple sclerosis monitored with enhanced MRI: new sample size calculations based on large data sets. *Journal of Neurosurgical Psychiatry*, 70:494–499.

Spiegelhalter, D., Abrams, K., and Myles, J. (2003). *Bayesian Approaches to Clinical Trials and Health-Care Evaluation*. Wiley, Chichester.

Spiegelhalter, D., Best, N., Carlin, B., and van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society Series B*, 64:1–34.

Steinbrook, R. (2013). The european medicines agency and the brave new world of access to clinical trial databrave new world of access to clinical trial data. *JAMA internal medicine*, 173(5):373–374.

Tan, H., Gruben, D., French, J., and Thomas, N. (2011). A case study of model-based Bayesian dose response estimation. *Statistics in Medicine*, 30:2622–2633.

Temple, R. (1999). Are surrogate markers adequate to assess cardiovascular disease drugs? *JAMA: The Journal of the American Medical Association*, 282(8):790–795.

Therneau, T., Atkinson, B., and Ripley, B. (2012). *rpart: Recursive Partitioning*. R package version 4.1-0.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288.

Trapnell, C., Williams, B., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M., Salzberg, S., Wold, B., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28:511–515.

Turner, R., Omar, R., and Thompson, S. (2006). Modelling multivariate outcomes in hierarchical data, with application to cluster randomised trials. *Biometrical journal*, 48(3):333–345.

U.S. Food and Drug Administration (2010). Draft guidance for industry on adaptive design clinical trials for drugs and biologics.

van den Elskamp, I., Knol, D., Uitdehaag, B., and Barkhof, F. (2009a). The distribution of new enhancing lesion counts in multiple sclerosis: further explorations. *Multiple Sclerosis*, 15:42–49.

van den Elskamp, I., Knol, D., Uitdehaag, B., and Barkhof, F. (2009b). The distribution of new enhancing lesion counts in multiple sclerosis: further explorations. *Multiple Sclerosis*, 1(15):42–49.

van der Laan, M. and Dudoit, S. (2005). Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. *Statistical Methodology*, 117(1):275–303.

van der Laan, M. and Pollard, K. (2003). Hybrid clustering of gene expression data with visualization and the bootstrap. *Journal of Statistical Planning and Inference*, 117(1):275–303.

van der Laan, M. and Rose, S. (2011). *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Series in Statistics.

Van Der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1):–.

van Houwelingen, H., Arends, L., and Stijnen, T. (2002). Advanced methods in meta-analysis: multivariate approach and meta-regression. *Statistics in Medicine*, 21(4):589–624.

Weir, C. and Walley, R. (2006). Statistical evaluation of biomarkers as surrogate endpoints: a literature review. *Statistics in Medicine*, 25(2):183–203.

Wolinsky, J. and Beck, C. (2011). The long march to surrogates of meaningful clinical outcomes in MS trials: are we there yet? *Neurology*, 77(18):1658–1659.

World Health Organization (2004). Neurology atlas.

Y.C., W., Meyerson, L., Tang, Y., and Qian, N. (2009). Statistical methods for the analysis of relapse data in MS clinical trials. *J Neurol Sci*, 285(1-2):206–11.

Yin, G. and Yuan, Y. (2009). Bayesian Model Averaging continual reassessment method in phase I clinical trials. *JASA*, 104:954–968.

Appendices

Appendix A

Details of the Algorithms

This appendix aims to describe the algorithmic details that have been omitted from the main corpus of the dissertation not to distract the reader. Section A.1 describes the algorithm used in Chapter 1 and explains the computational approach to the computation of the predictive probabilities.

Section A.2 describes the data extraction procedure used in the meta-analsis from Chapter 2.

A.1 Theoretical and Computational Details of Chapter 1

A.1.1 Programming Approach to the Computation of Predictive Quantities

The first step of the analysis conducted with WinBUGS, is the analysis of the interim data in order to compute the posterior distribution of the parameters; the predictive stage is then carried on with *ad hoc* R code in order to cope with the computational intensity of the task. In both cases the analysis carried on at the end of the study will be performed with WinBUGS.

To summarize our "predictive" pipeline is composed by three stages:

1. Approximate the posterior distribution $p(\theta|Y)$ of the parameters given the interim data (using WinBUGS);

2. Using the above sample simulate possible outcomes for stage 2 in order to compute the predictive quantities to be used for interim decision making (using dedicate R code) as described in the next section;

3. If any allocation has been chosen at the previous step run the post interim part of the trial and then assess the performances of the drug using the approximation of $p(\theta|Y, Y^*)$ obtained from the post interim data (using again WinBUGS).

A.1.2 The Sampling-Importance Resampling Algorithm: General Framework

The Sampling-Importance Resampling technique (SIR from now on), also called Bayesian bootstrapping, has been introduced in Smith and Gelfand (1992) and widely used in the field of computer vision and particle filtering, as in Doucet and Johansen (2008), and in genomics as for example in Trapnell et al. (2010) and Jiang and Wong (2009). Suppose we are able to sample θ_i $(i = 1, ..., N)$ easily from a distribution $g(\theta)$ but we are interested in sampling from $h(\theta) = f(\theta)/\int f(\theta')d\theta'$. For each sample θ_i from g one calculates $q_i = f(\theta_i)/g(\theta_i)$ and the normalized weight $w_i = q_i/\sum_{j=1}^{N} q_j$. Sample one θ^* from the θ_j, $j = 1, ..., N$ with probability w_j. This new sample is approximately distributed according to h, and the approximation becomes more precise with increasing N.

This approach has an application in a Bayesian context: take $f(\theta) = l(\theta; x) \cdot p(\theta)$, where $l(\theta; x)$ is the likelihood and $p(\theta)$ is the prior. As described in the above section we can easily draw a sample from $\theta \sim g(\theta) = p(\theta)$ but we need to sample from the posterior distribution $h(\theta) = p(\theta|x)$. Since $p(\theta|y) \propto l(\theta; x) \cdot p(\theta)$ apply SIR one draws samples with the resample weight given by

$$w_i = \frac{l(\theta_i; x)}{\sum_j l(\theta_j; x)}$$

for obtain m samples $\theta^* \sim p(\theta|x)$.

A.1.3 Application of SIR to Bayesian Dose-Finding

In our dose-finding context, let $\theta = (\boldsymbol{\alpha}, \beta)$ be sampled from the prior $p(\theta)$ for stage 2 which is the interim posterior, i.e. $p(\theta) = p(\theta|Y)$. We are interested in approximating the integrals in (1.4) by averaging over a number of possible outcomes Y^* in stage 2.

To do so, using WinBUGS select a posterior sample $(\boldsymbol{\alpha}, \beta)^{(1)}, ..., (\boldsymbol{\alpha}, \beta)^{(k)}, ..., (\boldsymbol{\alpha}, \beta)^{(N)}$. To simulate future outcomes in stage 2, draw a pair $(\boldsymbol{\alpha}, \beta)^{(l)}$ from this sample, use this parameter to simulate one post-interim posterior dataset $Y_d^{*(l)}$, where d is the dose to be explored, future placebo and Dose 5 response are also included in this data set.

$p\big(Y_d^{*(l)}|(\boldsymbol{\alpha},\beta)^{(k)}\big)$ is computed $k=1,...,N$. The resample weights are given by

$$w_k = \frac{l(\theta_k;y)}{\sum_j l(\theta_j;y)} = \frac{p(Y_d^{*(l)}|(\boldsymbol{\alpha},\beta)^{(k)})\cdot p((\boldsymbol{\alpha},\beta)^{(k)})}{\sum_j p(Y_d^{*(l)}|(\boldsymbol{\alpha},\beta)^{(j)})\cdot p((\boldsymbol{\alpha},\beta)^{(j)})} = \frac{p(Y_d^{*(l)}|(\boldsymbol{\alpha},\beta)^{(k)})}{\sum_j p(Y_d^{*(l)}|(\boldsymbol{\alpha},\beta)^{(j)})}$$

being the interim posterior sample uniformly distributed with probability $p((\boldsymbol{\alpha},\beta)^{(j)}) = 1/N$, $\forall j = 1,..N$.

Whether a sample $(\boldsymbol{\alpha},\beta)^{(k)}$ satisfies a decision criterion can be checked directly. We then obtain

$$\begin{aligned}\mathbb{P}\{\text{satisfy criterion}|Y^*\} &= \int_{\mathbb{R}_+^2} \mathbb{P}\{\text{satisfy criterion}|(\boldsymbol{\alpha},\beta),Y^*\}\mathbb{P}\{(\boldsymbol{\alpha},\beta)|Y^*\}\mathrm{d}(\boldsymbol{\alpha},\beta) \approx \\ &\approx \frac{1}{N}\sum_{k=1}^{N} \mathbb{1}\{(\boldsymbol{\alpha},\beta)^{(k)} \text{ satisfies the criterion}\}\cdot w_k = PP_d^{(l)}[\text{criterion}]\end{aligned}$$

We repeat this procedure for l going up to a very large number, say L,

$$PP_d = \frac{1}{L}\sum_{l=1}^{L} \mathbb{1}\Big\{\bigcap_{\{\text{criteria}\}} \{PP_d^{(l)}[\text{criterion}] > c\}\Big\}$$

is then an approximation of (1.4).

To summarize, our algorithm is as follows

1. Select Dose d;
2. sample $(\boldsymbol{\alpha},\beta)^{(1)},...,(\boldsymbol{\alpha},\beta)^{(k)},...,(\boldsymbol{\alpha},\beta)^{(N)}$ from the post interim posterior (with WinBUGS);
3. draw $(\boldsymbol{\alpha},\beta)^{(l)}$ from the posterior sample at interim of size N;
4. simulate one dataset $Y_d^{*(l)}|(\boldsymbol{\alpha},\beta)^{(l)}, A_d$, with A_d as defined above;
5. apply importance resampling (using R):
   - compute $p\big(Y_d^{(l)}|(\boldsymbol{\alpha},\beta)^{(k)}\big),\ k=1,...,N$;
   - compute w_k;
   - compute $PP_d^{(l)}[\text{criterion}]$ for each criteria success is defined upon;
6. repeat steps 3-5 for $l=1,...,L$, in the end

$$PP_d = \text{avg}\left[\left(\mathbb{1}\Big\{\bigcap_{\{\text{criteria}\}} \{PP_d^{(l)}[\text{criterion}] > c\}\Big\}\right)_l\right]$$

In the end, we choose the lowest dose d satisfying the criteria.

A.2 Full Data from Sormani's meta-analysis from Chapter 2

Trial	S.Size	$\log(RR_{ARR})(s.e.)$	$\log(RR_{EDSS})(s.e.)$	$\log(RR_{MRI})(s.e.)$
2,38	248	-0.083 (0.15)	0 (0.24)	-0.994 (0.276)
2,38	247	-0.416 (0.147)	-0.342 (0.263)	-0.892 (0.274)
21	26	-0.211 (0.687)	0.122 (0.811)	- (-)
22	251	-0.342 (0.213)	-0.128 (0.261)	- (-)
23,39	172	-0.386 (0.253)	-0.446 (0.301)	-0.4 (0.208)
24	150	-0.896 (0.269)	-0.357 (0.376)	- (-)
4	51	-1.08 (0.484)	-1.66 (0.816)	-0.734 (0.502)
25	40	-0.994 (0.479)	-0.223 (0.807)	- (-)
5,40	376	-0.342 (0.369)	-0.386 (0.189)	-0.545 (0.227)
5,40	371	-0.386 (0.178)	-0.315 (0.186)	-1.05 (0.222)
26	802	0.049 (0.112)	0 (0.116)	-0.117 (0.125)
27	188	-0.342 (0.27)	-0.844 (0.342)	- (-)
28	306	-0.163 (0.196)	0.077 (0.22)	- (-)
29	942	-1.14 (0.156)	-0.528 (0.141)	-1.77 (0.124)
30	1171	-0.799 (0.121)	-0.236 (0.115)	-1.77 (0.112)
31	223	-1.17 (0.304)	-1.14 (0.366)	- (-)
31	221	-1.51 (0.269)	-1.02 (0.354)	- (-)
32	764	0.03 (0.188)	0.285 (0.226)	-0.198 (0.151)
33,34	870	-0.868 (0.164)	-0.363 (0.165)	-1.31 (0.156)
33,34	893	-0.799 (0.161)	-0.308 (0.16)	-1.47 (0.149)
35	118	-0.139 (0.408)	0.207 (0.425)	- (-)
35	123	-0.357 (0.25)	0.039 (0.431)	- (-)
36	130	-0.994 (0.438)	-0.446 (0.397)	-0.261 (0.32)
37	1345	0.058 (0.23)	0.255 (0.112)	-0.328 (0.098)
37	1347	-0.03 (0.181)	0.049 (0.118)	-0.328 (0.082)

Table A.1: Data from Sormani's meta-analysis: $\log(RiskRatio)$ for the MS outcomes with the extracted standard errors

The calculation of approximate standard errors proceeded as follows for each study: Firstly, in the case of the EDSS, we assume a binomial model with treatment effects on the log relative risk scale. As such, $\sigma_\theta = \sqrt{\frac{1}{n_C} + \frac{1}{n_T}}$, where n_C equals the number of patients in the control group and n_T equals the number of patients randomized to the treatment group. Secondly, when calculating the standard error associated with the relapse treatment effects, a Poisson sampling model with multiplicative over-dispersion (Y.C. et al. (2009))

parameter was assumed. Therefore, $\sigma_\gamma = \sqrt{\phi\left(\frac{1}{e_C\lambda_C} + \frac{1}{e_T\lambda_T}\right)}$, where e_C and e_T represent the number of years of patient exposure associated with the control and treatment groups in study i, λ_C and λ_T represent the corresponding estimates for the annualized relapse rates and ϕ is used to control the amount of multiplicative over-dispersion. To calculate the exposure parameters an assumption must be made on how to combine the available drop-out information, which in Sormani et al. (2010) was only provided at the study level rather than the arm level, with the total number of randomized subjects and the planned length of follow-up. Therefore, it was assumed that the subjects with incomplete follow-up where spread across the treatment groups proportionally to the randomization ratio. Further, by assuming that in each study the time of drop-out was uniformly distributed throughout the follow-up period, the expected exposure from the subjects with incomplete follow-up was calculated. This was then added to the exposure from subjects with complete follow-up to calculate e_C and e_T. As the overdispersion could not be estimated directly from the available summary data, we selected a plug-in value of 2. The choice of 2 was based on a recently published investigation and simulation study examining sample size re-estimation Friede and Schmidli (2010b). Finally, the standard errors associated with the MRI treatment effects were based on the assumption of a negative binomial sampling model (Sormani et al. (2001a), van den Elskamp et al. (2009b)), using the parametrization described in Keene et al. (2007b). As such, $\sigma_{\psi i} = \sqrt{\frac{1}{n_C}\left(\frac{1}{\mu_C} + k\right) + \frac{1}{n_{T,i}}\left(\frac{1}{\mu_T} + k\right)}$, where μ_C and μ_T represent the average number of MRI lesions/patient observed, which could be taken from the summary data, and k represents the over-dispersion parameter associated with the negative binomial. In this case, the negative binomial over-dispersion parameter could not be derived from the summary data and therefore a value of $k = 3$ was selected (again based on Friede and Schmidli (2010b) and Friede and Schmidli (2010d)).

A.3 Variable Selection Algorithm

Input: Features $\mathbf{X}$, Outcome $\mathbf{Y}$, distance d, the maximum number of variables to use $maxVar$

Output: Partition that minimizes the empirical risk and predicted value of the outcome for each element of the partition

for $v = 1 \rightarrow V$ **do**
  $vars \leftarrow \{\}$;
  for $k = 1 \rightarrow maxVar$ **do**
    $q \leftarrow |vars| + 1$;
    for $j \notin vars$ **do**
      $tree \leftarrow \text{HOPACH}(\mathbf{X}_v^0[, vars \cup \{j\}])$;
      $\hat{\mathbf{Y}}_j(\alpha) \leftarrow \text{fit}(\text{prune}(tree, \alpha), \mathbf{Y}_v^0)$;
      $Err[j] \leftarrow \int \mathbb{E}_n \mathcal{L}(\mathbf{Y}_v^0, \hat{\mathbf{Y}}_j(\alpha)) \text{d}\alpha$;
    end
    $j^* \leftarrow \underset{j}{\text{argmin}} Err[j]$;
    $loss[v, q](\alpha) \leftarrow \mathbb{E}_n \mathcal{L}(\mathbf{Y}_v^1, \hat{\mathbf{Y}}_{j^*}(\alpha))$;
    $vars \leftarrow vars \cup \{j^*\}$;
  end
  $Risk[q](\alpha) = \frac{1}{V} \sum_{v=1}^{V} loss[v, q](\alpha)$;
end
$(q^*, \alpha^*) = \underset{\alpha, q}{\text{argmin}} Risk[q](\alpha)$;
$vars \leftarrow \{\}$;
for $k = 1 \rightarrow q^*$ **do**
  for $j \notin vars$ **do**
    $Tree \leftarrow \text{prune}(\text{HOPACH}(\mathbf{X}[, vars \cup \{j\}]), \alpha^*)$;
    $\hat{\mathbf{Y}}_j \leftarrow \text{fit}(Tree, \mathbf{Y})$;
    $Err[j] \leftarrow \int \mathbb{E}_n \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}_j(\alpha)) \text{d}\alpha$;
  end
  $j^* \leftarrow \underset{j}{\text{argmin}} Err[j]$;
  $vars \leftarrow vars \cup \{j^*\}$;
end
$Tree \leftarrow \text{prune}(\text{HOPACH}(\mathbf{X}[, vars]), \alpha^*)$;
$\hat{\mathbf{Y}} \leftarrow \text{fit}(Tree, \mathbf{Y})$;

Algorithm 2: HOPSLAM$(d, \mathbf{X}, \mathbf{Y})$

Appendix B

Code

This appendix is similar to Appendix A and contains some of the code that implements the methodology discussed in the Chapters to help replicating and reusing the results and to explain in a clearer way the details of the algorithms.

B.1 R code from Chapter 1

```
##############################################################################
#######################          ledSIR          #############################
##############################################################################
###                                                                        ###
###  This function performs the Sampling-Importance Resampling calculations to  ###
###  allow predictive decision.                                            ###
###  Input: alpha = a sample from interim posterior with alpha for Placebo, dose to  ###
###  be explored and d5;                                                   ###
###         beta = a sample from interim posterior;                        ###
###         N = the size of the posterior sample to be considered;         ###
###         s.size = the total post-interim sample size;                   ###
###         n.sim = number of simulated studies to perform prediction over.  ###
###  Output: PPw = the predictive power given the allocation.              ###
###                                                                        ###
##############################################################################
##############################################################################

ledSIR <- function(alpha,beta,N=1e4,s.size=150,n.sim=500){

    ## indicator function of the event success
    ## notice that: P(criterion) = sum P(criterion |a,b)*P(a,b) = sum IS(a,b) * w(a,b)

    IS1 <- ((log(alpha[,1])-log(alpha[,2]))>0)
    IS2 <- ((log(alpha[,1])-log(alpha[,2]))>log(2))
    IS3 <- ((log(alpha[,3])-log(alpha[,2])+log(1.2))>0)
    #IS <- ((log(alpha[,1])-log(alpha[,2]))>log(2))|((log(alpha[,3])-log(alpha[,2])+log↩
        (1.2))>0))
```

```
## negative binomial likelihood
nb.likelihood <- function(A,Y,E){
          pY <- prod(dnbinom(Y,size=A[1],prob=A[2]/(1+A[2]))^E)
          return(pY)
          }
cat("\n")

## loop over the simulated studies
PS <-  lapply(1:n.sim,function(l){

    cat(paste(100*l/n.sim,"% ",sep="")) # progress

    ## chose one (alpha,beta) couple randomly
    s <- as.integer(runif(1,1,length(beta)))

    ## generate a simulated data-set
    Y <- matrix(rpois(3*s.size,rgamma(3*s.size,shape=alpha[s,],rate=rep(beta[s],3))),
        ncol=3,byrow=TRUE)

    ## r counts the number of time t assumes each value
    r <- list("Pl" = table(Y[,1]),"d" = table(Y[,2]),"dmax" = table(Y[,3]))
    w <- apply(cbind(alpha[,1],beta),1,nb.likelihood,Y=as.numeric(names(r$Pl)),E=r$Pl)
        *
         apply(cbind(alpha[,2],beta),1,nb.likelihood,Y=as.numeric(names(r$d)),E=r$d)*
         apply(cbind(alpha[,3],beta),1,nb.likelihood,Y=as.numeric(names(r$dmax)),E=r$
             dmax)

    (sum(w*IS1)/sum(w)>0.95)&(max(sum(w*IS2)/sum(w), sum(w*IS3)/sum(w))>0.5)

    })

PPw <- mean(unlist(PS))

return(PPw)

}
```

B.2 WinBUGS code from Chapter 1

```
model
{

  ## set priors for the maximum effect

  for(i in 1:10){
    deltasum[i] ~ dnorm(meansum,tausum) }
  tausum <- 1/sigmasum


  ## define delta matrix and stick-breaking priors
  ## for each column (model) the maximum effect is randomly broke along different rows

  stick1[1] ~ dbeta(1,1) # stick variables identify the partition of the maximum effect
  stick1[2] ~ dbeta(1,1)
  stick1[3] ~ dbeta(1,1)
```

```
delta[1,1] <- deltasum[1] * ranked(stick1[],1)
delta[2,1] <- deltasum[1] * (ranked(stick1[],2) - ranked(stick1[],1) )
delta[3,1] <- deltasum[1] * (ranked(stick1[],3) - ranked(stick1[],2) )
delta[4,1] <- deltasum[1] * (1 - ranked(stick1[],3) )

stick2[1] ~ dbeta(1,1)
stick2[2] ~ dbeta(1,1)
delta[1,2] <- 0
delta[2,2] <- deltasum[2] * ranked(stick2[],1)
delta[3,2] <- deltasum[2] * (ranked(stick2[],2) - ranked(stick2[],1) )
delta[4,2] <- deltasum[2] * (1 - ranked(stick2[],2) )

delta[1,3] <- deltasum[3]
delta[2,3] <- 0
delta[3,3] <- 0
delta[4,3] <- 0

stick4 ~ dbeta(1,1)
delta[1,4] <- deltasum[4] * stick4
delta[2,4] <- deltasum[4] * (1 - stick4)
delta[3,4] <- 0
delta[4,4] <- 0

stick5 ~ dbeta(1,1)
delta[1,5] <- 0
delta[2,5] <- 0
delta[3,5] <- deltasum[5] * stick5
delta[4,5] <- deltasum[5] * (1 - stick5)

delta[1,6] <- 0
delta[2,6] <- deltasum[6]
delta[3,6] <- 0
delta[4,6] <- 0

delta[1,7] <- 0
delta[2,7] <- 0
delta[3,7] <- 0
delta[4,7] <- deltasum[7]

stick8 ~ dbeta(1,1)
delta[1,8] <- 0
delta[2,8] <- deltasum[8] * stick8
delta[3,8] <- deltasum[8] * (1 - stick8)
delta[4,8] <- 0

delta[1,9] <- 0
delta[2,9] <- 0
delta[3,9] <- deltasum[9]
delta[4,9] <- 0

stick10[1] ~ dbeta(1,1)
stick10[2] ~ dbeta(1,1)
delta[1,10] <- deltasum[10] * ranked(stick10[],1)
delta[2,10] <- deltasum[10] * (ranked(stick10[],2) - ranked(stick10[],1) )
delta[3,10] <- deltasum[10] * (1 - ranked(stick10[],2) )
delta[4,10] <- 0


## Y[i,j] is the number brain lesions
## for the i-th patient belonging to the j-th dose group
## lambda truncated to avoid underflow

for(j in 1:ndose){
```

```
    for(i in 1:ssize[j]){
      Y[i,j] ~ dpois(lambda[i,j])
      lambda[i,j] ~ dgamma(alpha[j],beta)I(0.001,) }
    csi[j] <- log(beta/alpha[j]) }

  beta <- exp(LB)
  LB ~ dnorm(0,tauB) ## we want beta to be around one
  tauB <- 1/sigmaB


  ## averaging over the models

  mod ~ dcat(prob[1:10])


  ## monotonicity constraints
  for(j in 2:ndose){
    alpha[j] <- alpha[j-1] * exp(-delta[j-1,mod]) }
  alpha[1] <- exp(LA)
  LA ~ dnorm(alpha0,tauA) # we put a prior on the Placebo effect
  tauA <- 1/sigmaA


  ## calculate the posterior probability of Futility & Success

  for(j in 1:ndose){
    futility[j] <- step(csi[1] - csi[j] - log(0.7) )
    success_i[j] <- 1 - step(csi[1] - csi[j])
    success_ii[j] <- max(step(csi[j] - csi[ndose] + log(1.2) ), step(csi[j] - csi[1] + log(0.5) ) )
    ##success_ii[j] <- min(step(csi[j] - csi[ndose] + log(1.2) ), step(csi[j] - csi[1] + log(0.5) ) ) }

}
```

This model can be called directly from R using a syntax like

```
library("R2WinBUGS")

## data for the WinBUGS model
dat <- list(ndose = 5, ssize = rep(50,5),
            meansum = 1.2, sigmasum = 0.81,
            sigmaB = 0.64, alpha0 = 0.1,
            sigmaA = 0.49,prob = rep(1/10,10),Y = Y)

model.name <- "LED.txt" # change to the appropriate path

## parameters to be monitored
para <- c("futility","success_i","success_ii")

## initial values
iniz <- function(){list(deltasum = rep(0,10),
                        stick1 = rbeta(3,1,1),stick2 = rbeta(2,1,1),
                        stick4 = rbeta(1,1,1),stick5 = rbeta(1,1,1),
                        stick8 = rbeta(1,1,1),stick10 = rbeta(2,1,1),
                        LB = 0,LA = 0.1, mod = as.integer(1))}

res  <- bugs(data = dat,
                  inits = iniz,
                  bugs.seed = runif(1) ,
                  parameters.to.save = para,
```

```
                model.file = model.name,
                n.chains = 3,
                n.burnin = 500,
                n.thin = 2,
                n.sims = 1000,
                debug = FALSE)
```

B.3 WinBUGS code from Chapter 2

B.3.1 3 Level Model

```
model{



        y1[1:4] ~ dmnorm(theta1[1:4],T1[1:4,1:4])

        theta1[1] <- alpha1 + beta1*theta1[2] + epsilon[1]
                     theta1[2] ~ dnorm(0,prec.gamma)
                     theta1[3] <- alpha1 + beta1*theta1[4] + epsilon[2]
                     theta1[4] ~ dnorm(0,prec.gamma)


        y2[1:4] ~ dmnorm(theta2[1:4],T2[1:4,1:4])

        theta2[1] <- alpha1 + beta1*theta2[2] + epsilon[3]
                     theta2[2] ~ dnorm(0,prec.gamma)
                     theta2[3] <- alpha1 + beta1*theta2[4] + epsilon[4]
                     theta2[4] ~ dnorm(0,prec.gamma)


        y3[1:2] ~ dmnorm(theta3[1:2],T3[1:2,1:2])

        theta3[1] <- alpha1 + beta1*theta3[2] + epsilon[5]
                     theta3[2] ~ dnorm(0,prec.gamma)


        y4[1:2] ~ dmnorm(theta4[1:2],T4[1:2,1:2])

        theta4[1] <- alpha1 + beta1*theta4[2] + epsilon[6]
                     theta4[2] ~ dnorm(0,prec.gamma)


        y5[1:2] ~ dmnorm(theta5[1:2],T5[1:2,1:2])

        theta5[1] <- alpha1 + beta1*theta5[2] + epsilon[7]
                     theta5[2] ~ dnorm(0,prec.gamma)


        y6[1:2] ~ dmnorm(theta6[1:2],T6[1:2,1:2])

        theta6[1] <- alpha1 + beta1*theta6[2] + epsilon[8]
                     theta6[2] ~ dnorm(0,prec.gamma)
```

```
y7[1:2] ~ dmnorm(theta7[1:2],T7[1:2,1:2])

theta7[1] <- alpha1 + beta1*theta7[2] + epsilon[9]
             theta7[2] ~ dnorm(0,prec.gamma)


y8[1:2] ~ dmnorm(theta8[1:2],T8[1:2,1:2])

theta8[1] <- alpha1 + beta1*theta8[2] + epsilon[10]
             theta8[2] ~ dnorm(0,prec.gamma)


y9[1:6] ~ dmnorm(theta9[1:6],T9[1:6,1:6])
theta9[1] <- alpha1 + beta1*theta9[2] + epsilon[11]
          theta9[2] <- alpha2 + beta2*theta9[3] + delta[1]
          theta9[3] ~ dnorm(0,prec.gamma)
          theta9[4] <- alpha1 + beta1*theta9[5] + epsilon[12]
          theta9[5] <- alpha2 + beta2*theta9[6] + delta[2]
          theta9[6] ~ dnorm(0,prec.gamma)


y10[1:6] ~ dmnorm(theta10[1:6],T10[1:6,1:6])
theta10[1] <- alpha1 + beta1*theta10[2] + epsilon[13]
          theta10[2] <- alpha2 + beta2*theta10[3] + delta[3]
          theta10[3] ~ dnorm(0,prec.gamma)
          theta10[4] <- alpha1 + beta1*theta10[5] + epsilon[14]
          theta10[5] <- alpha2 + beta2*theta10[6] + delta[4]
          theta10[6] ~ dnorm(0,prec.gamma)


y11[1:6] ~ dmnorm(theta11[1:6],T11[1:6,1:6])
theta11[1] <- alpha1 + beta1*theta11[2] + epsilon[15]
          theta11[2] <- alpha2 + beta2*theta11[3] + delta[5]
          theta11[3] ~ dnorm(0,prec.gamma)
          theta11[4] <- alpha1 + beta1*theta11[5] + epsilon[16]
          theta11[5] <- alpha2 + beta2*theta11[6] + delta[6]
          theta11[6] ~ dnorm(0,prec.gamma)


y12[1:6] ~ dmnorm(theta12[1:6],T12[1:6,1:6])
theta12[1] <- alpha1 + beta1*theta12[2] + epsilon[17]
          theta12[2] <- alpha2 + beta2*theta12[3] + delta[7]
          theta12[3] ~ dnorm(0,prec.gamma)
          theta12[4] <- alpha1 + beta1*theta12[5] + epsilon[18]
          theta12[5] <- alpha2 + beta2*theta12[6] + delta[8]
          theta12[6] ~ dnorm(0,prec.gamma)


y13[1:3] ~ dmnorm(theta13[1:3],T13[1:3,1:3])
theta13[1] <- alpha1 + beta1*theta13[2] + epsilon[19]
          theta13[2] <- alpha2 + beta2*theta13[3] + delta[9]
          theta13[3] ~ dnorm(0,prec.gamma)


y14[1:3] ~ dmnorm(theta14[1:3],T14[1:3,1:3])
theta14[1] <- alpha1 + beta1*theta14[2] + epsilon[20]
          theta14[2] <- alpha2 + beta2*theta14[3] + delta[10]
          theta14[3] ~ dnorm(0,prec.gamma)


y15[1:3] ~ dmnorm(theta15[1:3],T15[1:3,1:3])
theta15[1] <- alpha1 + beta1*theta15[2] + epsilon[21]
          theta15[2] <- alpha2 + beta2*theta15[3] + delta[11]
```

```
            theta15[3] ~ dnorm(0,prec.gamma)


    y16[1:3] ~ dmnorm(theta16[1:3],T16[1:3,1:3])
    theta16[1] <- alpha1 + beta1*theta16[2] + epsilon[22]
            theta16[2] <- alpha2 + beta2*theta16[3] + delta[12]
            theta16[3] ~ dnorm(0,prec.gamma)


    y17[1:3] ~ dmnorm(theta17[1:3],T17[1:3,1:3])
    theta17[1] <- alpha1 + beta1*theta17[2] + epsilon[23]
            theta17[2] <- alpha2 + beta2*theta17[3] + delta[13]
            theta17[3] ~ dnorm(0,prec.gamma)


    y18[1:3] ~ dmnorm(theta18[1:3],T18[1:3,1:3])
    theta18[1] <- alpha1 + beta1*theta18[2] + epsilon[24]
            theta18[2] <- alpha2 + beta2*theta18[3] + delta[14]
            theta18[3] ~ dnorm(0,prec.gamma)


    y19[1:3] ~ dmnorm(theta19[1:3],T19[1:3,1:3])
    theta19[1] <- alpha1 + beta1*theta19[2] + epsilon[25]
            theta19[2] <- alpha2 + beta2*theta19[3] + delta[15]
            theta19[3] ~ dnorm(0,prec.gamma)




    for(j in 1:25){
        epsilon[j]~dnorm(0,prec.tau)
        }

    for(j in 1:15){
        delta[j]~dnorm(0,prec.omega)
        }

    alpha1~dnorm(0,A1)
    beta1~dnorm(0,B1)
    alpha2~dnorm(0,A2)
    beta2~dnorm(0,B2)
    prec.gamma<-pow(1000,-2)
    prec.tau<-pow(tau,-2)
    tau~dnorm(0,v)I(0,)
    prec.omega<-pow(omega,-2)
    omega~dnorm(0,v)I(0,)

    A1 <- pow(a1,-2)
    B1 <- pow(b1,-2)
    A2 <- pow(a2,-2)
    B2 <- pow(b2,-2)
    V <- pow(v,-2)

}
```

B.3.2 Extended 3 Level Model (Equation 2.11)

```
model{

    y1[1:4] ~ dmnorm(theta1[1:4],T1[1:4,1:4])
    theta1[1] <- alpha1 + beta1*theta1[2] + epsilon[1]
    theta1[2] ~ dnorm(0,prec.gamma)
    theta1[3] <- alpha1 + beta1*theta1[4] + epsilon[2]
    theta1[4] ~ dnorm(0,prec.gamma)


    y2[1:4] ~ dmnorm(theta2[1:4],T2[1:4,1:4])
    theta2[1] <- alpha1 + beta1*theta2[2] + epsilon[3]
    theta2[2] ~ dnorm(0,prec.gamma)
    theta2[3] <- alpha1 + beta1*theta2[4] + epsilon[4]
    theta2[4] ~ dnorm(0,prec.gamma)


    y3[1:2] ~ dmnorm(theta3[1:2],T3[1:2,1:2])
    theta3[1] <- alpha1 + beta1*theta3[2] + epsilon[5]
    theta3[2] ~ dnorm(0,prec.gamma)


    y4[1:2] ~ dmnorm(theta4[1:2],T4[1:2,1:2])
    theta4[1] <- alpha1 + beta1*theta4[2] + epsilon[6]
    theta4[2] ~ dnorm(0,prec.gamma)


    y5[1:2] ~ dmnorm(theta5[1:2],T5[1:2,1:2])
    theta5[1] <- alpha1 + beta1*theta5[2] + epsilon[7]
    theta5[2] ~ dnorm(0,prec.gamma)


    y6[1:2] ~ dmnorm(theta6[1:2],T6[1:2,1:2])
    theta6[1] <- alpha1 + beta1*theta6[2] + epsilon[8]
    theta6[2] ~ dnorm(0,prec.gamma)


    y7[1:2] ~ dmnorm(theta7[1:2],T7[1:2,1:2])
    theta7[1] <- alpha1 + beta1*theta7[2] + epsilon[9]
    theta7[2] ~ dnorm(0,prec.gamma)


    y8[1:2] ~ dmnorm(theta8[1:2],T8[1:2,1:2])
    theta8[1] <- alpha1 + beta1*theta8[2] + epsilon[10]
    theta8[2] ~ dnorm(0,prec.gamma)


    y9[1:6] ~ dmnorm(theta9[1:6],T9[1:6,1:6])
    theta9[1] <- alpha1 + beta1*theta9[2] +beta3*theta9[3] +epsilon[11]
    theta9[2] <- alpha2 + beta2*theta9[3] + delta[1]
    theta9[3] ~ dnorm(0,prec.gamma)
    theta9[4] <- alpha1 + beta1*theta9[5] +beta3*theta9[6] + epsilon[12]
    theta9[5] <- alpha2 + beta2*theta9[6] + delta[2]
    theta9[6] ~ dnorm(0,prec.gamma)


    y10[1:6] ~ dmnorm(theta10[1:6],T10[1:6,1:6])
    theta10[1] <- alpha1 + beta1*theta10[2] +beta3*theta10[3]+ epsilon[13]
    theta10[2] <- alpha2 + beta2*theta10[3] + delta[3]
    theta10[3] ~ dnorm(0,prec.gamma)
    theta10[4] <- alpha1 + beta1*theta10[5] +beta3*theta10[6]+ epsilon[14]
    theta10[5] <- alpha2 + beta2*theta10[6] + delta[4]
    theta10[6] ~ dnorm(0,prec.gamma)
```

```
y11[1:6] ~ dmnorm(theta11[1:6],T11[1:6,1:6])
theta11[1] <- alpha1 + beta1*theta11[2] +beta3*theta11[3]+ epsilon[15]
theta11[2] <- alpha2 + beta2*theta11[3] + delta[5]
theta11[3] ~ dnorm(0,prec.gamma)
theta11[4] <- alpha1 + beta1*theta11[5] +beta3*theta11[6]+ epsilon[16]
theta11[5] <- alpha2 + beta2*theta11[6] + delta[6]
theta11[6] ~ dnorm(0,prec.gamma)


y12[1:6] ~ dmnorm(theta12[1:6],T12[1:6,1:6])
theta12[1] <- alpha1 + beta1*theta12[2] +beta3*theta12[3]+ epsilon[17]
theta12[2] <- alpha2 + beta2*theta12[3] + delta[7]
theta12[3] ~ dnorm(0,prec.gamma)
theta12[4] <- alpha1 + beta1*theta12[5] +beta3*theta12[6]+ epsilon[18]
theta12[5] <- alpha2 + beta2*theta12[6] + delta[8]
theta12[6] ~ dnorm(0,prec.gamma)


y13[1:3] ~ dmnorm(theta13[1:3],T13[1:3,1:3])
theta13[1] <- alpha1 + beta1*theta13[2] +beta3*theta13[3]+ epsilon[19]
theta13[2] <- alpha2 + beta2*theta13[3] + delta[9]
theta13[3] ~ dnorm(0,prec.gamma)


y14[1:3] ~ dmnorm(theta14[1:3],T14[1:3,1:3])
theta14[1] <- alpha1 + beta1*theta14[2] +beta3*theta14[3]+ epsilon[20]
theta14[2] <- alpha2 + beta2*theta14[3] + delta[10]
theta14[3] ~ dnorm(0,prec.gamma)


y15[1:3] ~ dmnorm(theta15[1:3],T15[1:3,1:3])
theta15[1] <- alpha1 + beta1*theta15[2] +beta3*theta15[3]+ epsilon[21]
theta15[2] <- alpha2 + beta2*theta15[3] + delta[11]
theta15[3] ~ dnorm(0,prec.gamma)


y16[1:3] ~ dmnorm(theta16[1:3],T16[1:3,1:3])
theta16[1] <- alpha1 + beta1*theta16[2] +beta3*theta16[3]+ epsilon[22]
theta16[2] <- alpha2 + beta2*theta16[3] + delta[12]
theta16[3] ~ dnorm(0,prec.gamma)


y17[1:3] ~ dmnorm(theta17[1:3],T17[1:3,1:3])
theta17[1] <- alpha1 + beta1*theta17[2] +beta3*theta17[3]+ epsilon[23]
theta17[2] <- alpha2 + beta2*theta17[3] + delta[13]
theta17[3] ~ dnorm(0,prec.gamma)


y18[1:3] ~ dmnorm(theta18[1:3],T18[1:3,1:3])
theta18[1] <- alpha1 + beta1*theta18[2] + beta3*theta18[3]+ epsilon[24]
theta18[2] <- alpha2 + beta2*theta18[3] + delta[14]
theta18[3] ~ dnorm(0,prec.gamma)


y19[1:3] ~ dmnorm(theta19[1:3],T19[1:3,1:3])
theta19[1] <- alpha1 + beta1*theta19[2] + beta3*theta19[3] + epsilon[25]
theta19[2] <- alpha2 + beta2*theta19[3] + delta[15]
theta19[3] ~ dnorm(0,prec.gamma)


for(j in 1:25){
      epsilon[j]~dnorm(0,prec.tau)
```

```
        }

    for(j in 1:15){
        delta[j]~dnorm(0,prec.omega)
        }

    alpha1~dnorm(0,A)
    beta1~dnorm(0,B)
    beta3~dnorm(0,B)
    alpha2~dnorm(0,A2)
    beta2~dnorm(0,B2)
    prec.gamma<-pow(1000,-2)
    prec.tau<-pow(tau,-2)
    tau~dnorm(0,v)I(0,)
    prec.omega<-pow(omega,-2)
    omega~dnorm(0,v)I(0,)

    A <- pow(a1,-2)
    B <- pow(b1,-2)
    A2 <- pow(a2,-2)
    B2 <- pow(b2,-2)
    V <- pow(v,-2)

    }
```

B.3.3 Multivariate meta-analysis (Equation 2.12)

```
model{

    y1[1:4] ~ dmnorm(theta1b[1:4],T1[1:4,1:4])
    theta1b[1]<-theta1[1]
    theta1b[2]<-theta1[2]
    theta1b[3]<-theta1[4]
    theta1b[4]<-theta1[5]

    theta1[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])
    theta1[4:6]~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


    y2[1:4] ~ dmnorm(theta2b[1:4],T2[1:4,1:4])
    theta2b[1]<-theta2[1]
    theta2b[2]<-theta2[2]
    theta2b[3]<-theta2[4]
    theta2b[4]<-theta2[5]

    theta2[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])
    theta2[4:6]~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


    y3[1:2] ~ dmnorm(theta3[1:2],T3[1:2,1:2])
    theta3[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


    y4[1:2] ~ dmnorm(theta4[1:2],T4[1:2,1:2])
        theta4[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


    y5[1:2] ~ dmnorm(theta5[1:2],T5[1:2,1:2])
```

```
theta5[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y6[1:2] ~ dmnorm(theta6[1:2],T6[1:2,1:2])
theta6[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y7[1:2] ~ dmnorm(theta7[1:2],T7[1:2,1:2])
theta7[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y8[1:2] ~ dmnorm(theta8[1:2],T8[1:2,1:2])
theta8[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


# create a blocked diagonal  covariance matrix for trials 9-12
for(i in 1:3){
      for(j in 1:3){
            omega.inv2[i,j]<-omega.inv[i,j]
        omega.inv2[i+3,j+3]<-omega.inv[i,j]
        omega.inv2[i,j+3]<-0
        omega.inv2[i+3,j]<-0
        }
    mu2[i]<-mu[i]
    mu2[i+3]<-mu[i]
    }

y9[1:6] ~ dmnorm(theta9[1:6],T9[1:6,1:6])
theta9[1:6] ~ dmnorm(mu2[1:6],omega.inv2[1:6,1:6])


y10[1:6] ~ dmnorm(theta10[1:6],T10[1:6,1:6])
theta10[1:6] ~ dmnorm(mu2[1:6],omega.inv2[1:6,1:6])


y11[1:6] ~ dmnorm(theta11[1:6],T11[1:6,1:6])
theta11[1:6] ~ dmnorm(mu2[1:6],omega.inv2[1:6,1:6])


y12[1:6] ~ dmnorm(theta12[1:6],T12[1:6,1:6])
theta12[1:6] ~ dmnorm(mu2[1:6],omega.inv2[1:6,1:6])


y13[1:3] ~ dmnorm(theta13[1:3],T13[1:3,1:3])
theta13[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y14[1:3] ~ dmnorm(theta14[1:3],T14[1:3,1:3])
theta14[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y15[1:3] ~ dmnorm(theta15[1:3],T15[1:3,1:3])
theta15[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y16[1:3] ~ dmnorm(theta16[1:3],T16[1:3,1:3])
theta16[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


y17[1:3] ~ dmnorm(theta17[1:3],T17[1:3,1:3])
theta17[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])

y18[1:3] ~ dmnorm(theta18[1:3],T18[1:3,1:3])
```

```
    theta18[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


    y19[1:3] ~ dmnorm(theta19[1:3],T19[1:3,1:3])
    theta19[1:3] ~ dmnorm(mu[1:3],omega.inv[1:3,1:3])


    A <- pow(a,-2)
    for(i in 1:3){
          mu[i]~dnorm(0,A)
          }

    omega.inv[1:3,1:3] <- inverse(rev[1:3,1:3])

    rev[1,1] <- omeg1*omeg1
    rev[2,2] <- omeg2*omeg2
    rev[3,3] <- omeg3*omeg3
    rev[1,2] <- rho12*omeg1*omeg2
    rev[2,1] <- rho12*omeg1*omeg2
    rev[1,3] <- rho13*omeg1*omeg3
    rev[3,1] <- rho13*omeg1*omeg3
    rev[2,3] <- rho23*omeg2*omeg3
    rev[3,2] <- rho23*omeg2*omeg3

    low12 <- rho13*rho23 - sq12
    hig12 <- rho13*rho23 + sq12
    low13 <- rho12*rho23 - sq13
    hig13 <- rho12*rho23 + sq13
    low23 <- rho12*rho13 - sq23
    hig23 <- rho12*rho13 + sq23

    sq12 <- sqrt(c23*c13)
    sq13 <- sqrt(c12*c23)
    sq23 <- sqrt(c12*c13)

    c12 <- 1-rho12*rho12
    c13 <- 1-rho13*rho13
    c23 <- 1-rho23*rho23

    omeg1 ~ dunif(0.01,omega.diag[1])
    omeg2 ~ dunif(0.01,omega.diag[2])
    omeg3 ~ dunif(0.01,omega.diag[3])

    rho12 ~ dunif(low12,hig12)
    rho13 ~ dunif(low13,hig13)
    rho23 ~ dunif(low23,hig23)

    }
```

B.4 HOPSLAM Demo Chapter 3

```
rm(list=ls())
gc()

library("HOPSLAM")
```

```
A <- simHop(n=100,p=1000,mu=0, muk=c(-10,0,10),sX=1,sY=1)

regression <- with(A,HOPSLAM(X,Y,fold=10L,verbose=TRUE,d="cosangle"))
regression

Z <- cut(A$Y,c(-Inf,-3,3,Inf))
classification <- with(A,HOPSLAM(X,Z,fold=10L,balanced=TRUE,verbose=TRUE,method="↩
    classification",d="corr"))
classification

selection <- with(A,HOPSLAM(X,Y,fold=5L,verbose=TRUE,d="cosangle",maxVars=2,bNb=TRUE,↩
    forward=TRUE))
selection
```

Appendix C

Additional Results

In this Appendix we include some additional results from the other Chapters.

C.1 Additional Scenarios from Chapter 1

C.1.1 Dose-Response Scenarios

In addition to the Moderate scenario discussed in Section 1.3.2, four additional dose-response scenarios are considered in the following (see also Figure C.1).

Optimistic is given by $\boldsymbol{\alpha}^{(0)}_{\text{Opt}} = [1.1, 0.06, 0.055, 0.05, 0.05]$:

1. $\mu_2^{(0)}/\mu_1^{(0)} = 0.05$ and $\mu_2^{(0)}/\mu_5^{(0)} = 1.2$, so this dose meets only the first of the **(ii)** conditions;
2. $\mu_3^{(0)}/\mu_1^{(0)} = 0.05$ and $\mu_3^{(0)}/\mu_5^{(0)} = 1.1$, so this dose meets both of the **(ii)** conditions;
3. $\mu_4^{(0)} = \mu_5^{(0)}$(same as the effect of Dose 5) $\Rightarrow \mu_4/\mu_1 = 0.04$ and $\mu_4^{(0)}/\mu_5^{(0)} = 1$, so this dose meets both **(ii)** condition.

We can see in the picture below a representation of this dose-response scheme, being the light grey bar the effect under the LED and any dose below the light grey line satisfying the first of the (ii) conditions.

Intermediate 1 given by $\boldsymbol{\alpha}^{(0)}_{\text{flat a}} = [1.1, 0.9, 0.85, 0.75, 0.7]$:

1. $\mu_2^{(0)}/\mu_1^{(0)} = 0.8$ and $\mu_2^{(0)}/\mu_5^{(0)} = 1.3$, so this dose is futile and meets none of the **(ii)** conditions;

2. $\mu_3^{(0)}/\mu_1^{(0)} = 0.77$ and $\mu_3^{(0)}/\mu_5^{(0)} = 1.2$, so this dose is also futile and meets none of the **(ii)** conditions;
3. $\mu_4^{(0)}/\mu_1^{(0)} = 0.68$ and $\mu_4^{(0)}/\mu_5^{(0)} = 1.07$, so this dose meets the second of the **(ii)** conditions and is therefore the LED.
4. $\mu_5^{(0)}/\mu_1^{(0)} = 0.6$, so the higher dose does not meet the first of the **(ii)** conditions.

Intermediate 2 $\boldsymbol{\alpha}_{\text{flat b}}^{(0)} = [1.1, 0.8, 0.6, 0.58, 0.55]$

1. $\mu_2^{(0)}/\mu_1^{(0)} = 0.7$ and $\mu_2^{(0)}/\mu_5^{(0)} = 1.4$, so this dose is futile and meets none of the **(ii)** conditions;
2. $\mu_3^{(0)}/\mu_1^{(0)} = 0.54$ and $\mu_3^{(0)}/\mu_5^{(0)} = 1.09$, so this dose meets the second of the **(ii)** conditions and is therefore the LED.
3. $\mu_4^{(0)}/\mu_1^{(0)} = 0.52$ and $\mu_4^{(0)}/\mu_5^{(0)} = 1.05$, so this dose meets the second of the **(ii)**.
4. $\mu_5^{(0)}/\mu_1^{(0)} = 0.6$, so the higher dose does barely meet the first of the **(ii)** conditions.

Pessimistic is given by $\boldsymbol{\alpha}_{\text{Pes}}^{(0)} = [1.1, 1.1, 0.9, 0.8, 0.05]$:

1. $\mu_2^{(0)} = \mu_1^{(0)}$(same as the effect of the placebo) $\Rightarrow \mu_2^{(0)}/\mu_1^{(0)} = 1$ and $\mu_2^{(0)}/\mu_5^{(0)} = 22$, so this dose is futile and meets none of the **(ii)** conditions;
2. $\mu_3^{(0)}/\mu_1^{(0)} = 0.8$ and $\mu_3^{(0)}\mu_5^{(0)} = 18$, so this dose is futile meets none of the **(ii)** conditions;
3. $\mu_4^{(0)}/\mu_1^{(0)} = 0.7$ and $\mu_4^{(0)}\mu_5^{(0)} = 16$, so this dose is futile and meets none of the **(ii)** conditions.

C.1.2 Results

The following Tables C.1 to C.10 present the results under the additional scenarios. A superscript $-$ will imply a rounding to the next greater integer (e.g. $1.00^- = 0.999$).

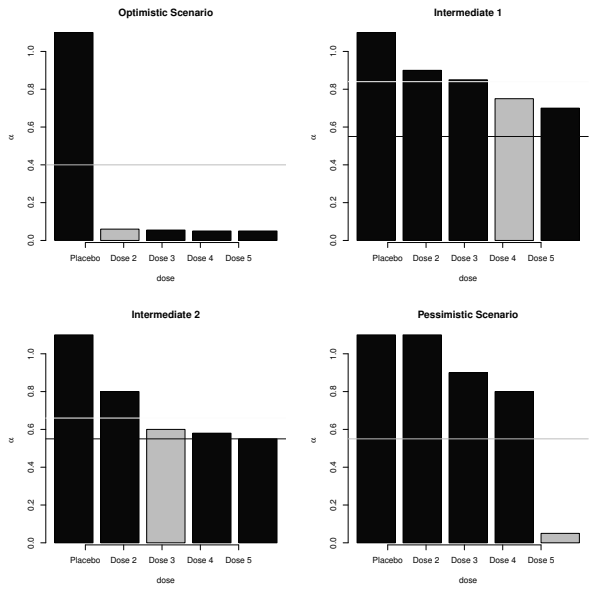

Figure C.1: True dose-response relationship under the Various Scenarios. All doses whose bar is lower that the dark grey line meet the second of the (ii) criteria.

	1:1:1		1:2:1	
	50%-50%	30%-70%	50%-50%	30%-70%
Optimistic	138	130	145	150
Intermediate 1	180	151	168	152
Intermediate 2	216	191	216	195
Pessimistic	183	177	189	188

Table C.1: Expected sample size for the adaptive design under different scenarios

	Optimistic	Intermediate 1	Intermediate 2	Pessimistic
$\mathbb{P}\{$Right dose$\}$	0.98	0.27	0.39	0.94
$\mathbb{P}\{$Selection$\}$	1.00	0.37	0.68	0.06

Table C.2: Non-Adaptive Design

	1:1:1-30%	1:1:1-50%	1:2:1-30%	1:2:1-50%
$\mathbb{P}\{$Selection$\}$	0.72	0.9	0.60	0.84
$\mathbb{P}\{$False Negative$\}$	0.00	0.00	0.00	0.00

Table C.3: Optimistic Scenario: Interim Decision

	1:1:1-0.4	1:1:1-0.5	1:1:1-0.6	1:2:1-0.4	1:2:1-0.5	1:2:1-0.6
$\mathbb{P}$\{Right dose|30%,70%\}	0.99	0.93	0.96	0.98	0.98	0.98
$\mathbb{P}$\{Selection|30%,70%\}	1.00	1.00	1.00	1.00	1.00	1.00
$\mathbb{P}$\{False Negative|30%,70%\}	0.00	0.00	0.00	0.00	0.00	0.00
$\mathbb{P}$\{Right dose|50%,50%\}	0.98	0.97	1.00^{-}	0.98	1.00^{-}	0.98
$\mathbb{P}$\{Selection|50%,50%\}	1.00	1.00	1.00	1.000	1.000	1.000
$\mathbb{P}$\{False Negative|50%,50%\}	0.00	0.00	0.00	0.00	0.00	0.00

Table C.4: Optimistic Scenario: different allocations and different predictive probability thresholds

	1:1:1-30%	1:1:1-50%	1:2:1-30%	1:2:1-50%
$\mathbb{P}$\{False Positive\}	0.00	0.00	0.00	0.00
$\mathbb{P}$\{False Negative\}	0.59	0.56	0.59	0.66

Table C.5: Intermediate 1 Scenario Interim Decision

	1:1:1-0.4	1:1:1-0.5	1:1:1-0.6	1:2:1-0.4	1:2:1-0.5	1:2:1-0.6
$\mathbb{P}$\{Right dose|30%,70%\}	0.24	0.24	0.24	0.21	0.20	0.21
$\mathbb{P}$\{Selection|30%,70%\}	0.32	0.32	0.32	0.29	0.29	0.29
$\mathbb{P}$\{False Negative|30%,70%\}	0.48	0.48	0.48	0.71	0.71	0.71
$\mathbb{P}$\{Right dose|50%,50%\}	0.24	0.24	0.22	0.18	0.18	0.20
$\mathbb{P}$\{Selection|50%,50%\}	0.32	0.32	0.32	0.26	0.26	0.26
$\mathbb{P}$\{False Negative|50%,50%\}	0.68	0.68	0.68	0.74	0.74	0.74

Table C.6: Intermediate 1 Scenario: different allocations and different predictive probability thresholds

	1:1:1-30%	1:1:1-50%	1:2:1-30%	1:2:1-50%
$\mathbb{P}$\{False Positive\}	0.02	0.00^{-}	0.02	0.00
$\mathbb{P}$\{False Negative\}	0.33	0.27	0.31	0.27

Table C.7: Intermediate 2 Scenario: Interim Decision

	1:1:1-0.4	1:1:1-0.5	1:1:1-0.6	1:2:1-0.4	1:2:1-0.5	1:2:1-0.6
$\mathbb{P}$\{Right dose|30%,70%\}	0.33	0.41	0.27	0.33	0.34	0.29
$\mathbb{P}$\{Selection|30%,70%\}	0.65	0.65	0.65	0.65	0.65	0.65
$\mathbb{P}$\{False Negative|30%,70%\}	0.35	0.35	0.35	0.35	0.35	0.35
$\mathbb{P}$\{Right dose|50%,50%\}	0.49	0.41	0.35	0.39	0.39	0.45
$\mathbb{P}$\{Selection|50%,50%\}	0.70	0.70	0.70	0.69	0.69	0.69
$\mathbb{P}$\{False Negative|50%,50%\}	0.30	0.30	0.30	0.31	0.31	0.31

Table C.8: Intermediate 2 Scenario: different allocations and different predictive probability thresholds

	1:1:1-30%	1:1:1-50%	1:2:1-30%	1:2:1-50%
$\mathbb{P}$\{False Positive\}	0.00	0.00	0.00	0.0
$\mathbb{P}$\{True Negative\}	0.44	0.54	0.37	0.5

Table C.9: Pessimistic Scenario: Interim Decision

	1:1:1-0.4	1:1:1-0.5	1:1:1-0.6	1:2:1-0.4	1:2:1-0.5	1:2:1-0.6
$\mathbb{P}$\{Right dose|30%,70%\}	0.41	0.41	0.41	0.47	0.47	0.47
$\mathbb{P}$\{False Positive|30%,70%\}	0.15	0.15	0.15	0.16	0.16	0.16
$\mathbb{P}$\{True Negative|30%,70%\}	0.85	0.85	0.85	0.84	0.84	0.84
$\mathbb{P}$\{Right dose|50%,50%\}	0.29	0.29	0.29	0.34	0.34	0.34
$\mathbb{P}$\{False Positive|50%,50%\}	0.17	0.17	0.17	0.16	0.16	0.16
$\mathbb{P}$\{True Negative|50%,50%\}	0.83	0.83	0.83	0.84	0.84	0.84

Table C.10: Pessimistic Scenario Scenario: different allocations and different predictive probability thresholds