RISK ESTIMATION USING RANDOM FORESTS

by

Mary M. Brown

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
March 2017

Table of Contents

List of Tables

List of Figures

Abstract

The random forest probability machine (RFPM) introduced by Dasgupta et al. (2014) is a consistent, non-parametric regression technique that, when applied to binary outcomes, enables calculation of predictor effect size estimates. Using simulation, RFPMs are found to estimate main effects for binary and categorical predictors, and interaction effects for binary predictors with minimal bias. These estimates are almost as efficient as those from a correctly specified logistic regression model when the data-generating model is logistic. The intuitive interaction detection method in Dasgupta et al. (2014) is shown to be a relatively quick screening process to identify any potential interaction effects, but should be used with caution. Using RFPMs to estimate the effect of a continuous predictor produces estimates with minimal bias when the effect size is linear and small. The RFPM methods are applied to a large Nova Scotia dataset to identify and quantify risk factors for fetal growth abnormalities.

List of Abbreviations and Symbols Used

$\boldsymbol{\beta}$ vector of regression coefficients.

$\boldsymbol{x}_i$ p dimensional vector of observed predictors for observation i.

y_i Observed response for observation i.

AR Attributable risk.

ARR Attributable risk reduction.

CART Classification and regression tree.

D Cross-entropy.

G Gini index.

ICE Individual causal effect.

NNT Number needed to treat.

NSAPD Nova Scotia Atlee Perinatal Database.

OOB Out-of-bag.

OR Odds ratio.

PAF Population attributable fraction.

PAR Population attributable risk.

PM Probability machine.

RD Risk difference.

RERI Relative excess risk due to interaction.

RFPM Random forest probability machine.

ROR Ratio of odds ratios.

RR Relative risk.

RSS Residual sum of squares.

SGA Small for gestational age.

Chapter 1

Introduction

Relationships between observed variables and a binary outcome are typically examined using logistic regression. Logistic regression models are fairly straightforward to estimate and interpret, and are easily implemented in all major statistical software packages. However, logistic regression models can be cumbersome in high dimensional problems and have a limited ability to incorporate complex interactions of predictors. In order to obtain unbiased effect estimates and predictions, the logistic regression model must be correctly specified, which is rarely achieved in practice.

Malley et al. (2012) have recently described the concept of a probability machine. A probability machine is a consistent, non-parametric regression technique that, when applied to binary outcomes, generates an estimated probability for each observation. This predicted probability is equivalent to the conditional probability of success for that observation given the set of predictors. A probability machine has many desirable properties including that it does not require any assumptions about the distribution of the data or the shape of the relationship of the predictors with the outcome, nor does it require the explicit structural specification of the presence of interactions, and can be used in high dimensional data sets.

Based on the concept of probability machines, Dasgupta et al. (2014) developed random forest probability machines (RFPMs) using slight alterations of the random forest learning algorithm introduced by Breiman (2001). A RFPM can be applied to large data sets and high dimensional problems, and only requires specification of which predictors are to be included rather than specifying an explicit model. RFPMs can be used to estimate counterfactual probabilities of success at the individual level, which enables for the calculation of various risk effect measures such as odds ratios, risk ratios, and attributable risks. Subgroup-specific or sample estimates of risk can be obtained by averaging over the appropriate individual estimates.

Dasgupta et al. (2014) have demonstrated the use of RFPMs in simulated scenarios and have proposed methods for main effect estimation, interaction estimation, and interaction discovery for binary predictors. The use of RFPMs in simulated scenarios to estimate effects for categorical or continuous predictors has yet to be shown. Also, the calculation of confidence intervals for probability and effect size estimates obtained using RFPMs has not yet been examined. Although the properties of RFPMs have been tested in simulated scenarios, they have not been used with real life data yet.

The objective of this thesis is to further expand on the RFPM methodology proposed by Dasgupta et al. (2014) and apply RFPM methods to a real life data set. Chapter 2 consists of a literature review of risk estimation in epidemiology, and describes both logistic regression and RFPMs in detail. RFPMs are built using various parameters of the random forest algorithm. The effects of these parameters in the RFPM methodology on main effect estimation for binary predictors are examined in Chapter 3. The problem of detecting and estimating interaction effects for binary predictors using the the intuitive interaction detection method and the four-machine RFPM method is addressed in Chapter 4. In Chapter 5, a method for constructing confidence intervals for risk estimates derived from RFPMs using a bootstrap method is outlined. The more complex issue of estimating risk effects for categorical and continuous predictors is considered in Chapter 6. RFPM methods are then used to identify potential risk factors for fetal growth abnormalities using a data set derived from the Nova Scotia Atlee Perinatal Database in Chapter 7.

Chapter 2

Risk Estimation

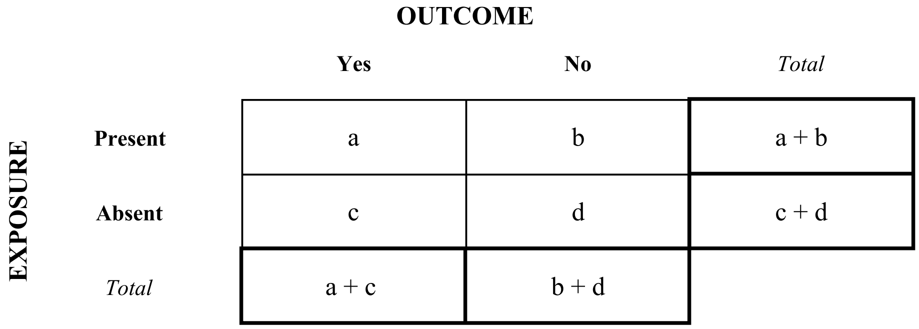
The main goal of epidemiological research is often identifying associations between exposures and outcomes. Traditionally speaking, these outcomes have been defined in terms of disease and can be either expressed as continuous or discrete variables. Linear regression is commonly used to analyze continuous outcomes such as blood pressure or glucose levels, whereas logistic regression is commonly used to analyze dichotomous outcome variables. In order to identify any association, exposures and outcomes need to first be measured quantitatively. These measures are referred to as absolute measures of disease frequency and can be of two different types, incidence or prevalence.

Once the outcomes and exposures have been measured, their association can be evaluated by calculating various measures of association or effect. Measures of association or effect fall into two major categories: absolute difference measures and relative difference or ratio measures (generally called relative risks). Lastly, the impact of the removal of an exposure on the outcome can be evaluated by computing measures of potential impact, such as attributable risk percent and population attributable risk fraction. In this chapter, measures of disease frequency, association or effect, and potential impact are discussed with a primary focus on relative difference or ratio measures. Following the description of these different types of measures, methods for estimation, including logistic regression, random forest and random forest probability machines are outlined.

2.1 Risk estimation in biostatistics and epidemiology

Exposures and outcomes need first to be measured quantitatively using absolute measures of disease frequency, such as incidence and prevalence. Incidence is used to indicate a proportion of newly developed cases of a disease or outcome, whereas prevalence measures the frequency of an existing outcome either at one point in time, denoted point prevalence, or during a given period, denoted period prevalence (Szklo and Nieto, 2014). Their association can be evaluated by calculating various measures of association or effect, particularly, absolute difference measures and relative difference or ratio measures. Examples of absolute difference measures include attributable risk or risk difference and number needed to treat, whereas examples of relative difference or ratio measures include risk ratio and odds ratio. Lastly, the impact of the removal of an exposure on the outcome can be evaluated by considering two measures of potential impact; impact of exposure removal on exposed and impact of exposure removal on population.

For the subsequent sections, refer to the following typical 2×2 epidemiological table with standard notation depicted in Figure 2.1. In this table there are a total of $(a+b)$ exposed and $(c+d)$ nonexposed individuals. a exposed and c nonexposed individuals develop the outcome of interest, whereas b exposed and d nonexposed individuals do not develop the outcome of interest.



OUTCOME

		Yes	**No**	*Total*
EXPOSURE	**Present**	a	b	a + b
	Absent	c	d	c + d
	Total	a + c	b + d	

Figure 2.1: Notation and setup for a standard 2×2 contingency table

2.1.1 Measures of association and potential impact

Absolute difference measures

When the researcher is not primarily interested in how strongly the exposure is associated with a particular outcome, but rather the real impact of exposure on the incidence of outcome in a specific population, absolute difference measures are used. When exposures are harmful, a common absolute difference measure is the attributable risk (AR) or risk difference (RD). AR or RD is described as the difference between the incidence rates in exposed and nonexposed groups. In other cases the exposure may be protective, so an equivalent measure to AR or RD is the absolute risk reduction (ARR), and is expressed as the difference in incidence rates in nonexposed and exposed groups. An alternative absolute difference measure often used in assessing the effectiveness of a treatment or exposure is the number needed to treat (NNT). This measure describes the number of individuals who would need to receive a specific treatment or exposure, on average, for one individual to benefit from the exposed treatment (Szklo and Nieto, 2014).

Relative difference or ratio measures

Relative difference measures estimate the extent of an association between an exposure and an outcome. Relative difference measures that use ratios to compare the frequency of an outcome include risk ratios and odd ratios (or collectively referred to as measures of relative risk). These ratios indicate how much more likely it is that an exposed individual will develop the outcome compared with an unexposed individual. If the relative risk is greater than one, then exposed individuals are at greater risk, less than one, then exposed individuals are at lower risk, and equal to one, then there exists no difference in risk between exposed and nonexposed individuals.

The risk ratio or relative risk (RR) of developing a specific outcome is expressed as the ratio of the risk (or incidence) in the exposed group to that in the nonexposed group. Risk estimates for the exposed and nonexposed groups are $\frac{a}{a+b}$ and $\frac{c}{c+d}$ respectively and thus the RR can be calculated as follows

$$\begin{aligned} RR &= \frac{\text{Risk of outcome in exposed}}{\text{Risk of outcome in nonexposed}} \\ &= \frac{a/(a+b)}{c/(c+d)}. \end{aligned}$$

The odds ratio (OR) is a measure of the relative probabilities of outcome or disease. The odds ratio compares the odds of the outcome among exposed individuals divided by the odds of disease among nonexposed individuals.

$$\begin{aligned} OR &= \frac{\mathrm{P(outcome|exposed)}/(1-\mathrm{P(outcome|exposed)})}{\mathrm{P(outcome|nonexposed)}/(1-\mathrm{P(outcome|nonexposed)})} \\ &= \frac{a/b}{c/d} \\ &= \frac{ad}{bc}. \end{aligned}$$

The odds ratio measure is appropriate for assessing the strength of the association between exposure and outcome variables in case-control studies (first two groups are identified where one is known to have the outcome and the other is known to not have the outcome and then the researcher traces back to investigate exposure). The risk ratio measure described above is more appropriate for measuring the association in cohort studies (the study population known to not have the outcome of interest is first identified by the exposure of interest and followed in time until the outcome of interest occurs) (Szklo and Nieto, 2014).

Measures of potential impact

Attributable risk or risk difference is an absolute difference measure of association that measures the excess incidence of the outcome that can be attributed to the exposure. Considering measures of potential impact, attributable risk percent (AR%) is the percent of the outcome in exposed individuals that can be attributed to the exposure. Population attributable risk (PAR) is a measure often used when researchers wish to apply measures of attributable risk from an epidemiological study to a real population. PAR is defined as the incidence of outcome in the population that can be attributed to the exposure of interest. A similar measure derived from PAR is the population attributable fraction (PAF) and is defined as the proportion of the outcome in the population that is attributable to the exposure (Szklo and Nieto, 2014).

2.2 Logistic regression

Logistic regression is a common model used to describe relationships between a binary outcome and one or more independent predictors. This model can be used to examine the conditional probability of the outcome of interest, given the set of predictors, and also evaluate the predictor effect size estimates using conditional odds ratios. In the following section the logistic regression model is introduced with a brief discussion on the additive and multiplicative interaction measures that can be obtained.

2.2.1 Theory and estimation

Consider the data set $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$ where each observation i has a p dimensional vector $\boldsymbol{x}_i$ and a binary outcome y_i. The observed binary response y_i of the random variable Y_i can take on values of 0 or 1 (where $y_i = 1$ indicates the occurrence of the event of interest) with corresponding probabilities π_i and $1 - \pi_i$ respectively. The response Y_i for each observation i has the following Bernoulli distribution

$$p(Y_i = y_i) = \pi_i^{y_i}(1 - \pi_i)^{1-y_i},$$

where the parameters $\boldsymbol{\pi} = (\pi_1, ..., \pi_n)^T$ are estimated from the data. Simply using a linear regression model to relate the probabilities π_i to the observed predictors $\boldsymbol{x}_i$, say

$$\pi_i = \boldsymbol{x}_i^T \boldsymbol{\beta},$$

where $\boldsymbol{\beta}$ is a vector of regression coefficients, allows π_i to take on any real value. This cannot guarantee that the predicted values π_i will be in the correct range of $(0, 1)$. To circumvent this problem and to map probabilities from the restricted range to the entire real line, the probability is transformed by first converting to odds and then taking the logarithm,

$$\eta_i = \text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right). \qquad (2.1)$$

Solving for π_i in (2.1) gives

$$\pi_i = \text{logit}^{-1}(\eta_i) = \frac{e^{\eta_i}}{1 + e^{\eta_i}}. \qquad (2.2)$$

By assuming $\text{logit}(\pi_i)$ rather than the probability itself follows a linear model, the logistic regression model can be defined as

$$\eta_i = \text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \boldsymbol{x}_i^T \boldsymbol{\beta}, \qquad (2.3)$$

with inverse relationship defined as

$$\pi_i = \frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}. \tag{2.4}$$

The regression coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T$ are estimated using the method of maximum likelihood. Since $Y_i \sim Bern(\pi_i)$ and assuming $Y_1, ..., Y_n$ are independent, the likelihood function is given by

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i}.$$

Taking logs, the log-likelihood function is given by

$$\begin{aligned} \ell(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \Big(y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i) \Big) \\ &= \sum_{i=1}^{n} \Big(y_i \log\left(\frac{\pi_i}{1 - \pi_i} \right) + \log(1 - \pi_i) \Big). \end{aligned} \tag{2.5}$$

Using the inverse relationship defined in (2.4), an expression for $1 - \pi_i$ is

$$1 - \pi_i = 1 - \frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}} = (1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}})^{-1}. \tag{2.6}$$

A final expression for the log-likelihood function is obtained by substituting (2.3) and (2.6) in (2.5), giving,

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \big(y_i \, \boldsymbol{x}_i^T \boldsymbol{\beta} - \log(1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}) \big). \tag{2.7}$$

The maximum likelihood estimates are obtained by solving the likelihood equations, which result from setting the partial derivatives $\frac{\partial \ell(\boldsymbol{\beta})}{\partial \beta_j}$ to zero, for $j \in \{1, \ldots, p\}$. The partial derivative of the log-likelihood for any parameter β_j is given by,

$$\begin{aligned} \frac{\partial \ell(\boldsymbol{\beta})}{\partial \beta_j} &= \sum_{i=1}^{n} \big(y_i \, x_{ij} - \frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}} \, x_{ij} \big) \\ &= \sum_{i=1}^{n} (y_i - \frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}) \, x_{ij} \\ &= \sum_{i=1}^{n} (y_i - \pi_i) \, x_{ij}, \end{aligned}$$

where π_i depends on both the covariates $\boldsymbol{x}_i$ and the regression coefficients $\boldsymbol{\beta}$ through the logit transformation of (2.3). These likelihood equations cannot be solved explicitly and so a method called iteratively re-weighted least squares (IRLS) is used to obtain $\hat{\boldsymbol{\beta}}$ numerically. The regression coefficients $\boldsymbol{\beta}$ in the model can be interpreted as log odds ratio estimates. For example, the parameter β_j is the change in the log odds of the outcome per unit change in x_j holding all other predictors constant. By exponentiating β_j, a much more intuitive value can be obtained, namely the odds ratio, which was described in section 2.1.1. If x_j is binary, the odds ratio $exp(\beta_j)$ is the odds for $x_j = 1$ compared with the odds when $x_j = 0$.

Other previously mentioned popular quantities of interest in epidemiological studies are risk ratios. Logistic regression can only be used to obtain odds ratio estimates and these estimates will approximate risk ratios only if the outcome is rare ($\leq 10\%$). When the outcome is rare, and referring to Figure 2.1, both a will be much smaller than b, and c will be much smaller than d. This means that $a + b \approx b$ and $c + d \approx d$ and thus,

$$RR = \frac{a(c+d)}{c(a+b)} \approx \frac{ad}{bc} = OR.$$

2.2.2 Additive and multiplicative interactions

The logistic regression model can also be used to evaluate both multiplicative and additive interaction effects among predictors. For simplicity, the case where both predictors under consideration are binary is considered. Since only odds ratios can be evaluated from the logistic regression model, interaction measures are evaluated on the odds ratio scale. A multiplicative interaction measures the extent to which the effect of both the predictors together exceeds the product of the effects of the two predictors considered separately. A multiplicative interaction measure on the odds ratio scale is the ratio of odds ratios (ROR) defined as

$$ROR = \frac{OR_{11}}{OR_{10}\, OR_{01}},$$

where OR_{10} and OR_{01} are exponentiated main effects, OR_{11} is the exponentiated sum of the main and interaction effects, and the whole quantity is the exponentiated interaction effect obtained from the model. If $OR_{11}/OR_{10}OR_{01} > 1$, the multiplicative interaction is said to be positive and if $OR_{11}/OR_{10}OR_{01} < 1$, the interaction is said to be negative.

An additive interaction measures the extent to which the effect of the two predictors together exceeds the effect of each considered individually. A measure of additive interaction on the odds ratio scale, called the relative excess risk due to interaction ($RERI_{OR}$), can also be estimated using the parameters of a logistic regression model. The $RERI_{OR}$ is defined as

$$RERI_{OR} = OR_{11} - OR_{10} - OR_{01} + 1.$$

If $RERI_{OR} > 0$, the additive interaction is said to be positive and if $RERI_{OR} < 0$, the interaction is said to be negative.

Although the logistic regression model is widely used for the analysis of binary outcomes, its application comes with several drawbacks. Logistic regression requires that the model be correctly specified, meaning that the user must exactly specify which predictors appear and how they interact with each other. This may be challenging for the researcher, and if the model is misspecified, both predictions and effect size estimates may be biased (Malley et al., 2012). Another drawback of the logistic regression model is that only estimates of odds ratios can be obtained. When risk ratios are the quantities of interest and cannot be obtained directly, the odds ratio is calculated and often interpreted as a risk ratio. However, only when the prevalence of the outcome is low (<10%) does the odds ratio approximate the risk ratio. The odds ratio will overestimate the risk ratio when the risk ratio is greater than 1, and will underestimate the risk ratio when it is less than 1 (Zhang and Yu, 1998).

2.3 Decision Trees and random Forests

The classification and regression tree (CART) model was first introduced by Breiman et al. (1984) and provides solutions to regression and classification problems that are both easily interpreted and can be clearly displayed graphically. CART machine learning algorithms, such as decision trees, involve recursively splitting the predictor space into smaller regions and using these regions to predict the response for a new observation. Predicting the response for a new observation typically involves using the mean of the training observations in the region to which the new observation belongs. Building on the notion of decision trees, Breiman (1996) introduced ensemble methods such as bagging and random forests that combine the predictions from multiple decision trees in order to make more accurate predictions than those from any individual tree.

Decision trees can be applied to both regression and classification problems. Typically, regression trees are used when the response is a continuous variable, whereas classification trees are more often used when the response is discrete. In the following section, the algorithm for building both types of decision trees and their limitations in response prediction is discussed. Following decision tree construction, the random forest and bagging algorithms are presented and differences between the two are highlighted.

2.3.1 Regression and classification decision trees

Since the classification and regression tree building algorithms differ only slightly in a couple of steps, the regression tree algorithm is first presented, and then alterations to the algorithm for building classification trees are considered.

Consider the data set $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$ where each observation i has a p dimensional predictor vector $\boldsymbol{x}_i$ and an outcome y_i with values on the real number line. In the first step in constructing a regression tree, the predictor space is divided. In this step, the goal is to take the predictor space, defined as the set of all possible values for $x_1, x_2, ..., x_p$, and split it into J distinct and non-overlapping regions, denoted $R_1, R_2, ..., R_J$. These regions can be thought of as high dimensional boxes that minimize the residual sum of squares (RSS) defined as

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{R_j}$ is the mean response for the observations within the j^{th} region.

The way in which in the predictor space is partitioned into regions that minimize the RSS is known as recursive binary splitting. Beginning at the top of the tree (all observations in one region), the predictor space is successively split into two new regions, which results in the formation of two new branches. The regions continue to be split until a certain stop criterion is met. At each potential split into two new branches, the best split is made at that particular point in time rather than a split that will lead to a better tree at a later step. The steps in regression tree construction using recursive binary splitting are as follows:

1. Consider all possible predictors $(x_1, ..., x_p)$ and all possible values for the cutpoint c for each of these predictors. Choose predictor x_j and cutpoint c such that splitting the predictor space into two distinct regions, $\{x|x_j < c\}$ and $\{x|x_j > c\}$, leads to the greatest possible reduction in the residual sum of squares. In other words, for any value of $j \in \{1, \ldots, p\}$ and cutpoint c, define the pair of regions R_1 and R_2,

   $$R_1(j, c) = \{x|x_j < c\} \text{ and } R_2(j, c) = \{x|x_j \geq c\},$$

   and choose j and c so as to minimize

   $$\sum_{i:x_i \in R_1(j,c)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,c)} (y_i - \hat{y}_{R_2})^2,$$

   where $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ are the mean responses for the observations in $R_1(j, c)$ and $R_2(j, c)$ respectively.

2. Split one of the previously identified regions that results in the smallest reduction in RSS by repeating the process described in step 1.

3. Continue splitting the regions until a stopping criterion is reached. A stopping criterion may be to continue until all regions contain no more than five observations.

Once all the regions or terminal nodes are defined, the resulting regression tree can be used to predict the response for a new observation. The predicted response for any new observation is found by taking the mean response of the observations in the terminal node in which the new observation resides. In many cases, the resulting regression tree is too complex, leading to poor prediction performance. One solution to this problem is to use tree pruning techniques such as reduced error pruning or cost complexity pruning, which essentially take the resulting regression tree and remove terminal nodes that do not provide additional information.

Building a classification tree is very similar to building a regression tree except that the response variable is often qualitative rather than quantitative. Again, the goal is to split the predictor space into J distinct and non-overlapping regions, denoted $R_1, R_2, ..., R_J$. However, the residual sum of squares is no longer used as the split criterion. For making binary splits, measures such as the Gini index (G) or cross-entropy (D) are used to evaluate the quality of a particular split. These two measures are defined as

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

and

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}),$$

where $\hat{p}_{mk}$ is the proportion of observations in the m^{th} region that are from the k^{th} class. If all of the $\hat{p}_{mk}$s are close to 0 or 1, both the Gini index and cross-entropy will take a small value. These measures can be thought of as measures of node purity, meaning that a node contains predominantly observations from a single class. The best split is one that leads to an increase in node purity or minimizes either G or D. Since the response is qualitative for classification trees, the predicted response is the majority vote or the most commonly occurring class of the observations in the terminal node in which the new observation resides.

Classification and regression trees are widely used since they are very easily interpreted and represented graphically. However, these methods can be limited in their predictive accuracy compared to other regression and classification approaches. Decision trees can be very non-robust, meaning that a small change in the data set used to construct the tree can have a large effect in the final estimated structure of the tree. The predictive accuracy and robustness of decision trees can be improved by aggregating many decision trees, and using machine learning ensemble methods such as bagging and random forests.

2.3.2 Random forests

In order to improve the predictive accuracy and robustness of decision trees, Breiman (1996) introduced the ensemble method bagging or bootstrap aggregation. Bagging is a common statistical procedure for reducing the variance of a statistical learning method. An undesirable feature of decision trees is that the building procedure could potentially yield quite different tree structures if it is applied repeatedly to distinct data sets. Thus, bagging can be used to reduce this inherent high variance of the method. Building on the bagging method, Breiman (2001) proposed the random forest technique, which adds an additional layer of randomness that decorrelates the trees. A collection of bagged trees could look very similar if there exists a very strong predictor in the data set, since most or all trees will use this predictor in the top split. If all bagged trees have a similar tree structure, the predictions from the bagged trees will be highly correlated and not much improvement will be made over the use of a single tree. Random forests provide a solution to this problem by considering only a random subset of the predictors at each split. In both bagging and random forests, a fixed number of trees are constructed, each using a different bootstrap sample of the data. In bagging, each node is split using the best split among all predictors, whereas in random forest, each node is split using the best among a subset of predictors randomly chosen at that node. The algorithm for both ensemble methods is illustrated below.

Consider a training data set drawn from a sample of independently and identically distributed random variables $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$, where each observation i has a p dimensional predictor vector $\boldsymbol{x}_i$ and an outcome y_i taking values on the real number line. The random forest (and bagging) algorithm for both classification and regression is as follows:

1. Draw a bootstrap sample b of size n from the training data set where b is drawn with replacement. Any observations in the original training data set not drawn in b are denoted “out-of-bag” (OOB) observations.

2. Grow an unpruned classification or regression tree from the bootstrap sample b using the recursive binary splitting procedure described in section 2.2.1 until a minimum node size is achieved in each node. For the bagging ensemble method, consider at each node the best split among all predictors, whereas for random forests, choose the best split from among a random sample of predictors. For a continuous response, the best split is the one that minimizes the residual sum of squares (or the mean square error). For a discrete response, the best split is the one that minimizes a dichotomous purity measure (Gini index or cross-entropy).

3. For a regression tree, calculate the mean response in each terminal node of the tree; for a classification tree, calculate the majority vote in each terminal node of the tree.

4. Repeat steps 1-3 until the desired number of trees have been constructed.

5. Predict the response for a new observation by dropping it down each tree until it resides in a terminal node and aggregate the predictions of all trees. For regression random forest take an average of the mean responses in the terminal nodes (step 3) in which the new observation resides. For classification random forest, take the majority vote of the terminal nodes (step 3) in which the new observation resides.

2.4 Probability machines

Machine learning methods such as random forest are often used in binary classification problems due to their good discriminatory performance. A similar problem, when group membership is not entirely the goal, is the problem of estimating the probability of group membership. Classical parametric methods such as logistic regression have been widely used for probability estimation, but come with several drawbacks as discussed in section 2.2. Malley et al. (2012) propose a solution to this problem by simply treating it as a non-parametric regression problem. Their solution involves using readily available machine learning methods to estimate the conditional probability function for a binary outcome. They refer to such learning machines as probability machines (PMs).

A PM produces, on the individual level, a predicted conditional probability of success given a set of predictors. This predicted probability is calculated without imposing any restrictions on the structure of the predictors or the distribution of the data. Using individual predicted conditional probabilities of success, various effect measures can be estimated both on the individual level and for specific groups of observations. Malley et al. (2012) have studied non-parametric regression machines, including random forest regression, and have shown these regression machines to have provable consistency properties under fairly general conditions. They refer to the use of random forest regression as probability machines as random forest probability machines (RFPMs).

Estimating potential or counterfactual outcomes is often of interest to researchers. When using the potential outcome framework, the outcome under each possible value of a predictor, say x_1, must be observable. However, only the outcome under the actual value for x_1 is observed, whereas the potential outcomes under the other possible values for x_1 are considered to be missing data. Dasgupta et al. (2014) propose that using probability machines, counterfactual outcomes in the context of a binary outcome can be directly observed. In the following section, the use of random forest as probability machines introduced by Malley et al. (2012) is discussed. Further consideration of the potential outcome framework, and methods proposed for estimating potential outcomes including G-computation, Imbens' method, and the two-machine RFPM are outlined in subsequent sections.

2.4.1 Random forest probability machines

In order to use the random forest algorithm developed by Breiman (2001) to estimate the conditional probability function for a binary outcome, slight alterations must be made. Consider a training data set drawn from a sample of independently and identically distributed random variables $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$ where each observation i has a p dimensional predictor vector $\boldsymbol{x}_i$ and a binary outcome y_i. As in the usual random forest procedure, a test subject is dropped down each tree in the forest until it resides in a terminal node. The predicted response for a new observation can then be found by either taking an average of the mean responses in the terminal nodes for regression random forest, or taking the majority vote of the terminal nodes for classification random forest.

In methods involving random forest probability machines, the goal is to estimate conditional probabilities rather than predict an expected response. In regression RFPMs, each tree provides a conditional probability estimate, which is obtained by taking the proportion of observations in the training data set with an outcome value of 1 in the residing node. The final probability estimate is obtained by taking an average of all the individual tree estimates in the forest. The general procedure for regression random forest as a probability machine is as follows:

1. Draw a bootstrap sample b of size n from the training data set where b is drawn with replacement. Any observations in the original training data set not drawn in b are denoted "out-of-bag" (OOB) observations.

2. Grow a regression tree using the bootstrap sample b until a minimum node size is achieved. This is done using the recursive binary splitting procedure described in section 2.3.1 and using the random forest method, the best split at each node is determined by using a random sample of the predictors.

3. Calculate the proportion of 1's in each terminal node of the tree.

4. Repeat steps 1-3 to grow a specific number of trees.

5. The predicted probability of success for an observation is obtained by first dropping the observation down each tree in the random forest until it resides in a terminal node. The proportion of 1's in these final nodes are then calculated. The probability estimate of success for an observation is taken as the average of the proportion of 1's over all trees.

2.4.2 Estimating counterfactual outcomes

Consider a situation where there exists two groups of subjects, and each subject is identical to another in the other group except for the value of one predictor, say x_1, where x_1 is binary. In this ideal situation, it would be feasible to directly observe the change in outcome in each observation that results from changing the value of x_1 by simply considering that observation's counterpart in the opposite group. In a more realistic situation, each subject can only take on a single value for x_1 and as a result, only one of two potential outcomes $Y_{x_1=0}$ and $Y_{x_1=1}$ is observed. These two variables $Y_{x_1=1}$ and $Y_{x_1=0}$ are termed potential outcomes or counterfactual outcomes because for each subject, one of $Y_{x_1=1}$ or $Y_{x_1=0}$ describes the subject's true outcome value (observed effect), and the other describes the outcome that would have been observed in a situation that did not happen (counterfactual effect). Individual causal effects can then be defined as the difference in a subject's counterfactual outcomes, $ICE = Y_{x_1=1} - Y_{x_1=0}$. However, these individual causal effects cannot be calculated exactly for any individual since both $Y_{x_1=1}$ and $Y_{x_1=0}$ are not observed (Hernan, 2004).

Although both $Y_{x_1=1}$ and $Y_{x_1=0}$ are not observed, predictive models can be used to judge reasonably well how one's counterpart will behave. Several approaches have been proposed to estimate counterfactual outcomes, two of which are regression-based approaches, and are called G-Computation and Imbens' method. The G-Computation method, proposed by Snowden et al. (2011), uses a single multivariable regression model to regress the outcome on baseline covariates. One baseline covariate say, x_j, is binary and estimating the counterfactual outcomes due to a change in this covariate is of primary interest. Using the single regression model, two predicted outcomes are estimated for each subject; one as if that subject had a x_j value of zero and the other as if that subject had a x_j value of one. Individual causal effect (ICE) estimates of x_j for each subject can be calculated using the two predicted outcomes.

The other regression-based approach proposed to estimate counterfactual outcomes was introduced by Imbens (2004). In this method, the outcome is regressed on the baseline covariates using two models, say I_0 and I_1, where I_0 is fit using the data of observations with $x_j = 0$, and I_1 is fit using the data of observations with $x_j = 1$. Using I_0 and I_1, the predicted outcome is estimated for all subjects regardless of their value of x_j. The prediction from I_0 is the predicted outcome had that subject expressed a x_j value of zero (even if $x_j = 1$) and the prediction from I_1 is the predicted outcome had that subject expressed a x_j value of one (even if $x_j = 0$). Thus for each subject, the two counterfactual outcomes can be estimated, $\hat{I}_0$ and $\hat{I}_1$ and like G-computation, ICE estimates for x_j for each subject can be obtained.

Both of the above regression-based methods require an explicit statement of the regression model. Since probability machines provide estimates without a particular model structure, they provide an alternative non-parametric approach to a similar problem. The two regression-based methods provide a technique to estimate counterfactual outcomes for a continuous outcome, whereas researchers may be interested in estimating counterfactual probabilities of success for a binary outcome. Similarly to Imbens' regression-based method, regression random forests can be used as the two predictive models rather than two regression models to estimate counterfactual outcomes. When counterfactual probabilities of success rather than counterfactual outcomes are of interest, two random forest probability machines can be used. Dasgupta et al. (2014) call the use of two RFPMs to estimate counterfactual probabilities of success the two-machine RFPM method. The two-machine RFPM method is the primary focus of the following section, and so its use is discussed in detail with simulations to follow in Chapter 3.

2.4.3 Two-machine random forest probability machine

Consider a sample of size n with p binary predictors $(x_1, x_2, \ldots, x_p)$ and a binary outcome y_i. Suppose the main interest is determining individual counterfactual probabilities of success when the value of predictor x_1 is changed. Predicting counterfactual probabilities of success using the two-machine RFPM method is depicted in Figure 2.2 and outlined as follows:

1. Split the data set into two groups based on whether $x_1 = 0$ or $x_1 = 1$. Denote these subgroups G_1 and G_0.

2. Within each subgroup, train a random forest probability machine ($RFPM_0$ and $RFPM_1$) on the remaining $p-1$ predictors $(x_2, x_3, \ldots, x_p)$.

3. Obtain observed and counterfactual probabilities of success for each observation by predicting from the two RFPMs. For an observation with $x_1 = 1$, its observed probability of success will be predicted from $RFPM_1$ and its counterfactual probability of success will be predicted from $RFPM_0$, and vice versa for an observation with $x_1 = 0$.

For each observation i, an observed and a counterfactual probability of success are predicted: $p_{1i}(y = 1|x_2, x_3, \ldots, x_p)$ and $p_{0i}(y = 1|x_2, x_3, \ldots, x_p)$. This enables for the calculation of various effect measures for each observation i, e.g.

Risk ratio

$$RR_i = \frac{p_{1i}}{p_{0i}}$$

Odds ratio

$$OR_i = \frac{p_{1i}/1 - p_{1i}}{p_{0i}/1 - p_{0i}}$$

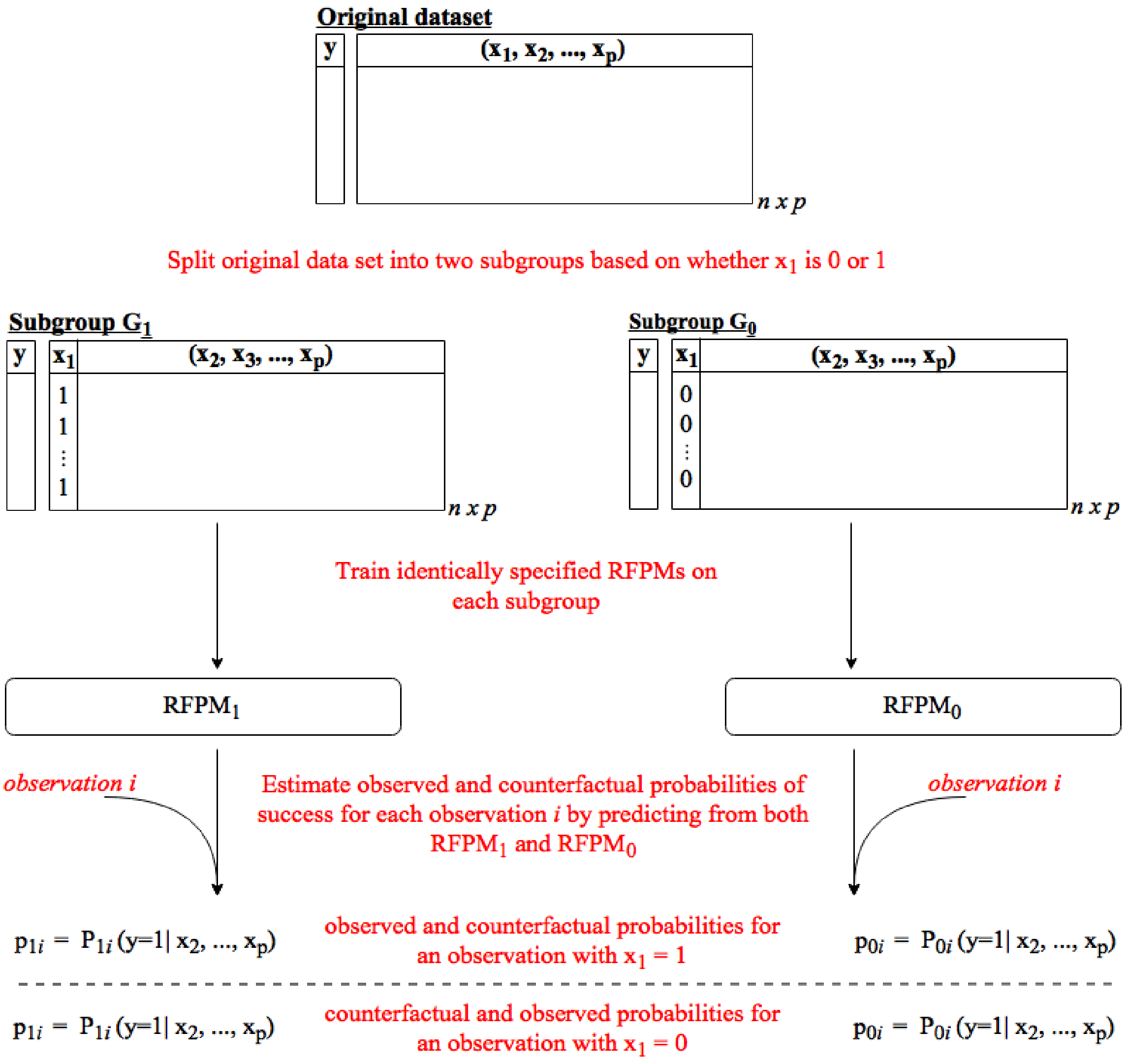

Original dataset
y
(x1, x2, ..., xp)
n x p
Split original data set into two subgroups based on whether x1 is 0 or 1
Subgroup G1
y
x1
(x2, x3, ..., xp)
1
1
⋮
1
n x p
Subgroup G0
y
x1
(x2, x3, ..., xp)
0
0
⋮
0
n x p
Train identically specified RFPMs on each subgroup
RFPM1
RFPM0
observation i
Estimate observed and counterfactual probabilities of success for each observation i by predicting from both RFPM1 and RFPM0
observation i
p1i = P1i (y=1| x2, ..., xp)
observed and counterfactual probabilities for an observation with x1 = 1
p0i = P0i (y=1| x2, ..., xp)
p1i = P1i (y=1| x2, ..., xp)
counterfactual and observed probabilities for an observation with x1 = 0
p0i = P0i (y=1| x2, ..., xp)

Figure 2.2: Procedure for predicting individual observed and counterfactual probabilities of success using the two-machine RFPM method

Attributable risk

$$AR_i = p_{1i} - p_{0i}.$$

Risk estimates for the full sample are calculated by averaging over the individual estimates

$$RR_{sample} = \frac{\sum_{i=1}^{n} RR_i}{n},$$

$$OR_{sample} = \frac{\sum_{i=1}^{n} OR_i}{n},$$

or

$$AR_{sample} = \frac{\sum_{i=1}^{n} AR_i}{n}.$$

When the underlying generative model is logistic, and the correct model is specified, the sample odds ratio estimates obtained using the two-machine RFPM method are equivalent to the $\hat{\boldsymbol{\beta}}$ coefficients obtained using logistic regression. Since probability machines provide non-parametric estimates, different estimates will be obtained using an incorrectly specified logistic model.

Chapter 3

Tuning parameter optimization for two-machine RFPM method

The two-machine RFPM method outlined in Chapter 2 is implemented in R using the *randomForest* (Liaw and Wiener, 2002) package. Although the outcome is binary, it is stored as a numeric variable rather than a factor variable, so as to apply regression random forest (typically classification random forest is used for a discrete outcome). Conditional probabilities of success for classification random forest are estimated as the proportion of component trees which classify the result as a success. Malley et al. (2012) have found classification random forest to be far less efficient at probability estimation than regression random forest, and the consistency of the probability estimates produced has yet to be demonstrated.

There are several tuning parameters that can affect the construction of a random forest including *mtry*, *ntree*, and *nodesize*. The *mtry* parameter controls the number of randomly selected predictors used to determine the best split at each node, and typically remains constant throughout the tree building process. The parameter *ntree* specifies the number of trees to be used to construct the forest, and *nodesize* is the minimum size of the terminal nodes. Here, *nodesize* acts as a stopping criterion, meaning that once a minimum node size is achieved, no further splits will be performed on this node.

After several uses of the two-machine RFPM method to estimate a number of main effects and corresponding 95% confidence intervals, it became clear that the computation time is an issue, especially when the sample size is large. For example, consider a data set of size $n = 50000$ where both the outcome and the predictors $x_1, x_2, \ldots, x_{10}$ are binary. Suppose three main effects are estimated with tuning parameters set at $mtry = 9$ (all predictors used in random forest construction), $ntree = 500$, and $nodesize = 250$. This is the most computationally intensive simulation setting that is considered, and the computation time required to run this setting once on a 4-core

CPU is approximately 300 seconds. This computation time gets larger as the number of main effects needed to estimate, and the number of predictors in the data set increases. Running this simulation setting 1000 times is very time consuming and unfeasible.

In order to decrease the computation time needed to both run each simulation and compute individual main effect estimates, parallel computing is implemented through use of the R packages *doParallel* (Analytics and Weston, 2015a) and *foreach* (Analytics and Weston, 2015b). Parallel computing allows both main effect estimate computations and simulations to be executed simultaneously rather than sequentially. Alternative ways to reduce computation times such as the R package *ranger*, a fast implementation of random forests for high dimensional data (Wright and Ziegler, 2015), were explored, but produced computation times similar to parallelelizing the *randomForest* commands using *doParallel* and *foreach*. For the reasons given above, simulations are run using a 48-core processor and any computation times reported reflect this.

3.1 Simulations

In order to illustrate the two-machine RFPM method and compare its performance to that of logistic regression when the generative model is truly logistic, consider Model 1

$$\text{logit}(p) = \beta_0 + \log(1.2)\, x_1 + \log(1.5)\, x_2 + \log(2)\, x_3 + 0\, x_4 + ... + 0\, x_{10},$$

where x_1, x_2 and x_3 are uncorrelated binary predictors with main effects of $\beta_1 = \log(1.2) = 0.182$, $\beta_2 = \log(1.5) = 0.405$, and $\beta_3 = \log(2) = 0.693$. The remaining predictors are independently generated, binary, and have no association with the outcome. The intercept β_0 is chosen so that $P(y = 1) \approx 0.3$ and the predictors are generated to have a randomly selected $P(x_i = 1)$ between 0.05 and 0.95.

3.1.1 Simulation 1: Optimizing tuning parameters for the two-machine RFPM method

In Simulation 1, the effect of altering several tuning parameters on the log odds ratio estimates obtained using the two-machine RFPM method is demonstrated. The tuning parameters considered are: *mtry*, the number of predictors randomly selected at each node; *ntree*, the number of trees in the random forest; *nodesize*, the minimum node size; and n, the sample size. Values considered for these parameters are: for *mtry*, the suggested value $\lceil\sqrt{p}\rceil = 3$, the largest integer greater than $\sqrt{p}$, and 9; for *ntree*, 100, 200 and 500; for *nodesize*, 250 and 5% of the sample size; and for sample size, 5000, 10000, and 50000. Bias in the estimates is measured using % relative bias and is calculated as

$$\%\text{ relative bias} = \frac{\bar{\hat{\theta}} - \theta}{\theta} \times 100,$$

where $\bar{\hat{\theta}}$ is the mean of the simulated estimates and θ is the true value.

For each combination of parameter values (n, *nodesize*, *mtry*, and *ntree*), 1000 simulations are completed. Predictors are kept fixed under each choice of n, and a new outcome vector is generated each time. As discussed in section 2.4.3, the two-machine RFPM method can be used to estimate several risk measures, including log odds ratios, risk ratios, and attributable risks, but only estimation of log odds ratios is considered. Subject-specific log odds ratio estimates are calculated using counterfactual probability estimates, and the main effect log odds ratio estimates for β_1, β_2, and β_3 are obtained by averaging over the subject-specific estimates. The results are summarized in Table 3.1 where $ns = nodesize$, $\%bias = \%$ relative bias and the β values are log(1.2), log(1.5), and log(2), respectively.

Table 3.1: Sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias ($\%bias$) of log odds ratio estimates, β_1, β_2, and β_3, from Model 1 using various parameter specifications of the two-machine RFPM method. For completeness, two simulations are reported for n =5000 at ns =250 =5% of 5000.

Simulation setting				$\beta_1 = \log(1.2) = 0.182$			$\beta_2 = \log(1.5) = 0.405$			$\beta_3 = \log(2) = 0.693$		
n	ns	$mtry$	$ntree$	mean	$s_{\hat{\beta}_1}$	$\%bias$	mean	$s_{\hat{\beta}_2}$	$\%bias$	mean	$s_{\hat{\beta}_3}$	$\%bias$
5000	250	3	100	0.179	0.076	-1.78	0.403	0.068	-0.61	0.683	0.125	-1.49
			200	0.175	0.074	-3.78	0.405	0.065	-0.08	0.681	0.120	-1.70
			500	0.178	0.075	-2.43	0.405	0.069	-0.00	0.685	0.124	-1.21
		9	100	0.181	0.078	-0.94	0.407	0.069	0.31	0.686	0.122	-1.07
			200	0.176	0.076	-3.49	0.410	0.069	1.19	0.697	0.125	0.51
			500	0.181	0.075	-1.01	0.408	0.068	0.58	0.692	0.123	-0.14
	5%	3	100	0.177	0.074	-2.99	0.404	0.068	-0.27	0.680	0.123	-1.95
			200	0.179	0.076	-1.67	0.405	0.067	-0.12	0.683	0.123	-1.42
			500	0.178	0.075	-2.47	0.405	0.067	-0.23	0.687	0.122	-0.94
		9	100	0.184	0.073	0.66	0.407	0.067	0.46	0.692	0.127	-0.14
			200	0.187	0.074	2.34	0.408	0.067	0.55	0.696	0.124	0.48
			500	0.184	0.075	0.97	0.409	0.069	0.85	0.691	0.123	-0.28
10000	250	3	100	0.189	0.054	3.44	0.403	0.053	-0.56	0.691	0.044	-0.38
			200	0.189	0.055	3.57	0.407	0.052	0.45	0.694	0.047	0.11
			500	0.187	0.053	2.69	0.407	0.053	0.44	0.692	0.045	-0.14
		9	100	0.187	0.056	2.27	0.408	0.055	0.67	0.695	0.046	0.33
			200	0.184	0.057	0.89	0.407	0.055	0.35	0.699	0.045	0.77
			500	0.181	0.056	-0.62	0.409	0.054	0.77	0.696	0.046	0.48
	5%	3	100	0.191	0.056	4.63	0.405	0.054	-0.23	0.692	0.046	-0.19
			200	0.191	0.055	4.74	0.406	0.054	0.08	0.692	0.047	-0.17
			500	0.189	0.054	3.91	0.407	0.054	0.27	0.691	0.046	-0.37
		9	100	0.185	0.057	1.43	0.405	0.053	-0.12	0.692	0.046	-0.18
			200	0.184	0.057	1.00	0.409	0.055	0.78	0.694	0.045	0.16
			500	0.184	0.056	1.05	0.408	0.054	0.70	0.697	0.046	0.48
50000	250	3	100	0.182	0.020	-0.19	0.406	0.031	0.02	0.693	0.021	0.04
			200	0.181	0.020	-0.65	0.405	0.028	-0.07	0.695	0.021	0.23
			500	0.181	0.020	-0.50	0.404	0.029	-0.26	0.693	0.022	-0.04
		9	100	0.183	0.021	0.50	0.408	0.030	0.53	0.697	0.021	0.57
			200	0.184	0.021	0.69	0.407	0.030	0.38	0.697	0.021	0.58
			500	0.182	0.020	-0.12	0.408	0.030	0.62	0.697	0.021	0.56

Simulation setting				$\beta_1 = \log(1.2) = 0.182$			$\beta_2 = \log(1.5) = 0.405$			$\beta_3 = \log(2) = 0.693$		
n	ns	$mtry$	$ntree$	mean	$s_{\hat{\beta}_1}$	$\%bias$	mean	$s_{\hat{\beta}_2}$	$\%bias$	mean	$s_{\hat{\beta}_3}$	$\%bias$
50000 *cont.*	5%	3	100	0.185	0.020	1.43	0.406	0.028	0.15	0.691	0.021	-0.32
			200	0.183	0.020	0.45	0.405	0.030	-0.22	0.691	0.021	-0.33
			500	0.184	0.021	1.07	0.406	0.029	0.13	0.692	0.020	-0.20
		9	100	0.182	0.021	0.00	0.408	0.030	0.56	0.694	0.021	0.10
			200	0.181	0.021	-0.71	0.407	0.031	0.25	0.692	0.020	-0.20
			500	0.183	0.020	0.14	0.409	0.029	0.80	0.695	0.020	0.26

Although the two *nodesize* parameter settings for $n = 5000$ are 250 observations, two simulations are completed for consistency. From Table 3.1, all combinations of parameter settings perform fairly well with estimates reasonably close to the true values. Both the standard deviation of the estimates and the % relative bias decrease with n. Within each sample size, only minimal differences in the estimates are seen when the *nodesize* parameter is decreased from 5% of the sample size to 250 observations. Minimal differences are also seen within *mtry* values by increasing the number of trees used in the construction of the forest. By changing the *mtry* parameter from 3 to 9, a slight decrease in % relative bias is seen for $n = 5000$, but as n gets larger, the difference in bias between the two parameter settings is small. Overall, no combination of parameter settings produced estimates with a % relative bias greater than 5% in absolute value.

The purpose of this simulation is to demonstrate the effect of changing the tuning parameters, and to determine the optimal parameter settings for future simulations. For all combinations of parameter settings, minimal differences are seen in the estimates produced, whereas there are large differences in computation times. As previously mentioned, simulations are run in parallel using a 48-core CPU, and the approximate time to complete each block of simulations (each value of n corresponds to one block) is 35, 115, and 770 minutes for $n = 5000$, 10000, and 50000 respectively. Due to minimal differences in estimates, but large differences in computation times, remaining simulations are run at parameter settings of *nodesize* $= 5\%$, *mtry* $= 3$ and *ntree* $= 100$. It was decided to continue running future simulations using all three values of sample size in order to demonstrate the improvement in precision by increasing the sample size.

3.1.2 Simulation 2: Comparing log odds ratio estimates from the two-machine RFPM method to logistic regression

In Simulation 2, the optimal parameter settings determined in Simulation 1 are used to compare the performance of the two-machine RFPM method to that of logistic regression. Data is generated using the same logistic model (Model 1) given above and for each value of $n = 5000$, 10000, and 50000, a thousand simulations are completed. Similarly to Simulation 1, predictors are generated once, and a new outcome vector is generated each time. Log odds ratio estimates for β_1, β_2, and β_3 using the two-machine RFPM method ($nodesize = 5\%$, $mtry = 3$, and $ntree = 100$) are calculated as an average over the subject-specific log odds ratio estimates. Log odds ratio estimates using logistic regression are obtained using a correctly specified model. The sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias ($\%bias$) for each β estimate from the two-machine RFPM method are reported in Table 3.2. The p-value corresponds to the t-test assessing if the mean estimate value is significantly different from the true value.

Table 3.2: Sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias ($\%bias$) of log odds ratio estimates, β_1, β_2, and β_3, from Model 1 using the two-machine RFPM method. The p-values listed are the results from a t-test comparing the mean RFPM estimate to the true parameter value.

β	n	mean	$s_{\hat{\beta}}$	$\%bias$	p-value
$\beta_1 = \log(1.2) = 0.182$	5000	0.183	0.063	0.43	0.693
	10000	0.182	0.045	-0.42	0.587
	50000	0.182	0.020	-0.01	0.983
$\beta_2 = \log(1.5) = 0.405$	5000	0.381	0.118	-6.13	< 0.001
	10000	0.398	0.045	-1.74	< 0.001
	50000	0.406	0.020	0.11	0.490
$\beta_3 = \log(2) = 0.693$	5000	0.696	0.065	0.40	0.184
	10000	0.691	0.046	-0.37	0.080
	50000	0.686	0.034	-0.96	< 0.001

Figure 3.1 shows the results of the simulation, where the box plots represent the estimates produced using the two-machine RFPM method (green) and logistic regression (yellow). The figure is separated into three plots, where each plot represents

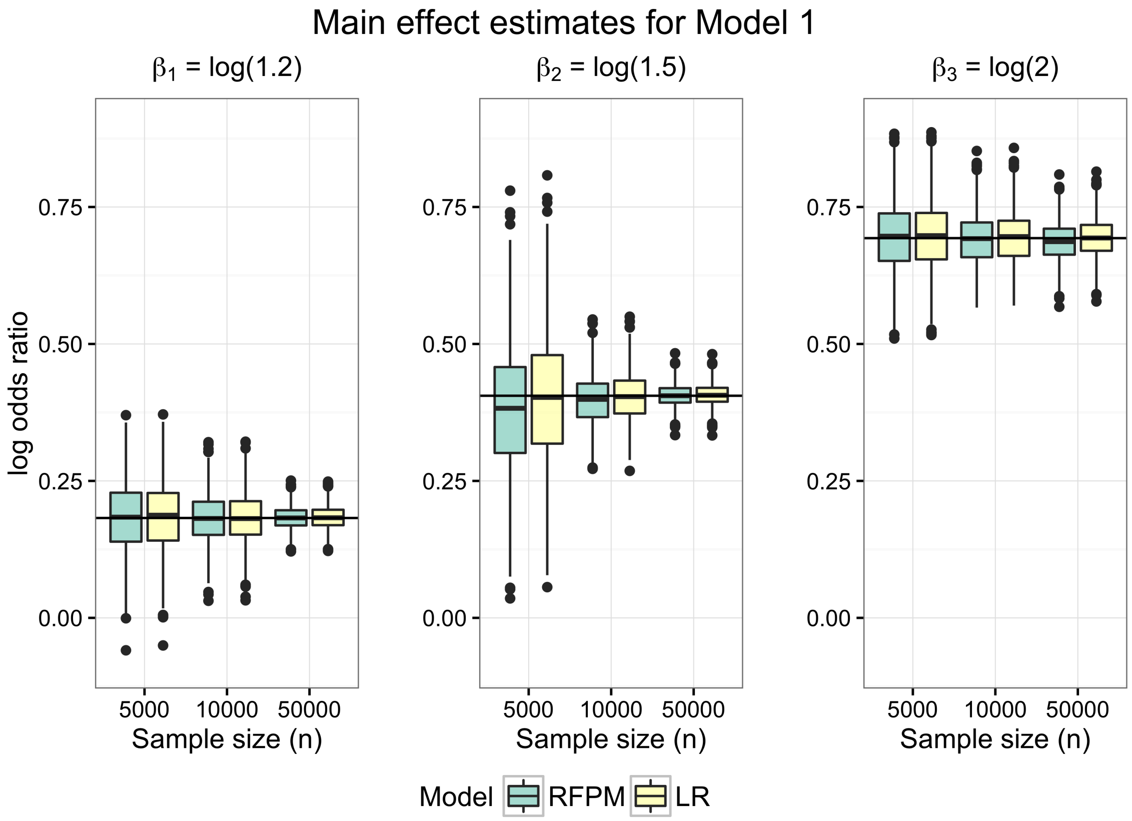


Figure 3.1: Log odds ratio estimates for β_1, β_2, and β_3 from Model 1 using the two-machine RFPM method (green) and logistic regression (yellow). Each plot represents one β estimate and the three sets of box plots are the estimates for the three different samples sizes, $n = 5000$, 10000, and 50000. The true parameter value is indicated in each plot by the horizontal line.

one β estimate. The three sets of box plots per β are the estimates for the three different sample sizes in increasing order. In almost all applications of the two-machine RFPM method, approximately unbiased estimates are produced indicated by the medians of estimated log odds ratios being very close to the true values (shown by the horizontal lines). The two-machine RFPM method performs comparatively to logistic regression in that the variability among the estimates between the two methods is very similar. In this and future simulations, both logistic regression and RFPM methods are used on the same data sets. Figure 3.2 demonstrates the large correlation between the two sets of estimates for two of the simulations.

Using results from Simulations 1 and 2, the two-machine RFPM method produces log odds ratio estimates with minimal bias, and performs with similar efficiency to that

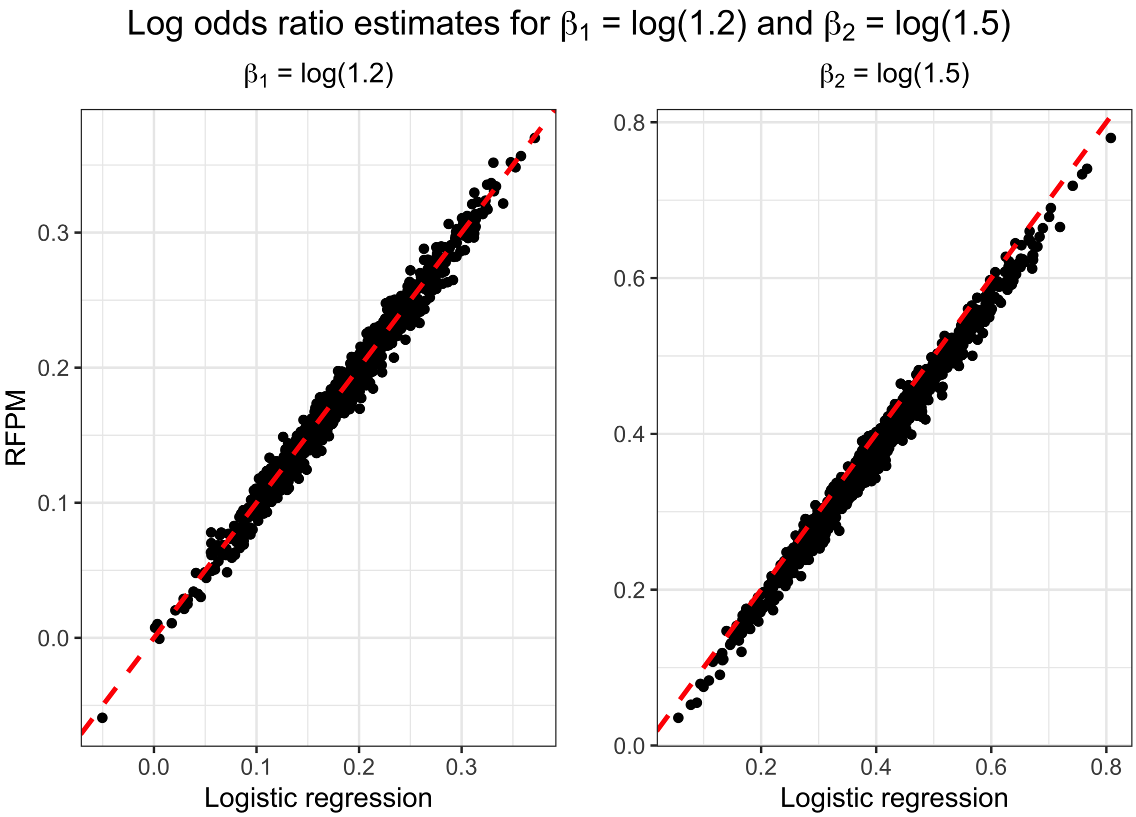


Figure 3.2: Plot of log odds ratio estimates for $\beta_1 = \log(1.2)$ and $\beta_2 = \log(1.5)$ for $n = 5000$ obtained using RFPMs against those obtained using logistic regression. The dashed line indicates the line of equality.

of logistic regression. The two-machine RFPM method is non-parametric and requires no explicit specification of the model, or the data generating process. Correctly specifying the model can be challenging, and an incorrectly specified logistic regression model will produce both inaccurate predictions and biased effect size estimates.

Chapter 4

Detecting and estimating interaction effects for binary predictors using RFPMs

The two-machine RFPM method can be used to estimate log odds ratios, and produces estimates similar to those obtained using logistic regression. It is an effective way to estimate risk effects, is not restricted to data generated using a logistic model, and can be applied to any regression problem with binary outcomes. The estimation of interaction effects, both multiplicative and additive, is now considered.

As discussed in section 2.2.2, measures of multiplicative interaction are easily obtained using logistic regression, whereas additional work is needed to obtain measures of additive interaction when the outcome prevalence is greater than 10%. Additive interaction measures are often relevant in health research, but are less frequently reported than measures of multiplicative interaction. Random forest probability machines provide estimates of counterfactual probabilities of success, which can be used to estimate measures of both multiplicative and additive interaction easily and nonparametrically. In this chapter, two methods pertaining to interaction detection and estimation proposed by Dasgupta et al. (2014) are explored, namely, the intuitive interaction detection method and the four-machine RFPM method. Their use is demonstrated in several simulations, and application of RFPMs to estimate more complicated interactions is briefly discussed.

4.1 Intuitive interaction screening method

In order to avoid computing risk estimates for all possible combinations of predictors, Dasgupta et al. (2014) propose a method to detect the presence of both multiplicative and additive interactions using a single random forest probability machine. Since only a single machine is fit to the data, this method should not be used for direct estimation.

Consider a sample of size n with p binary predictors $(x_1, x_2, \ldots, x_p)$ and a binary outcome y. Suppose the main interest is to determine if there exists a significant interaction between x_1 and x_2. The procedure for detecting such interaction is as follows:

1. Fit a single random forest probability machine to the data, $RFPM_1$.

2. Split the data set into four subgroups based on the four combinations of x_1 and x_2. Denote these subgroups G_{00}, G_{01}, G_{10}, G_{11}, where G_{ij} is the subgroup of observations with $x_1 = i$ and $x_2 = j$. Also, let n_{00}, n_{01}, n_{10}, and n_{11} denote the numbers of observations in each subgroup.

3. Using $RFPM_1$, predict estimates of the conditional probability, $P(y = 1|x_1 = i, x_2 = j, x_3, \ldots, x_p)$, for each observation in the four subgroups. Let p_{ijc} denote the predicted conditional probability estimate for the c^{th} observation in the ij^{th} subgroup.

4. Compute subgroup averages of the logits of the predicted conditional probability estimates defined as

$$\ell_{00} = \frac{\sum_{c=1}^{n_{00}} \text{logit}\left(p_{00c}\right)}{n_{00}}$$

$$\ell_{01} = \frac{\sum_{c=1}^{n_{01}} \text{logit}\left(p_{01c}\right)}{n_{01}}$$

$$\ell_{10} = \frac{\sum_{c=1}^{n_{10}} \text{logit}\left(p_{10c}\right)}{n_{10}}$$

$$\ell_{11} = \frac{\sum_{c=1}^{n_{11}} \text{logit}\left(p_{11c}\right)}{n_{11}},$$

   and averages of the predicted conditional probability estimates defined as

$$p_{00} = \frac{\sum_{c=1}^{n_{00}} p_{00c}}{n_{00}}$$

$$p_{01} = \frac{\sum_{c=1}^{n_{01}} p_{01c}}{n_{01}}$$

$$p_{10} = \frac{\sum_{c=1}^{n_{10}} p_{10c}}{n_{10}}$$

$$p_{11} = \frac{\sum_{c=1}^{n_{11}} p_{11c}}{n_{11}},$$

The two-machine RFPM method produces two counterfactual probabilities of success for each observation, p_{1i} and p_{0i}. These two probabilities can be used to calculate sample log odds ratio estimates by averaging over individual log odds ratio estimates. Since only a single RFPM is used in the intuitive interaction detection method, each observation only has one conditional probability estimate, and thus individual log odds ratio estimates cannot be computed. Due to there only being one conditional probability estimate per observation, ℓ_{ij} is calculated as the mean of the log odds of the conditional probability estimates for observations in subgroup G_{ij}.

Once subgroup-specific conditional probability estimates are calculated on the log odds scale, multiplicative interaction effects can be estimated using the log ratio of odds ratios (ℓROR) defined by

$$\ell ROR = \ell_{11} - \ell_{10} - \ell_{01} + \ell_{00}.$$

If there is no multiplicative interaction effect between x_1 and x_2, ℓROR will be close to zero. The following odds ratios can also be calculated,

$$OR_{10} = \frac{p_{10}(1 - p_{00})}{p_{00}(1 - p_{10})},$$

$$OR_{01} = \frac{p_{01}(1 - p_{00})}{p_{00}(1 - p_{01})},$$

and

$$OR_{11} = \frac{p_{11}(1 - p_{00})}{p_{00}(1 - p_{11})}.$$

Additive interaction effects can be estimated on the risk ratio scale using the $RERI$ (relative excess risk due to interaction) defined as

$$RERI_{RR} = \frac{p_{11}}{p_{00}} - \frac{p_{10}}{p_{00}} - \frac{p_{01}}{p_{00}} + 1.$$

If there is no additive interaction effect between x_1 and x_2, $RERI_{RR}$ will be close to zero. The information provided from these five estimates, in addition to area expertise, can aid in determining important interactions. This method can be used in the first steps of data analysis to provide quick insight into any interesting interactions prior to computing estimates for all possible pairs of predictors.

4.1.1 Simulations 3 and 4: Screening for possible interactions using the intuitive interaction detection method

Refer to the following logistic regression model as Model 2,

$$\text{logit}(p) = \beta_0 + \log(1.2)\, x_1 + \log(1.5)\, x_2 + \log(2)\, x_3 + \log(2)\, x_1 x_2$$

$$+ \log(5)\, x_2 x_3 + 0\, x_4 + ... + 0\, x_{10},$$

where x_1, x_2, and x_3 are uncorrelated binary predictors with main effects of $\beta_1 = \log(1.2) = 0.182$, $\beta_2 = \log(1.5) = 0.405$, and $\beta_3 = \log(2) = 0.693$. The remaining predictors are independently generated, binary and have no association with the outcome. Two interaction effects between corresponding predictor pairs (x_1, x_2) and (x_2, x_3) are added with effect sizes of $\beta_4 = \log(2) = 0.693$ and $\beta_5 = \log(5) = 1.609$. The intercept β_0 is chosen so $P(y = 1) \approx 0.3$ and the predictors are generated to have a randomly selected $P(x_i = 1)$ between 0.05 and 0.95.

The main goal of the intuitive interaction detection method is to quickly identify any potential interactions prior to estimation. For this reason, the estimates obtained using the detection method are not examined, but instead the method's ability to detect an interaction when an interaction exists is evaluated. Only multiplicative interactions are considered in these simulations, but the R code provided in Appendix A reports both multiplicative and additive interaction estimates. For the purpose of these simulations, an interaction estimate is deemed relevant if it is either less than $\log(1/1.05) = -0.04879$ or greater than $\log(1.05) = 0.04879$. When used in practice, these boundaries should be chosen using area expertise and knowledge of the subject matter.

The intuitive interaction detection method is assessed using two sets of 1000 simulations. In both sets, and under each choice of sample size n, a single RFPM is used with parameters set at $mtry = 3$, $ntree = 100$, and $nodesize = 5\%$. For Simulation 3, a single set of predictors is generated for each choice of n, and a new outcome vector is generated 1000 times. In Simulation 4, new predictors and a new outcome vector is generated each time. Considering all possible combinations of predictors ($\binom{10}{2} = 45$), counts of the number of times the intuitive interaction detection method reported a possible interaction (less than log(1/1.05) or greater than log(1.05)) are recorded in Table 4.1 (Simulation 3) and Table 4.2 (Simulation 4). The total number of false positives (where no interaction exists, but the estimate is deemed relevant) identified for each value of n is recorded at the end of the table.

In Simulation 3, the intuitive interaction method correctly identifies the true interacting pairs of predictors (x_1, x_2) and (x_2, x_3) in all runs for all choices of n. In Simulation 4, where new predictors and a new outcome vector is generated each time, true interacting pairs are identified almost 100% of the time. The number of correctly identified counts tends to increase with both increasing sample size and effect size. Both simulations report large amounts of false positives; however, fewer false positives occur with a larger sample size. There are also more false positives identified for pairs of predictors where one predictor is associated with the outcome (x_1, x_2, x_3).

Considering pairs of unassociated predictors (predictors after the dotted line in both tables), it is expected that the number of interactions identified is close to zero. Due to the particular choice of cutoff points, some false positives for each predictor pair are expected by chance, and this number should decrease with n. This behaviour is seen in Table 4.2, where the number of false positives identified for unassociated pairs of predictors stays relatively constant for each choice of n, and decreases with n. This behaviour is not evident in Table 4.1, where the number of false positives does not stay constant or decrease with n. For some predictor pairs, there are very small and very large false positives for each choice of n.

Consider unassociated pairs of predictors in Table 4.1 for $n = 50000$. For predictor pairs (x_4, x_6), (x_4, x_8), (x_4, x_{10}), (x_5, x_6) and (x_6, x_{10}), there are 0 false positives, and for predictor pair (x_7, x_{10}) there are 991 false positives. To further investigate these anomalies, consider predictor pair (x_7, x_{10}) for $n = 50000$. A histogram of the 1000 interaction estimates shows an approximate bell-shaped curve with a shifted mean of 0.074. Since the interaction estimate is calculated as a contrast of the four mean logit conditional probabilities, this shift in mean may be due to one of ℓ_{00}, ℓ_{01}, ℓ_{10} or ℓ_{00}

Table 4.1: Counts of the number of times the intuitive interaction detection method reported a possible interaction effect for predictor pair (x_j, x_k) for $j, k \in \{1, \ldots, 10\}$ and $j \neq k$ in Simulation 3. An interaction effect estimate is deemed relevant if it is either less than log(1/1.05) or greater than log(1.05). The true interacting pairs of predictors (x_1, x_2) and (x_2, x_3) are listed first, and the total number of false positives identified for each sample size is listed at the bottom.

x_j	x_k	counts		
		n		
		5000	10000	50000
1	2	1000	1000	1000
2	3	1000	1000	1000
1	3	664	996	998
1	4	973	214	48
1	5	409	305	147
1	6	925	776	13
1	7	268	213	6
1	8	842	342	2
1	9	242	390	6
1	10	994	197	16
2	4	483	289	242
2	5	785	199	346
2	6	489	855	267
2	7	434	594	580
2	8	464	191	417
2	9	491	984	235
2	10	604	371	354
3	4	524	451	94
3	5	668	391	319
3	6	533	952	95
3	7	476	635	391
3	8	395	328	317
3	9	423	1000	163
3	10	563	273	264
4	5	56	12	77
4	6	374	20	0
4	7	569	22	93
4	8	1000	72	0
4	9	115	19	1
4	10	953	24	0
5	6	996	66	0
5	7	344	117	4
5	8	46	376	18
5	9	820	950	60
5	10	57	219	1
6	7	77	843	7
6	8	104	982	74
6	9	171	17	9
6	10	927	878	0
7	8	88	37	276
7	9	424	520	391
7	10	996	69	991
8	9	833	21	0
8	10	347	41	2
9	10	75	93	1
Total FP		22021	16344	7325

Table 4.2: Counts of the number of times the intuitive interaction detection method reported a possible interaction effect for predictor pair (x_j, x_k) for $j, k \in \{1, \ldots, 10\}$ and $j \neq k$ in Simulation 4. An interaction effect estimate is deemed relevant if it is either less than log(1/1.05) or greater than log(1.05). The true interacting pairs of predictors (x_1, x_2) and (x_2, x_3) are listed first, and the total number of false positives identified for each sample size is listed at the bottom.

x_j	x_k	counts		
		n		
		5000	10000	50000
1	2	939	961	968
2	3	990	996	998
1	3	782	771	749
1	4	570	438	185
1	5	558	423	168
1	6	572	425	164
1	7	577	424	183
1	8	556	442	185
1	9	566	423	163
1	10	573	421	171
2	4	575	503	408
2	5	636	501	422
2	6	596	526	395
2	7	619	536	404
2	8	595	498	381
2	9	589	512	424
2	10	594	521	378
3	4	592	533	394
3	5	612	521	403
3	6	591	522	407
3	7	616	522	389
3	8	592	514	392
3	9	632	524	416
3	10	618	522	362
4	5	488	310	61
4	6	474	333	42
4	7	496	329	55
4	8	480	313	59
4	9	495	346	52
4	10	516	332	52
5	6	501	327	56
5	7	507	320	51
5	8	472	313	55
5	9	494	340	51
5	10	525	311	51
6	7	474	326	50
6	8	447	319	55
6	9	506	337	53
6	10	483	344	47
7	8	487	312	67
7	9	475	305	58
7	10	487	294	62
8	9	495	353	58
8	10	447	342	55
9	10	501	317	44
Total FP		23461	17845	8677

defined in step 4. For the 1000 simulations, define the following mean logit estimates,

$$\begin{aligned}\overline{\ell_{00}} &= \frac{\sum_{i=1}^{1000} \ell_{00i}}{1000}\\ \overline{\ell_{01}} &= \frac{\sum_{i=1}^{1000} \ell_{01i}}{1000}\\ \overline{\ell_{10}} &= \frac{\sum_{i=1}^{1000} \ell_{10i}}{1000}\\ \overline{\ell_{11}} &= \frac{\sum_{i=1}^{1000} \ell_{11i}}{1000},\end{aligned}$$

where ℓ_{00}, ℓ_{01}, ℓ_{10}, and ℓ_{11} are defined above. Since both x_7 and x_{10} are independently generated and have no association with the outcome, all of the above logits should be approximately equal to

$$\begin{aligned}\text{logit}\, P(y = 1|x_7, x_{10}) &= \text{logit}\, P(y = 1)\\ &= \text{logit}(0.3)\\ &= -0.8473.\end{aligned}$$

Using the results from Simulation 3, the above logits are found to be:

$$\overline{\ell_{00}} = -0.7741 \qquad \overline{\ell_{01}} = -0.8427$$

$$\overline{\ell_{10}} = -0.8431 \qquad \overline{\ell_{11}} = -0.8382.$$

The reduction in $\overline{\ell_{00}}$ explains the shift in the mean of the interaction estimates. A possible explanation for this reduction is that changes in x_7 and x_{10} are related to changes in the associated predictors x_1, x_2, and x_3. To investigate this, three different generalized linear models (GLMs) are used to regress x_1, x_2, and x_3 on x_7, x_{10}, and x_7x_{10}. These models will help determine if there exists some dependence in the predictors that is not revealed in the correlation matrix. The regression of x_1 reveals no significant terms, but for the x_2 and x_3 models, all three terms are significant (Tables 4.3 and 4.4). The results from these two GLMs reveal that although no simple correlation exists between predictors, both the probability $x_2 = 1$ and $x_3 = 1$ depend on x_7, x_{10} and their interaction. When $x_7 = 0$ and $x_{10} = 0$, both GLMs have a larger predicted probability of $x_2 = 1$ and $x_3 = 1$ than the values used to generate the data (0.284 and 0.670 for x_2 and x_3 respectively). This implies that the probability $x_2 = 1$ and $x_3 = 1$ is larger in the subgroup containing observations with $x_7 = 0$ and $x_{10} = 0$. When a new outcome vector is generated, $P(y = 1)$ is larger than the expected value of 0.3 in the G_{00} subgroup, thus increasing p_{00}. Therefore in Simulation 3, dependence is randomly incorporated when generating predictors, and since predictors remain the same, this dependence affects all outcome vectors. In Simulation 4, new predictors and a new outcome vector is generated each time, and thus any dependence randomly incorporated only affects one outcome vector rather than all 1000.

Table 4.3: Results of the generalized linear model regressing x_2 on x_7, x_{10}, and x_7x_{10}. Predictors x_2, x_7, and x_{10} are from the simulated data generated in Simulation 3.

	Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	-0.8277	0.0504	-16.43	0.0000
x_7	-0.1163	0.0555	-2.10	0.0361
x_{10}	-0.1125	0.0574	-1.96	0.0498
x_7x_{10}	0.1254	0.0631	1.99	0.0470

Table 4.4: Results of the generalized linear model regressing x_3 on x_7, x_{10}, and x_7x_{10}. Predictors x_3, x_7, and x_{10} are from the simulated data generated in Simulation 3.

	Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	0.8175	0.0503	16.26	0.0000
x_7	-0.1201	0.0550	-2.19	0.0288
x_{10}	-0.1543	0.0566	-2.73	0.0064
x_7x_{10}	0.1654	0.0619	2.67	0.0076

For predictor pairs with a large number of false positives, a possible explanation is a shift in mean due to a complex dependence on the associated predictors x_1, x_2, x_3. Other noticeable anomalies in Table 4.1 are the detection of 0 false positives for unassociated pairs of predictors. Consider predictor pair (x_4, x_6). A histogram of the 1000 interaction estimates shows an approximate bell-shaped curve with a mean of zero. The ℓROR estimates for this predictor pair, along with other unassociated pairs of predictors, have a standard deviation of approximately 0.01. Due to the small variance in the estimates and the particular choice of cutoff points, very little or no estimates are falling outside these bounds.

In Table 4.2, the method correctly identified the true interacting pairs of predictors in almost all simulations. For unassociated pairs of predictors, the number of false positives stays relatively constant and decreases with n. An investigation of the mean and standard deviation of the ℓROR estimates reveals unassociated pairs of predictors to have an approximate mean ℓROR value of 0 and a standard deviation of 0.025. Removing any possible correlations between predictors requires generating new predictors in each simulation, and so an increase in the variance of these estimates is expected. The variance for predictor pairs where one predictor is associated and the other is not is higher than for unassociated predictor pairs. This is expected as the effect of the associated predictor will affect the variance of the ℓROR estimates.

Overall, the intuitive interaction detection method is a reasonable approach to quickly screen for any interesting interactions prior to estimation. In Simulation 3, the true interacting pairs of predictors are identified in all simulations, and in almost all simulations in Simulation 4. This method performs better with a larger sample size as the total number of false positives reported decreases with n. Although there are many false positives identified, this method is relatively quick and is only a screening process that serves to provide insight into any potential interactions rather than provide interaction effect estimates.

4.2 Four-machine RFPM method

The effects of previously identified, or known interacting predictors, can be estimated using a procedure analogous to the two-machine RFPM method. Dasgupta et al. (2014) propose the four-machine RFPM method, which calculates counterfactual probabilities of success on the individual level. These estimated probabilities enable for the calculation of various multiplicative and additive interaction measures.

Consider a sample of size n with p binary predictors $(x_1, x_2, \ldots, x_p)$ and a binary outcome y. Suppose estimating the interaction effect due to predictors x_1 and x_2 is of primary interest. Use of the four-machine RFPM method to predict counterfactual probabilities of success is as follows:

1. Split the data into four subgroups based on the four possible combinations of

x_1 and x_2. Denote these subgroups G_{11}, G_{01}, G_{10}, and G_{00}, where G_{ij} is the subgroup containing observations with $x_1 = i$ and $x_2 = j$.

2. Within each subgroup, train identically specified RFPMs ($RFPM_{11}$, $RFPM_{01}$, $RFPM_{10}$, and $RFPM_{00}$) on the remaining $p-2$ predictors $(x_3, x_4, \ldots, x_p)$.

3. Obtain counterfactual probabilities of success for every observation by predicting from the four RFPMs. Each observation has a vector of four predicted probabilities of success $(p_{00i}, p_{01i}, p_{10i}, p_{00i})$. One predicted probability corresponds to an observed probability of success, and the remaining three correspond to counterfactual probabilities of success. For example, an observation with $x_1 = 1$ and $x_2 = 1$ has observed probability of success p_{11i} (prediction from $RFPM_{11}$) and counterfactual probabilities of success p_{10i}, p_{01i}, and p_{00i} (predictions from $RFPM_{01}$, $RFPM_{10}$, and $RFPM_{00}$).

The four predicted probabilities of success for each observation i, $(p_{00i}, p_{01i}, p_{10i}, p_{00i})$, enables for the calculation of several measures of interaction. For a measure of multiplicative interaction, the following three odds ratio estimates can be obtained for observation i,

$$OR_{10i} = \frac{p_{10i}/(1-p_{10i})}{p_{00i}/(1-p_{00i})},$$

$$OR_{01i} = \frac{p_{01i}/(1-p_{01i})}{p_{00i}/(1-p_{00i})},$$

and

$$OR_{11i} = \frac{p_{11i}/(1-p_{11i})}{p_{00i}/(1-p_{00i})}.$$

These odds ratios can be used to calculate the ratio of odds ratios (ROR). The ROR for observation i is given by,

$$ROR_i = \frac{OR_{11i}}{OR_{10i}OR_{01i}},$$

and is a measure of multiplicative interaction on the odds ratio scale. An additive interaction measure can also be estimated, namely the RERI on the risk ratio scale, for each observation i. The $RERI_{RR}$ for observation i is given by,

$$RERI_{RRi} = \frac{p_{11i}}{p_{00i}} - \frac{p_{10i}}{p_{00i}} - \frac{p_{01i}}{p_{00i}} + 1.$$

Interaction estimates for the full sample are calculated by averaging over the individual estimates,

$$ROR_{sample} = \frac{\sum_{i=1}^{n} ROR_i}{n},$$

and

$$RERI_{sample} = \frac{\sum_{i=1}^{n} RERI_{RRi}}{n}.$$

Now that use of the four-machine RFPM method to estimate multiplicative and additive interaction effects has been outlined, its use is demonstrated and compared to logistic regression in the following simulation.

4.2.1 Simulation 5: Comparing log odds ratio estimates from the four-machine RFPM method to logistic regression

In this simulation, optimal parameter settings determined in Simulation 1 ($ntree = 100$, $mtry = 3$, $nodesize = 5\%$) are used to compare the performance of the four-machine RFPM method to that of logistic regression. Consider Model 2 as mentioned in section 4.1.1,

$$\begin{aligned} \text{logit}(p) = \beta_0 + \log(1.2)\, x_1 + \log(1.5)\, x_2 + \log(2)\, x_3 + \log(2)\, x_1 x_2 \\ + \log(5)\, x_2 x_3 + 0\, x_4 + ... + 0\, x_{10}. \end{aligned}$$

1000 simulations are completed for each value of the sample size n = 5000, 10000, and 50000. A single set of predictors is generated under each choice of n, and a new outcome vector is generated 1000 times. Using the four-machine RFPM method, ℓROR estimates for $\beta_4 = \log(2) = 0.693$ and $\beta_5 = \log(5) = 1.609$ are obtained by averaging over the individual ℓROR_i estimates. Logistic regression log odds ratio estimates are obtained using a correctly specified model. For a given value of n, the sample mean, standard deviation ($s_{\hat{\beta}}$) and % relative bias ($\%bias$) for each β estimate using the four-machine RFPM method are reported in Table 4.5. The p-values listed are the results from a t-test comparing the mean of the RFPM estimates to the true parameter value.

Figure 4.1 shows the results of the simulation, where the box plots represent the interaction estimates produced from the four-machine RFPM method (green)

Table 4.5: Sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias ($\%bias$) of log odds ratio estimates β_4 and β_5 from Model 2 using the four-machine RFPM method. The p-values listed are the results from a t-test comparing the mean RFPM estimate to the true parameter value.

β	n	mean	$s_{\hat{\beta}}$	$\%bias$	p-value
$\beta_4 = \log(2) = 0.693$	5000	0.626	0.237	-9.75	< 0.001
	10000	0.669	0.130	-3.44	< 0.001
	50000	0.640	0.053	-7.63	< 0.001
$\beta_5 = \log(5) = 1.609$	5000	1.576	0.268	-2.10	< 0.001
	10000	1.571	0.167	-2.36	< 0.001
	50000	1.558	0.093	-3.20	< 0.001

and logistic regression (yellow). The figure is split in two plots based on the two interacting pairs of predictors ((x_1, x_2) and (x_2, x_3)) and the three sets of box plots per plot are the three different sample sizes in increasing order. The true parameter value is indicated in the two plots by the horizontal line. From both Table 4.5 and Figure 4.1, the bias in the estimates obtained from the four-machine RFPM method is slightly larger than those obtained using logistic regression. The four-machine RFPM method performs comparatively to the logistic regression in that the variability among the estimates is very similar.

Overall the four-machine RFPM method performs comparatively to a correctly specified logistic regression model. The four-machine RFPM method performs almost as efficiently as logistic regression, and does this without any specification of the model or the data generating process. It should be addressed that the four-machine RFPM method can only be used to estimate two-way interactions and the two-machine RFPM method cannot be used to obtain main effect estimates of interacting predictors. The next section entails the discussion of estimating more complex interaction effects and main effects simultaneously.

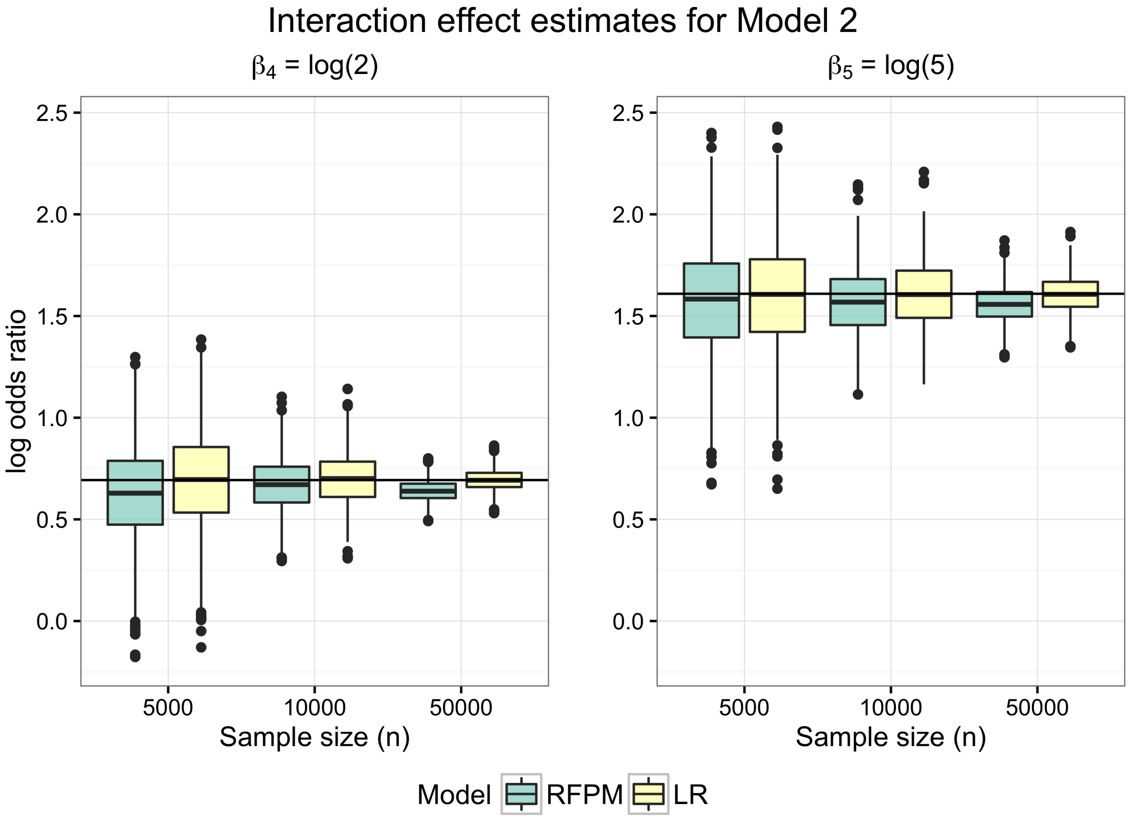


Figure 4.1: Log odds ratio estimates for β_4 and β_5 from Model 2 using the four-machine RFPM method (green) and logistic regression (yellow). Each plot represents one β estimate and the three sets of box plots are the estimates for the three different samples sizes, $n = 5000$, 10000, and 50000. The true parameter value is indicated in each plot by the horizontal line.

4.3 More complex interactions

The methodology behind both the two-machine and four-machine RFPM can easily be extended to estimate more complex interactions, such as 3-way and 4-way interactions. To estimate a k-way interaction effect, the data set is split into 2^k subgroups based on all the possible combinations of the predictors involved, and a RFPM is fitted to each of these subgroups. Using the 2^k RFPMs, observed and counterfactual probabilities of success for each individual can be used to calculate various risk estimates on the individual level and for the sample. Although this can be done, it may be burdensome and computationally intensive. Also, creating many subgroups may produce subgroups with varying sample sizes or subgroups with no observations. If this is the case, there may be a need to adjust the random forest parameter settings, such as the terminal node size to accommodate for these issues.

As previously demonstrated, the two-machine RFPM and the four-machine RFPM methods can be used to estimate interaction and main effect estimates respectively. Although main and interaction effects can be estimated, the two-machine RFPM method cannot be used to obtain main effect estimates for interacting predictors. To estimate both main and interaction effects, 2^m machines, where m is the highest order term in the fully saturated model of interacting predictors, must be used. For example, consider Model 2 used in Simulation 5. Estimating main and interaction effects requires $2^3 = 8$ machines since the highest order term in the fully saturated model of interacting predictors is the 3-way interaction term between x_1, x_2, and x_3. If only one interaction exists, say between x_1 and x_2, but x_1, x_2, and x_3 are all associated with the outcome, the two-machine RFPM method can be used to obtain a main effect estimate for x_3. The four-machine RFPM method can be used to obtain interaction and main effect estimates for x_1 and x_2, since the highest order term in the fully saturated model of interacting predictors x_1 and x_2 is the 2-way interaction term.

Chapter 5

Confidence intervals for RFPM risk estimates

In many cases, researchers may not only be interested in obtaining a point estimate for a given risk measure, but are also interested in bounding this estimate with a confidence interval. Constructing confidence intervals when the data is drawn from a known distribution is easily done and implemented in statistical software packages. However, it can be much more difficult when the distribution is unknown. Since RFPMs make no assumption about the data generating process and are completely non-parametric, standard approaches to constructing confidence intervals cannot be used. In order to obtain confidence intervals non-parametrically, the use of a bootstrapping method is explored.

In this chapter, the percentile bootstrap method is presented and its use in constructing confidence intervals for RFPM risk estimates is outlined. Using a simulation, confidence intervals constructed for RFPM estimates using percentile bootstrap samples are shown to have appropriate coverage probabilities, and to be an effective way to obtain confidence intervals. The computational aspects surrounding the construction are also discussed.

5.1 Bootstrap percentile method

The bootstrap method is a general approach in statistical inference used for empirical estimation, or approximation of sampling distributions. When the sampling distribution of an estimator cannot be defined mathematically, such as risk estimates derived from RFPMs, bootstrap methods are especially useful. The main idea of any bootstrap method is to use the observed data as an empirical estimate of the unknown distribution. The empirical distribution of the data is estimated from numerous bootstrap samples of the original data, where these bootstrap samples contain the same number of observations and are drawn with replacement. From these bootstrap samples, the statistic of interest is calculated, and the distribution of these values is used to approximate the population sampling distribution.

Consider a sample of size n with p binary predictors $(x_1, x_2, \ldots, x_p)$ and a binary outcome y. Suppose obtaining a point estimate and constructing a $(1-\alpha)\%$ confidence interval for risk measure θ corresponding to predictor x_1 is of interest. The bootstrap percentile method is as follows:

1. Use the two-machine RFPM method to construct θ estimates for each observation i using the counterfactual probabilities of success, $p_{0i}(y = 1|x_2, x_3, \ldots, x_p)$ and $p_{1i}(y = 1|x_2, x_3, \ldots, x_p)$. Calculate the point estimate $\hat{\theta}$ by averaging over the individual θ_i.

2. Take b random samples of size n, where b is drawn with replacement. For each sample, compute $\hat{\theta}$ using the two-machine RFPM method. Denote the collection of the b bootstrap estimates as $(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_b)$.

3. Order the b bootstrap estimates in ascending order, $(\hat{\theta}_{(1)}, \hat{\theta}_{(2)}, \ldots, \hat{\theta}_{(b)})$, where $\hat{\theta}_{(n)}$ is the n^{th} ordered bootstrap estimate.

The bounds of the bootstrap percentile confidence interval at the $(1-\alpha)\%$ confidence level are found by taking the $(\alpha/2)^{\text{th}}$ and the $(1-\alpha/2)^{\text{th}}$ percentiles of the bootstrap estimates. This bootstrapping approach does not assume normality, but may produce large coverage error if the distribution of the statistic is not approximately symmetric around the observed value (La Torre, 2010). However from previous simulation results, the distributions of the estimates derived from the two-machine and four-machine RFPM methods appear to be fairly symmetric.

The bootstrap percentile method is shown here to construct a $(1-\alpha)\%$ confidence interval for a main effect estimate using the two-machine RFPM method. The procedure can easily be extended to risk estimates derived from multiple machines, including the four-machine RFPM method. RFPMs compute counterfactual probabilities of success and so, the percentile bootstrap method can be used to construct confidence intervals for any of the risk measures mentioned in Chapter 2.

5.1.1 Simulations 6 and 7: Computing confidence intervals for log odds ratio estimates derived using the two-machine and four-machine RFPM methods

Refer to the following logistic model as Model 3,

$$\text{logit}(p) = \beta_0 + \log(2)\, x_1 + 0\, x_2 + ... + 0\, x_{10},$$

where x_1 is an independent, binary predictor with a main effect of $\beta_1 = \log(2) = 0.693$. The remaining predictors are independently generated, binary and have no association with the outcome. The intercept β_0 is chosen so that $P(y = 1) \approx 0.3$ and the predictors are generated to have a randomly selected $P(x_i = 1)$ between 0.05 and 0.95.

The bootstrap percentile method is used to obtain 95% confidence intervals for the log odds ratio of predictor x_1. Using $b = 200$, 500, and 1000 bootstrap samples, 10000 confidence intervals are constructed using a sample size of $n = 5000$. The endpoints of the bootstrap percentile confidence interval are produced by the 5th and 195th, and the 25th and the 975th values of the ordered bootstrap estimates for $b = 200$ and 1000 respectively. For $b = 500$, the endpoints are taken as an average of the 12th and 13th, and the 487th and 488th ordered values. An estimate of the coverage probability is calculated as the proportion of the 10000 confidence intervals for which the true parameter value ($\log(2) = 0.693$) is contained. 10000 simulation runs give a coverage probability estimate with a standard error of 0.0043 at the 5% significance level.

Estimated coverage probabilities and the average time in seconds to construct each confidence interval are reported in Table 5.1 under the Model 3 heading. The coverage probability for all three values of b is approximately 0.95, with $b = 500$ and 1000 having slightly higher probabilities. Although the estimated coverage probabilities differ only slightly and are all close to the desired 0.95, the computation times differ dramatically. The time to construct 10000 confidence intervals at a sample size of 5000 quickly increases with the number of bootstrap samples.

The bootstrap percentile method is shown to produce confidence intervals with appropriate coverage probabilities for a main effect estimated using the two-machine RFPM method. As an alternative, and more complex scenario, consider Model 2 as

Table 5.1: Estimated coverage probabilities of 95% confidence intervals for log odds ratio estimates of β_1 for Model 3, and of β_4 and β_5 for Model 2. 10000 confidence intervals produced for each β estimate using the bootstrap percentile method using $b = 200$, 500, and 1000 bootstrap samples of size $n = 5000$. Average time required in seconds to construct each confidence interval is reported under each model heading. Confidence intervals for Model 2 constructed using only b =200 bootstrap samples.

b	Model 3		Model 2		
	β_1	time (sec/CI)	β_4	β_5	time (sec/CI)
200	0.9479	2.9	0.9561	0.9444	4.5
500	0.9494	8.9	-	-	-
1000	0.9491	16.6	-	-	-

mentioned in section 4.1.1,

$$\text{logit}(p) = \beta_0 + \log(1.2)\, x_1 + \log(1.5)\, x_2 + \log(2)\, x_3 + \log(2)\, x_1x_2 \\ + \log(5)\, x_2x_3 + 0\, x_4 + ... + 0\, x_{10}.$$

The bootstrap percentile method is used to obtain confidence intervals for the log odds ratios, $\beta_4 = \log(2)$ and $\beta_5 = \log(5)$, corresponding to predictor pairs (x_1, x_2) and (x_2, x_3). Log odds ratio estimates for β_4 and β_5 are calculated using the four-machine RFPM method. Due to lengthy computation times, and the results from above, only $b = 200$ is considered. 10000 confidence intervals are constructed for each β estimate using a sample size of $n = 5000$. As estimate of the coverage probability is calculated as the proportion of the 10000 confidence intervals for which the true parameter values ($\log(2) = 0.693$ and $\log(5) = 1.609$ for β_4 and β_5 respectively) are contained.

The estimated coverage probability and the average time in seconds to construct each confidence interval are reported in Table 5.1 under the Model 2 heading. The estimated coverage probability of the confidence intervals produced for both β_4 and β_5 are approximately 0.95. Overall, the bootstrap percentile method is an efficient way to obtain confidence intervals for RFPM risk estimates even when the number of bootstrap samples taken is small. This method produces confidence intervals with appropriate coverage probabilities for estimates derived from the two-machine and the four-machine RFPM methods.

Chapter 6

Main effect estimation for categorical and continuous predictors using RFPMs

Thus far, obtaining risk estimates using RFPMs only for binary predictors has been considered. Although binary predictors are common in health research, researchers often encounter categorical (variable with two or more levels) and continuous (variable with infinite number of values) predictors. Since binary predictors are simply a special case of categorical predictors, the use of RFPMs to obtain risk estimates for a binary predictor can easily be extended to variables with more than two levels. The use of RFPMs to estimate risk measures for a continuous predictor is not as easily done, and requires the continuous predictor to be split into a number of bins and treated as a categorical predictor.

In this chapter, obtaining risk estimates for both categorical and continuous predictors are discussed. First, categorical predictors are considered and presented as an extension of binary predictors. Secondly, continuous predictors are considered, and the main ideas behind binning are presented using conclusions drawn from linear regression. Simulations demonstrating the use of RFPMs to obtain risk estimates follow the discussion of each predictor type.

6.1 Categorical predictors

Binary predictors are simply a special case of categorical predictors, and so the theory behind the two-machine RFPM method can be extended for variables with more than two levels. Consider a sample of size n with $p-1$ binary predictors $(x_2, x_3, \ldots, x_p)$ and a categorical predictor, x_1, with k levels, $(0, 1, 2, \ldots, k-1)$. Suppose obtaining risk estimates for each level of x_1 relative to the first is of interest. Predicting counterfactual probabilities of success using RFPMs is as follows:

1. Split the data set into k subgroups based on the different levels of x_1. Denote

these subgroups G_0, G_1, ..., G_{k-1} where G_j for $j \in \{0, 1, \ldots, k-1\}$ corresponds to the subgroup containing observations with $(x_1 = j, x_2, \ldots, x_p)$.

2. Within each subgroup, train identically specified RFPMs ($RFPM_0$, $RFPM_1$, ..., $RFPM_{(k-1)}$) on the remaining $p-1$ predictors $(x_2, x_3, \ldots, x_p)$.

3. Obtain observed and counterfactual probabilities of success for each observation by predicting from all k RFPMs. Each observation will have a vector of k counterfactual probabilities of success, $(p_{0i}, p_{1i}, \ldots, p_{(k-1)i})$.

For each observation i, risk estimates for each level of x_1 relative to the first can be obtained by comparing the appropriate probabilities of success. Various risk measure estimates for level j relative to baseline (level 0) for each observation i are calculated as follows;

Risk ratio

$$RR_{ji} = \frac{p_{ji}}{p_{0i}}$$

Odds ratio

$$OR_{ji} = \frac{p_{ji}/1 - p_{ji}}{p_{0i}/1 - p_{0i}}$$

Attributable risk

$$AR_{ji} = p_{ji} - p_{0i}.$$

Sample estimates for the effect of level j relative to baseline for each of the above risk measures is found by averaging over the individual estimates. Therefore, to obtain counterfactual probabilities of success for a categorical predictor with k levels, k RFPMs are needed. Effect estimates are obtained by comparing predicted probabilities for level $j \in \{0, 1, \ldots, k-1\}$, p_{ji}, to baseline predicted probabilities, p_{0i}. In the following simulation, the use of RFPMs to calculate log odds ratios for a categorical variable with four levels is demonstrated.

6.1.1 Simulation 8: Comparing log odds ratio estimates for a categorical predictor using RFPMs to logistic regression

Refer to the following logistic model as Model 4

$$\text{logit}(p) = \beta_0 + \beta_1\, x_1 + 0\, x_2 + \ldots 0\, x_{10},$$

where x_1 is an independent categorical predictor with levels (A, B, C, D) and the remaining predictors are independently generated, binary and have no association with the outcome. The intercept β_0 is chosen so that $P(y = 1) \approx 0.3$ and predictors $x_2, \ldots, x_{10}$ are generated to have a randomly selected $P(x_i = 1)$ between 0.05 and 0.95. Categorical predictor x_1 is generated with $P(x_1 = A) = 0.55$, $P(x_1 = B) = 0.2$, $P(x_1 = C) = 0.15$ and $P(x_1 = D) = 0.1$. The effects relative to baseline (level A) are generated to be $\log(1.5) = 0.405$, $\log(2) = 0.693$ and $\log(4) = 1.386$ for levels B, C and D respectively.

In order to demonstrate the use of RFPMs to obtain risk estimates for each level of x_1 relative to baseline, rewrite Model 4 as

$$\text{logit}(p) = \beta_0 + \log(1.5)\, x_B + \log(2)\, x_C + \log(4)\, x_D + 0\, x_2 + \cdots + 0\, x_{10},$$

where x_B, x_C and x_D are indicator variables for categories B, C, and D taking values of 0 or 1. In Simulation 8, optimal parameter settings determined in Simulation 1 ($ntree$ = 100, $mtry$ = 3, $nodesize$ = 5%) are used to compare the performance of RFPMs to that of logistic regression in estimating log odds ratios for categorical predictor x_1. 1000 simulations are completed for each value of n = 5000, 10000, and 50000, where a single set of predictors is generated for each sample size and a new outcome vector is generated 1000 times. Log odds ratio estimates for β_1, β_2, and β_3 are obtained by averaging over the subject-specific log odds ratio estimates from four RFPMs. Logistic regression log odds ratio estimates are obtained using a correctly specified model. For each value of n, the sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias (%*bias*) for each β estimate using four RFPMs are reported in Table 6.1. The p-values listed are the results from a t-test comparing the mean of the RFPM estimates to the true parameter value.

Figure 6.1 shows the results of the simulation where the box plots represent the log odds ratio estimates produced from the four RFPMs (green) and logistic regression

Table 6.1: Sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias ($\%bias$) of log odds ratio estimates for β_1, β_2, and β_3, from Model 4 using RFPMs. Four RFPMs are used to obtain log odds ratio estimates for each level of the categorical predictor (B, C, D) relative to first level (A). The p-values reported are the results from a t-test comparing the mean RFPM estimate to the true parameter value.

β	n	mean	$s_{\hat{\beta}}$	$\%bias$	p-value
$\beta_1 = \log(1.5) = 0.405$	5000	0.410	0.085	1.12	0.091
	10000	0.403	0.058	-0.71	0.116
	50000	0.406	0.025	0.24	0.212
$\beta_2 = \log(2) = 0.693$	5000	0.699	0.088	0.87	0.031
	10000	0.694	0.061	0.08	0.776
	50000	0.693	0.028	-0.02	0.856
$\beta_3 = \log(4) = 1.386$	5000	1.389	0.103	0.18	0.435
	10000	1.381	0.070	-0.38	0.019
	50000	1.386	0.032	0.01	0.915

(yellow). The figure is split in three plots based on the main effects for the three indicator variables x_B, x_C, and x_D, and the three sets of box plots per plot represent the three different sample sizes in increasing order. The true parameter value is indicated in each plot by the horizontal line. In almost all applications of the RFPM method, approximately unbiased estimates are produced. Using RFPMs to obtain log odds ratio estimates for a categorical predictor performs comparatively to a correctly specified logistic regression model in that the variability among the estimates is very similar. RFPMs provide an efficient non-parametric approach to obtaining not only log odds ratio estimates, but other risk effect measures derived from the counterfactual probabilities.

6.2 Continuous predictors

Predicting risk effects for categorical predictors with more than two levels using RPFMs can be thought of as an extension of the two-machine RFPM method. By converting the categorical predictor with k levels into $k-1$ binary indicator variables, estimates can be obtained using a similar two-machine RFPM method procedure.

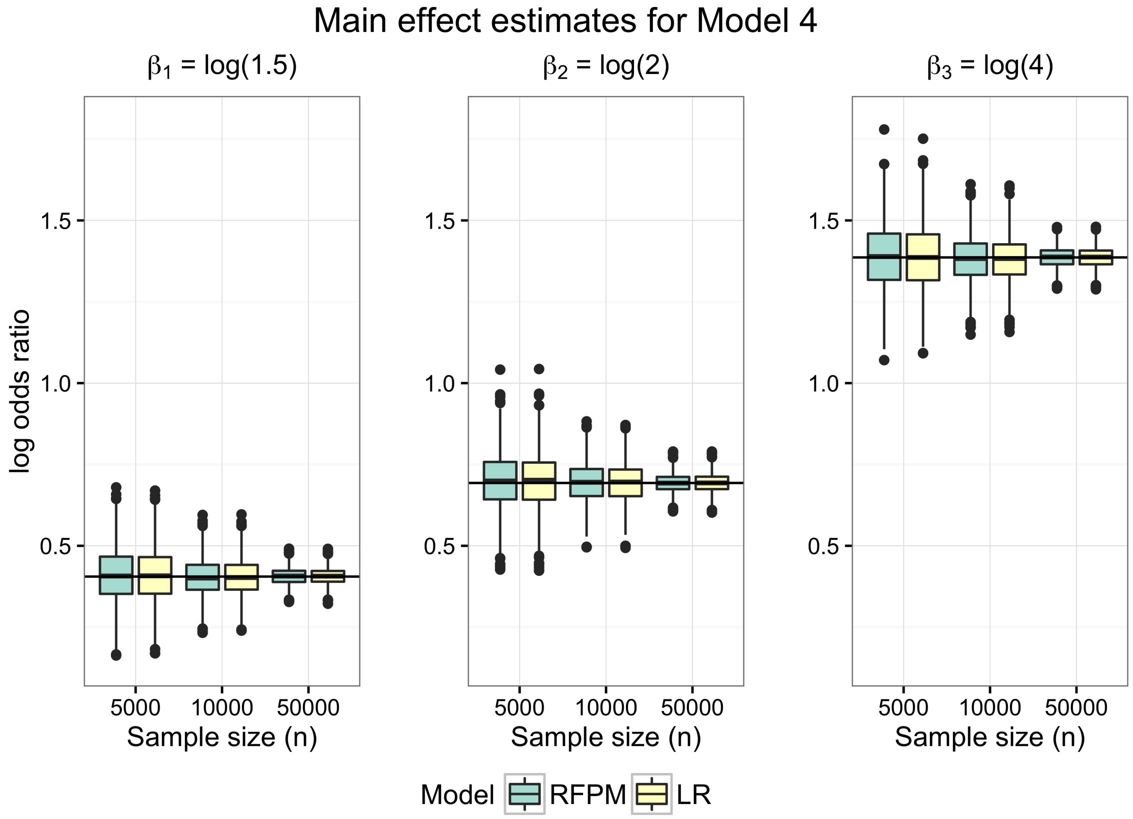


Figure 6.1: Log odds ratio estimates for β_1, β_2, and β_3 from Model 4 using RFPMs (green) and logistic regression (yellow). Four RFPMs are used to obtain log odds ratio estimates for each level of the categorical predictor (B, C, D) relative to the first level (A). Each plot represents one β estimate and the three sets of box plots are the estimates for the three different samples sizes, $n = 5000$, 10000, and 50000. The true parameter value is indicated in each plot by the horizontal line.

Continuous predictors present a different challenge in that it is not feasible to convert to indicator variables based on the perhaps infinite "levels" of the continuous predictor. One approach to using RFPMs in predicting risk effects for a continuous predictor is to use the concept of binning.

The goal is to split the continuous predictor into a number of distinct bins based on some splitting criterion, and use RFPMs to obtain estimates for each bin. These bin estimates can be plotted against the mean bin values of the continuous predictor to examine graphically the relationship between the continuous predictor and the outcome. For a linear relationship, as seen in an additive model for example, the plot would display a linear relationship, and various techniques such as a linear model can be used to estimate the slope (effect).

In order to understand what happens when a continuous variable is discretized, consider first the example of simple linear regression. Let the true relationship between continuous predictor x and response y be

$$y_i = \beta_0 + \beta_1\, x_i + \epsilon_i, \tag{6.1}$$

where the deviations ϵ_i satisfy the usual assumptions for ordinary least squares. These are assumed to be independent and approximately normal with zero mean and constant variance. Consider a binary split, where x is split into two bins. Define $\tilde{x}$ to be 0 for x less than its median m, and 1 for x greater than m. Assume the number of observations N is even so the median is not equal to any of the values, and splitting into two bins creates equal sample sizes, namely $n_0 = n_1 = N/2$.

The mean response for observations in bin 1 ($\tilde{x} = 0$) is

$$\begin{aligned}
\bar{y}_{\tilde{x}=0} &= \frac{1}{n_0} \sum_{i=1}^{N} y_i\, I(x_i < m) \\
&= \frac{1}{n_0} \sum_{i=1}^{N} (\beta_0 + \beta_1\, x_i + \epsilon_i)\, I(x_i < m) \\
&= \beta_0 + \beta_1 \frac{\sum_{i=1}^{N} x_i\, I(x_i < m)}{n_0} + \frac{\sum_{i=1}^{N} \epsilon_i\, I(x_i < m)}{n_0} \\
&= \beta_0 + \beta_1\, \bar{x}_{\tilde{x}=0} + \bar{\epsilon}_{\tilde{x}=0},
\end{aligned}$$

where $I(\cdot)$ represents the indicator function, $\bar{x}_{\tilde{x}=0}$ is the mean of x's in bin 1 ($\tilde{x} = 0$), and $\bar{\epsilon}_{\tilde{x}=0}$ is the mean of the deviations for x's in bin 1. Similarly, the mean response for observations in bin 2 ($\tilde{x} = 1$) can be expressed as

$$\bar{y}_{\tilde{x}=1} = \beta_0 + \beta_1\, \bar{x}_{\tilde{x}=1} + \bar{\epsilon}_{\tilde{x}=1}.$$

The least squares fit to the model with the new discretized variable $\tilde{x}$ is

$$\hat{y} = \hat{\tilde{\beta}}_0 + \hat{\tilde{\beta}}_2\, \tilde{x}$$

with slope estimate

$$\hat{\tilde{\beta}}_2 = \frac{\Delta y}{\Delta \tilde{x}} = \frac{\bar{y}_{\tilde{x}=1} - \bar{y}_{\tilde{x}=0}}{1 - 0} = \bar{y}_{\tilde{x}=1} - \bar{y}_{\tilde{x}=0}.$$

Substituting expressions for $\bar{y}_{\tilde{x}=1}$ and $\bar{y}_{\tilde{x}=0}$,

$$\begin{aligned}\hat{\tilde{\beta}}_2 &= \bar{y}_{\tilde{x}=1} - \bar{y}_{\tilde{x}=0} \\ &= \beta_0 + \beta_1\,\bar{x}_{\tilde{x}=1} + \bar{\epsilon}_{\tilde{x}=1} - (\beta_0 + \beta_1\,\bar{x}_{\tilde{x}=0} + \bar{\epsilon}_{\tilde{x}=0}) \\ &= \beta_1\,\bar{x}_{\tilde{x}=1} - \beta_1\,\bar{x}_{\tilde{x}=0} + \bar{\epsilon}_{\tilde{x}=1} - \bar{\epsilon}_{\tilde{x}=0} \\ &= \beta_1\,(\bar{x}_{\tilde{x}=1} - \bar{x}_{\tilde{x}=0}) + \bar{\epsilon}_{\tilde{x}=1} - \bar{\epsilon}_{\tilde{x}=0}. \qquad (6.2)\end{aligned}$$

By (6.2), the estimated slope $\hat{\tilde{\beta}}_2$ is an unbiased estimator of the true slope, β_1, multiplied by the difference in mean x values for each bin.

The theory behind two bins can be extended to discretizing continuous predictor x into k bins of equal sizes using $k-1$ splits. Define the following indicator variables $(\tilde{x}_2, \tilde{x}_3, \ldots, \tilde{x}_k)$, where $\tilde{x}_j$ for $j \in \{2, \ldots, k\}$ is 1 for any x belonging to bin j and is 0 otherwise. Using these indicator variables, the following regression model is fitted

$$\hat{y} = \hat{\tilde{\beta}}_0 + \hat{\tilde{\beta}}_2\, x_2 + \cdots + \hat{\tilde{\beta}}_k\, x_k,$$

where $\tilde{\beta}_j$ for $j \in \{2, \ldots, k\}$ is a measure of the difference in mean response for x belonging to the j^{th} bin relative to the mean response for x in the first bin. Since the primary goal is obtaining an estimate for β_1 in (6.1), result (6.2) from the binary case is extended to show the least squares estimate of $\tilde{\beta}_j$ to be

$$\hat{\tilde{\beta}}_j = \beta_1(\bar{x}_j - \bar{x}_1) + \bar{\epsilon}_j - \bar{\epsilon}_1, \qquad (6.3)$$

where $\bar{x}_j$ is the mean x values for x belonging to bin j, and $\bar{\epsilon}_j$ is the mean of the deviations for x in bin j. The equation above is expanded as

$$\hat{\tilde{\beta}}_j = -\beta_1\,\bar{x}_1 + \beta_1\,\bar{x}_j + \bar{\epsilon}_j - \bar{\epsilon}_1,$$

and since $\beta_1\,\bar{x}_1$ is constant, this equation has the form of a simple linear regression,

$$y = \theta_0 + \theta_1\,x + u,$$

where $y = \hat{\tilde{\beta}}_j$, $\theta_0 = -\beta_1\,\bar{x}_1$, $\theta_1 = \beta_1$ and $u = \bar{\epsilon}_j - \bar{\epsilon}_1$. The random deviations, u, have zero mean, constant variance and no correlation. Thus, an unbiased estimator for the slope β_1 is obtained by regressing the bin estimates $\hat{\tilde{\beta}}_j$ for $j \in \{2, \ldots, k\}$ on the mean bin values $\bar{x}_j$ for $j \in \{2, \ldots, k\}$. Separate estimates for β_1 from each bin are given by $\hat{\hat{\beta}}_j/(\bar{x}_j - \bar{x}_1)$ for $j \in \{2, \ldots, k\}$.

Obtaining an unbiased estimator for the slope β_1 for a discretized continuous predictor using logistic regression is not as straightforward as the cases presented above. Consider a binary outcome y with $P(y = 1) = p$, where y is dependent on continuous predictor x through the linear predictor η,

$$\eta = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1\, x_1 + \epsilon.$$

The case where x is split into two bins is considered first. Split x in two equal sized bins and define $\tilde{x}$ as before, where $\tilde{x} = 0$ if x is less than the median (m) and $\tilde{x} = 1$ for $x > m$. Since the response is binary, there are a number of 0 and 1 responses at both $\tilde{x} = 0$ and $\tilde{x} = 1$. Let $\hat{p}_j$ for $j \in \{0, 1\}$ be the proportion of 1s at each $\tilde{x}$. Estimates for coefficients in the fitted model with the new discretized variable $\tilde{x}$,

$$\hat{\hat{\eta}} = \hat{\hat{\beta}}_0 + \hat{\hat{\beta}}_2\, \tilde{x}$$

are obtained using maximum likelihood estimation. The maximum likelihood estimator of the slope is

$$\hat{\hat{\beta}}_2 = \log\left(\frac{\hat{p}_1}{1-\hat{p}_1}\right) - \log\left(\frac{\hat{p}_0}{1-\hat{p}_0}\right). \tag{6.4}$$

Relating this estimator to the true slope, β_1, is more complicated than in the linear regression case because the true slope depends on the responses y_i in a complex way. In order to obtain a result analogous to that for linear regression, two approximations must be made. The first approximation is made by replacing the expectation in (6.4) with the expectations of the observed proportions,

$$\begin{aligned} E\left[\hat{\hat{\beta}}_2\right] &= E\left[\log\left(\frac{\hat{p}_1}{1-\hat{p}_1}\right) - \log\left(\frac{\hat{p}_0}{1-\hat{p}_0}\right)\right] \\ &\approx \log\left(\frac{E[\hat{p}_1]}{1-E[\hat{p}_1]}\right) - \log\left(\frac{E[\hat{p}_0]}{1-E[\hat{p}_0]}\right). \end{aligned} \tag{6.5}$$

Since there are $N/2$ observations in bin 1 corresponding to $\tilde{x} = 0$ and $N/2$ observations in bin 2 corresponding to $\tilde{x} = 1$, the observed proportions at $\tilde{x} = 0$ and $\tilde{x} = 1$ can be written as

$$\hat{p}_0 = \frac{\sum_{i=1}^{N} y_i\, I(x_i < m)}{N/2}$$

and

$$\hat{p}_1 = \frac{\sum_{i=1}^{N} y_i \, I(x_i > m)}{N/2},$$

where $I(\cdot)$ is the indicator function. The expectations of $\hat{p}_0$ and $\hat{p}_1$ are

$$\begin{aligned} E[\hat{p}_0] &= \frac{2}{N}\sum_{i=1}^{N} I(x_i < m) p_i \\ &= \frac{2}{N}\sum_{i=1}^{N} I(x_i < m)\frac{exp(\eta_i)}{1 + exp(\eta_i)} \end{aligned}$$

and

$$E[\hat{p}_1] = \frac{2}{N}\sum_{i=1}^{N} I(x_i > m)\frac{exp(\eta_i)}{1 + exp(\eta_i)},$$

where $\eta_i = \beta_0 + \beta_1 \, x_i$ is the true linear predictor for observation i.
The second approximation is to replace the average of the probabilities in these expectation expressions by the probability at the average value of x_i, giving

$$E[\hat{p}_0] \approx \frac{exp(\bar{\eta}_0)}{1 + exp(\bar{\eta}_0)}$$

and

$$E[\hat{p}_1] \approx \frac{exp(\bar{\eta}_1)}{1 + exp(\bar{\eta}_1)},$$

where $\bar{\eta}_0 = \beta_0 + \beta_1 \, \bar{x}_0$ and $\bar{\eta}_1 = \beta_0 + \beta_1 \, \bar{x}_1$. In these expressions, $\bar{x}_0$ and $\bar{x}_1$ are the average x in bin 1 and bin 2 respectively. Using this second approximation in (6.5), the expected value of the slope estimator is

$$\begin{aligned} E[\hat{\tilde{\beta}}_2] &\approx \bar{\eta}_1 - \bar{\eta}_0 \\ &= \beta_1(\bar{x}_1 - \bar{x}_0), \end{aligned}$$

which is analogous to result (6.2) in the binary case for linear regression.

This result can be extended to the case of splitting a continuous predictor x into k equal sized bins. Similarly to the linear regression case, define the indicator variables $(\tilde{x}_2, \tilde{x}_3, \ldots, \tilde{x}_k)$, where $\tilde{x}_j$ for $j \in \{2, \ldots, k\}$ is 1 for any x belonging to bin j and is

0 otherwise. The logistic regression model with the new discretized variable has the linear predictor

$$\tilde{\eta} = \tilde{\beta}_0 + \tilde{\beta}_2\,\tilde{x}_2 + \cdots + \tilde{\beta}_k\,\tilde{x}_k.$$

Using the same two approximations as used in the binary case leads to

$$E\left[\hat{\tilde{\beta}}_j\right] \approx \beta_1\,(\bar{x}_j - \bar{x}_1), \tag{6.6}$$

which is similar to result (6.3) obtained in linear regression. An approximately unbiased estimate of β_1 is obtained by regressing the $\hat{\tilde{\beta}}_j$ estimates on the difference in bin means $(\bar{x}_j - \bar{x}_1)$ using a zero intercept. Individual β_1 estimates from each bin are given by $\hat{\tilde{\beta}}_j/(\bar{x}_j - \bar{x}_1)$ for $j \in \{2, \ldots, k\}$.

The analyses given above show how the actual linear term can be recovered when a continuous variable is discretized, exactly for linear regression and approximately for logistic regression. A similar approach is proposed for continuous predictors with RFPMs.

In order to obtain risk estimates for each bin j relative to the first bin, $\hat{\tilde{\beta}}_j$ for $j \in \{2, \ldots, k\}$, RFPMs can be used in a similar fashion as in the case for categorical predictors. Consider a sample of size n with $p-1$ binary predictors $(x_2, x_3, \ldots, x_p)$ and a continuous predictor, x_1. Suppose x_1 is split into k equal sized bins and obtaining risk estimates for each bin relative to the first is of interest. Predicting counterfactual probabilities of success using RFPMs is as follows:

1. Determine cutpoints that will split x_1 into k distinct bins (using quantiles for example). Using these cutpoints, convert x_1 into a factor variable, $\tilde{x}$, with levels $(1, 2, \ldots, k)$.

2. Split the data set into k groups based on the different levels of $\tilde{x}$. Denote these subgroups G_1, G_2, $\ldots$, G_k, where G_j for $j \in \{1, 2, \ldots, k\}$ corresponds to the subgroup containing observations with $(\tilde{x} = j, x_2, \ldots, x_p)$.

3. Within each subgroup, train identically specified RFPMs ($RFPM_1$, $RFPM_2$, $\ldots$, $RFPM_k$) on the remaining $p-1$ predictors $(x_2, x_3, \ldots, x_p)$.

4. Obtain observed and counterfactual probabilities of success for each observation by predicting from all k RFPMs. Each observation i has a vector of k counterfactual probabilities of success, $(p_{0i}, p_{1i}, \ldots, p_{ki})$.

Similarly to categorical predictors, risk estimates for each bin relative to the first can be obtained by comparing the appropriate predicted probabilities of success.

The theory given above is only valid when the effect of x_1 is linear. Since the relationship between x_1 and the outcome can take many forms, the bin estimates $\hat{\hat{\beta}}_j$ can be plotted against mean x bin values $\bar{x}_j$ to examine this relationship graphically. Rather than using methods for estimating a linear relationship, methods such as spline regression may be helpful in estimating the relationship. In the following simulation, the use of RFPMs to obtain bin estimates for a $N(0,1)$ continuous predictor is demonstrated. Two types of relationships, namely a linear and quadratic, are explored.

6.2.1 Simulation 9: Computing individual bin and overall log odds ratio estimates using RFPMs for a continuous predictor

Refer to the following logistic model as Model 5,

$$\text{logit}(p) = \beta_0 + \beta_1\, x_1 + 0\, x_2 + \ldots 0\, x_{10},$$

where x_1 is an independent and normally distributed random variable with a mean of 0 and a variance of 1. The remaining predictors $(x_2, \ldots, x_{10})$ are independently generated, binary and have no association with the outcome. The intercept β_0 is chosen so that $P(y = 1) \approx 0.3$ and the binary predictors are generated to have a randomly selected $P(x_i = 1)$ between 0.05 and 0.95.

The relationship of x_1 with the outcome is linear and so by the results given in section 6.2, an approximate estimate for β_1 is obtained by regressing the bin estimates on the difference in mean bin x_1 values using a zero intercept. For Simulation 9, 200 simulations are completed using three different effect sizes for β_1, $\log(1.2) = 0.182$, $\log(2) = 0.693$, and $\log(5) = 1.609$, and using sample sizes of $n = 5000$, 10000 and 50000. Predictors for each combination of n and β_1 are generated once, and a new outcome vector is generated for each simulation.

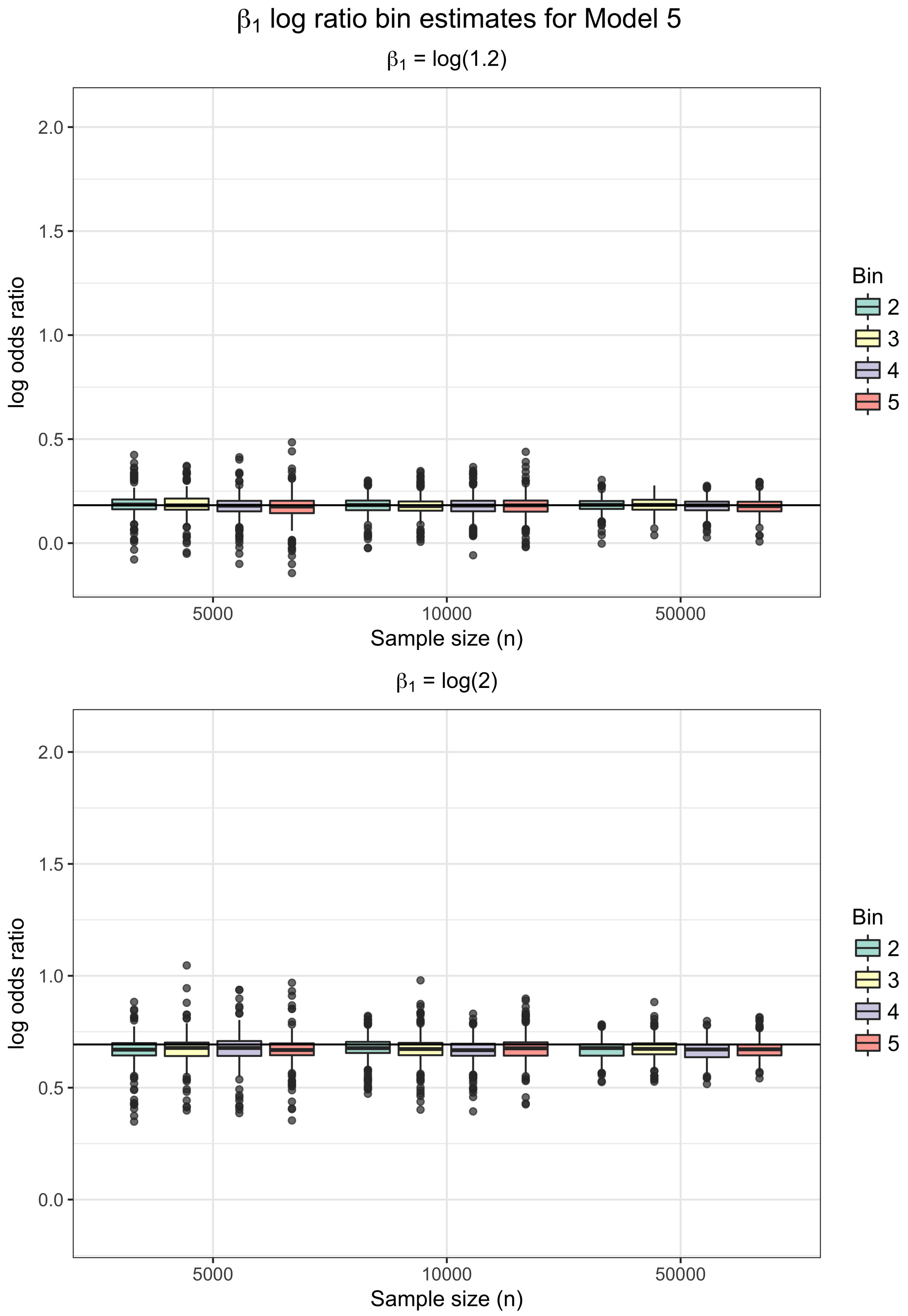
The predictor x_1 is split into 5 distinct bins using the *quantcut* command in the *gtools* R package. This function converts x into a factor variable using the intervals specified by sample quantiles corresponding to the given probabilities (0.2 for 5 bins). Counterfactual probabilities of success are calculated by five RFPMs ($mtry = 3$, *ntree* = 100, *nodesize* = 5%), and from these, four log odds ratio estimates $\hat{\hat{\beta}}_j$ for $j \in \{2, 3, 4, 5\}$ are obtained for each bin by comparing to the first bin. For each $\hat{\hat{\beta}}_j$ for $j \in \{2, 3, 4, 5\}$, an estimate for β_1 is obtained by dividing $\hat{\hat{\beta}}_j$ by $(\bar{x}_j - \bar{x}_1)$, where $\bar{x}_j$ is the mean of x_1 in bin j. Sample mean, standard deviation ($s_{\hat{\beta}_1}$) and % relative bias (*% bias*) for each β_1 estimate from the four bins per sample size are reported in Table 6.2. The p-values listed are the results from a t-test comparing the mean of the RFPM estimates to the true parameter value.

Table 6.2: Sample mean, standard deviation ($s_{\hat{\beta}_1}$), and % relative bias (%*bias*) for each log odds ratio estimate, β_1, from the four bins produced by discretizing continuous $N(0, 1)$ predictor x_1. Estimate of β_1 for each bin calculated as $\hat{\hat{\beta}}_j / (\bar{x}_j - \bar{x}_1)$. The p-values listed are the results from a t-test comparing the mean RFPM estimate to the true parameter value.

β_1	n	bin	mean	$s_{\hat{\beta}_1}$	%*bias*	p-value
$\beta_1 = \log(1.2) = 0.182$	5000	2	0.174	0.122	-4.47	0.344
		3	0.177	0.072	-2.79	0.317
		4	0.174	0.056	-4.33	0.049
		5	0.184	0.036	0.76	0.587
	10000	2	0.180	0.084	-1.31	0.689
		3	0.183	0.048	0.45	0.806
		4	0.180	0.037	-1.36	0.350
		5	0.180	0.026	-1.03	0.316
	50000	2	0.180	0.038	-1.10	0.454
		3	0.183	0.022	0.33	0.698
		4	0.181	0.016	-0.69	0.261
		5	0.182	0.011	-0.00	0.977
$\beta_1 = \log(2) = 0.693$	5000	2	0.655	0.134	-5.48	< 0.001
		3	0.670	0.083	-3.36	< 0.001
		4	0.679	0.059	-2.08	0.001
		5	0.680	0.037	-1.96	< 0.001
	10000	2	0.658	0.096	-5.07	< 0.001
		3	0.669	0.060	-3.41	< 0.001
		4	0.676	0.047	-2.47	< 0.001
		5	0.679	0.029	-2.04	< 0.001

β_1	n	bin	mean	$s_{\hat{\beta}_1}$	$\%bias$	p-value
$\beta_1 = \log(2) = 0.693$ *cont.*	50000	2	0.655	0.044	-5.53	< 0.001
		3	0.672	0.027	-3.10	< 0.001
		4	0.676	0.018	-2.44	< 0.001
		5	0.679	0.011	-2.07	< 0.001
$\beta_1 = \log(5) = 1.609$	5000	2	1.431	0.198	-11.07	< 0.001
		3	1.499	0.120	-6.89	< 0.001
		4	1.523	0.087	-5.40	< 0.001
		5	1.495	0.063	-7.09	< 0.001
	10000	2	1.430	0.136	-11.14	< 0.001
		3	1.487	0.082	-7.58	< 0.001
		4	1.517	0.058	-5.76	< 0.001
		5	1.495	0.043	-7.10	< 0.001
	50000	2	1.409	0.055	-12.45	< 0.001
		3	1.477	0.034	-8.26	< 0.001
		4	1.510	0.023	-6.19	< 0.001
		5	1.489	0.017	-7.51	< 0.001

Figure 6.2 shows the results of the simulation where the box plots represent the log odds ratio β_1 estimates for each of the bins. The figure is split in three plots based on the three effect sizes, log(1.2), log(2), and log(5), and the three sets of box plots per plot represent the three different sample sizes in increasing order. The plots are printed on the same scale for comparison purposes. The standard deviations reported in Table 6.2 decrease for increasing bins, and this is due to the increasing difference between $\bar{x}_j$ and $\bar{x}_1$ for increasing j. The % relative bias tends to increase with effect size and does not decrease with sample size. This indicates that although the standard deviation decreases at a bin level as the sample size increases, there isn't a large reduction in bias by increasing the sample size. The bias becomes statistically significant for a larger effect size. Consider the individual bin estimates for $\beta_1 = \log(5)$. The majority of the bin estimates underestimate the true β_1 value (indicated by the horizontal line), and the bias remains relatively constant for all sample sizes.



β_1 log ratio bin estimates for Model 5

β_1 = log(1.2)

β_1 = log(2)

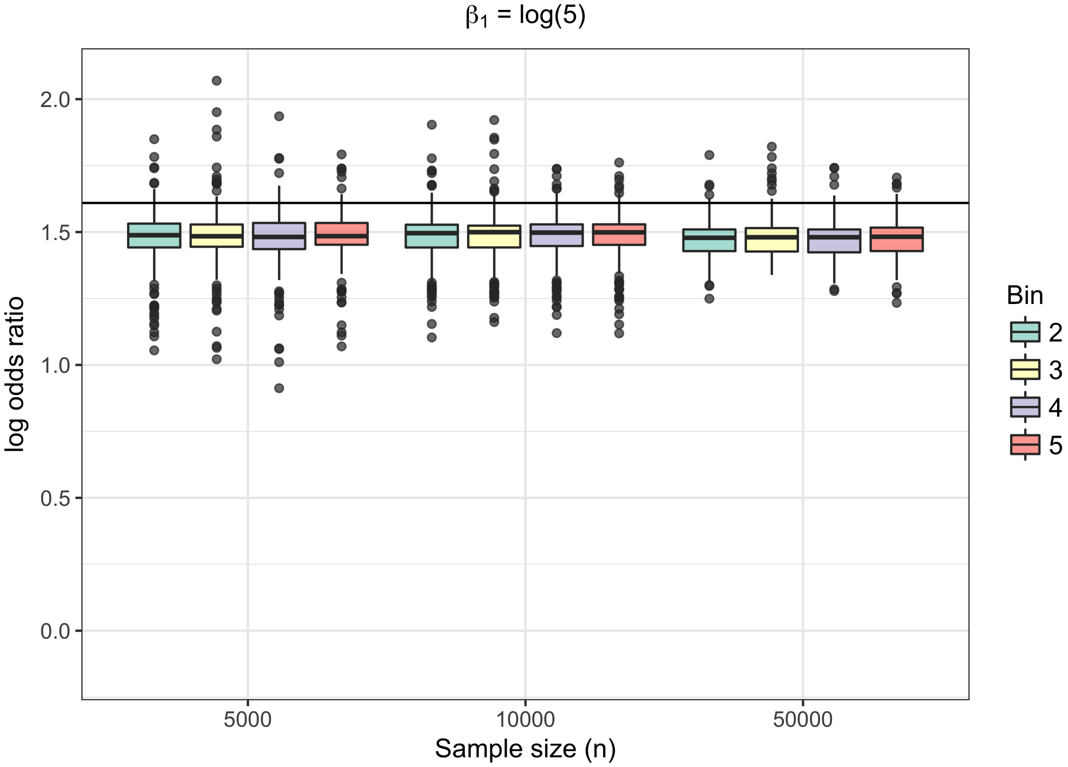


Figure 6.2: Individual bin log odds ratio estimates of β_1 in Model 5 with β_1 effect sizes of log(1.2), log(2), and log(5). Each plot represents estimates for one β_1 value, and the three sets of box plots are the individual β_1 bin estimates calculated by $\hat{\hat{\beta}}_j/(\bar{x}_j - \bar{x}_1)$ for $n = 5000$, 10000, and 50000. The true parameter value is indicated in each plot by the horizontal line.

Since it is known that the relationship of x_1 with the outcome is linear, the $\hat{\hat{\beta}}_j$ for $j \in \{2, 3, 4, 5\}$ estimates are regressed on the difference in mean x_1 bin values using a linear model with zero intercept in order to obtain an overall estimate for β_1. The sample mean, standard deviation ($s_{\hat{\beta}_1}$) and % relative bias ($\%bias$) of the regression β_1 slope estimates for each of the effect sizes are reported in Table 6.3. The p-values reported are the results from a t-test comparing the mean of the β_1 regression estimates to the true parameter value. The box plots in Figure 6.3 represent the regression β_1 slope estimates (log odds ratio) for the different effect sizes. The three box plots per plot are the estimates for the varying samples sizes in increasing order. For both the smaller effect size (log(1.2)), the bias is not statistically significant, whereas significant bias exists for all samples sizes with an effect size of $\beta_1 = \log(5)$. For increasing sample size, the standard deviation of the estimates decreases, but as seen before, the bias tends to remain constant.

Table 6.3: Sample mean, standard deviation ($s_{\hat{\beta}}$), and % relative bias ($\%bias$) of log odds ratio estimates for varying β_1 values in Model 5. Estimates for β_1 are obtained by regressing individual log odds ratio bin estimates $\hat{\hat{\beta}}_j$ on the difference in bin means $(\bar{x}_j - \bar{x}_1)$. The p-values listed are the results from a t-test comparing the mean of the β_1 regression estimates to the true parameter value.

β	n	mean	$s_{\hat{\beta}}$	$\%bias$	p-value
$\beta_1 = \log(1.2) = 0.182$	5000	0.180	0.042	-1.32	0.417
	10000	0.181	0.029	-0.93	0.414
	50000	0.182	0.012	-0.20	0.684
$\beta_1 = \log(2) = 0.693$	5000	0.677	0.046	-2.36	< 0.001
	10000	0.676	0.037	-2.50	< 0.001
	50000	0.676	0.015	-2.49	< 0.001
$\beta_1 = \log(5) = 1.609$	5000	1.500	0.079	-6.83	< 0.001
	10000	1.496	0.053	-7.04	< 0.001
	50000	1.488	0.021	-7.53	< 0.001

Interestingly, although some of the the β_1 estimates from the individual bins have a significantly large % relative bias, the overall β_1 estimate from the regression has a much smaller bias. This may be indication that even if all the individual β_1 estimates are underestimated, the slope of regression line between these points is close to the true value. This is most notable in the large effect size where the bias for individual bin estimates are as large as 12%, yet the bias in the regression estimates are all approximately 7%. Although there is a decrease in bias, the bias is still statistically significant for a large effect size.

6.2.2 Simulation 10: Plotting bin estimates against mean bin values to detect relationship with the outcome for a continuous predictor

In practice, the relationship of a continuous predictor with the outcome may not always be linear. As seen above when the relationship is linear and the effect size is small, the regression of the bin estimates $\hat{\hat{\beta}}_j$ on the mean x bin values is an effective way to estimate the effect of a continuous predictor using RFPMs. When the relationship is unknown, plotting the bin estimates $\hat{\hat{\beta}}_j$ against the mean x bin values may

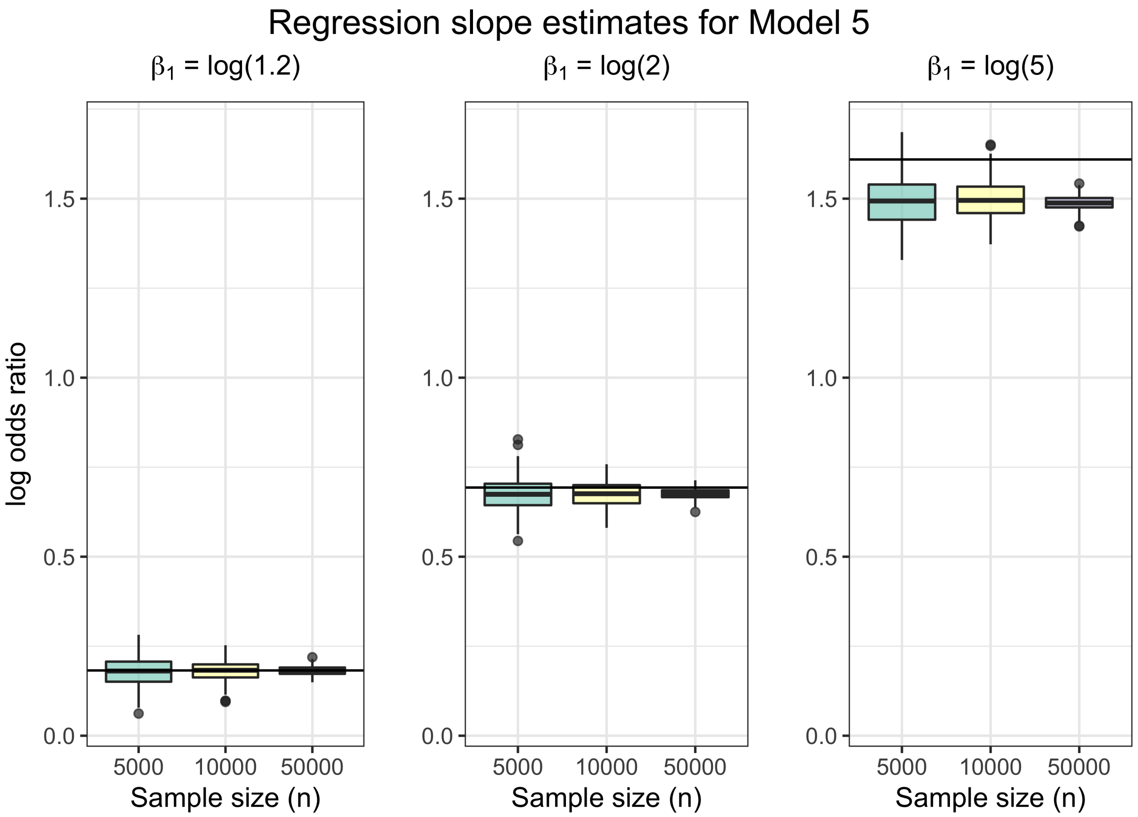


Figure 6.3: Regression β_1 slope estimates (log odds ratios) for varying β_1 values of log(1.2), log(2), and log(5) in Model 5. β_1 estimates obtained by regressing $\hat{\hat{\beta}}_j$ on the difference in mean bin values $\bar{x}_j$. The three box plots per plot are the estimates for the varying sample sizes in increasing order. The true parameter value indicated by the horizonal line.

provide insight into this relationship and further techniques may be used to estimate the effect. For example, consider the linear relationship of x_1 with y in Model 5 where $\beta_1 = \log(1.2)$. A single data set with $n = 5000$ is generated and continuous predictor x_1 is split into 10 bins. For this example, 10 bins are used to better gauge the type of relationship whereas, once the relationship is established, less bins may be used to estimate the effect. Using RFPMs, 9 log odds ratio estimates $\hat{\hat{\beta}}_j$ for $j \in \{2, 3, \ldots, 10\}$ are obtained and plotted against the mean x bin values $\bar{x}_j$ as depicted in Figure 6.4. The $\hat{\hat{\beta}}_j$ estimates align in an approximately straight line with a positive slope, indicating that the effect of x_1 is positive and it has a linear relationship with the outcome.

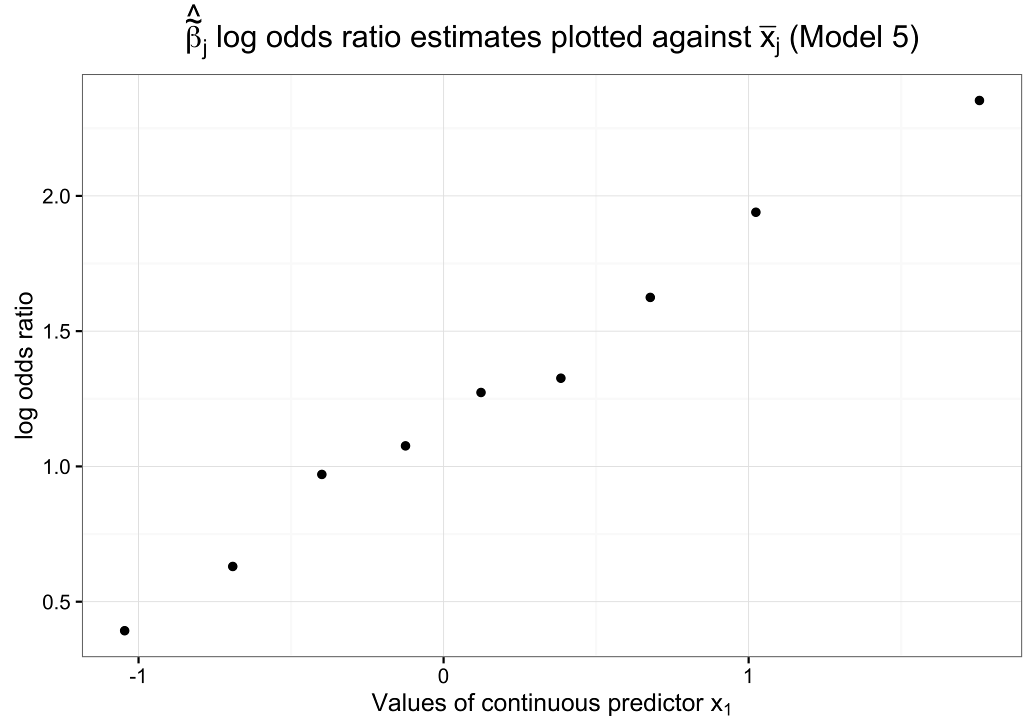


Figure 6.4: Plot of bin $\hat{\hat{\beta}}_j$ log odds ratio estimates for $j \in \{2, \ldots, 10\}$ for continuous $N(0,1)$ predictor x_1 with a main effect of $\beta_1 = \log(1.2)$ from Model 5. Each $\hat{\hat{\beta}}_j$ for $j \in \{2, \ldots, 10\}$ is plotted against the mean bin x value $\bar{x}_j$.

As a second example, consider the nonlinear relationship of x_1 with y in Model 6,

$$\text{logit}(p) = \beta_0 + \beta_1\, x_1 + \beta_1\, x_1^2 + 0\, x_2 + \ldots 0\, x_{10}$$

where x_1 is an independent and normally distributed random variable with a mean of 0 and a variance of 1. The coefficient for both the x_1 and x_1^2 terms is chosen to be log(1.2). The remaining predictors $(x_2, \ldots, x_{10})$ are independently generated, binary and have no association with the outcome. The results in section 6.2 aren't applicable when the relationship is nonlinear, but plotting the bin estimates $(\hat{\hat{\beta}}_j)$ against bin means $(\bar{x}_j)$ may provide insight into the relationship. A single data set with $n = 5000$ is generated and continuous predictor x_1 is split into 10 bins as before. Using RFPMs, 9 log odds ratio estimates $\hat{\hat{\beta}}_j$ for $j \in \{2, 3, \ldots, 10\}$ are obtained and plotted against the mean x bin values $(\bar{x}_j)$ as depicted in Figure 6.5. The estimates no longer align in an approximately straight line, but now exhibit curvature associated with a quadratic relationship. From this figure the researcher may conclude that the relationship is nonlinear and the coefficients of the x_1 and x_1^2 terms are positive. Instead of fitting a straight line, the researcher may apply other techniques such as spline regression to obtain an approximate estimate for β_1.

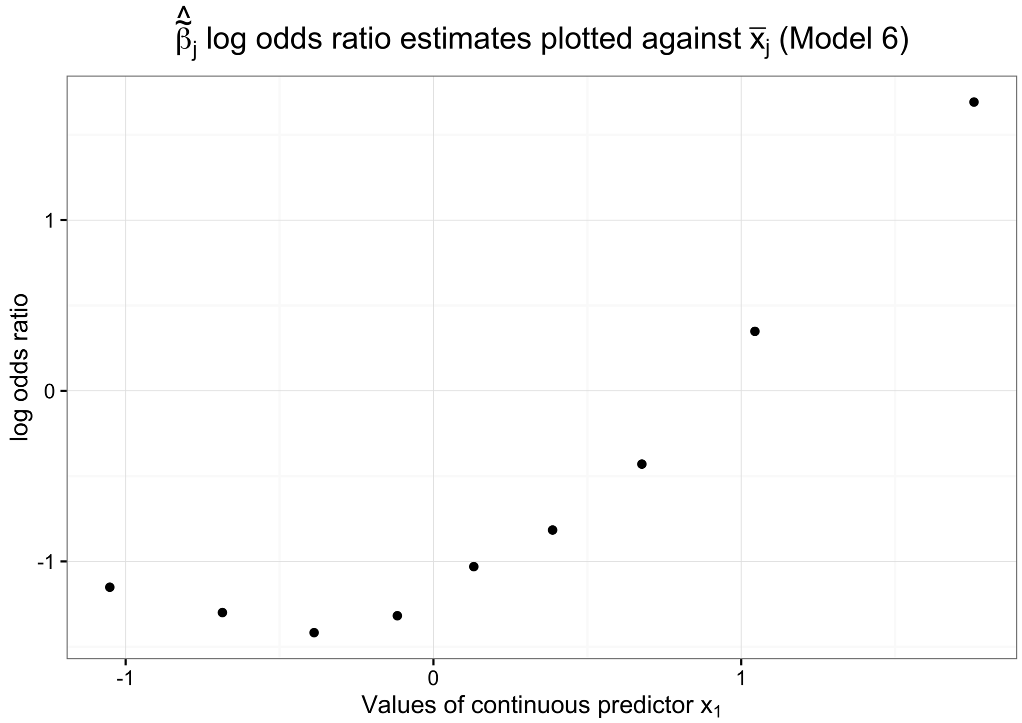


Figure 6.5: Plot of bin $\hat{\tilde{\beta}}_j$ log odds ratio estimates for $j \in \{2, \ldots, 10\}$ for continuous $N(0, 1)$ predictor x_1 with $\beta_1 = \log(1.2)$ from Model 6. Each $\hat{\tilde{\beta}}_j$ for $j \in \{2, \ldots, 10\}$ is plotted against the mean bin x value $\bar{x}_j$.

Chapter 7

Identifying risk factors for fetal growth abnormalities using random forest probability machine methods

Infants with a birth weight below the 10th percentile (SGA; small for gestational age) for gestational age and sex are at a higher risk for perinatal mortality and morbidity. SGA infants also have higher health care utilization, including during the delivery admission and for re-admissions within two weeks of delivery, compared to infants born appropriate for gestational age.

Knowledge of the risk factors for SGA births has important implications for preconception counselling and antenatal risk assessment. Smoking, young maternal age, previous birth of a low birth weight infant, lower socioeconomic status, hypertensive disorders of pregnancy, and maternal underweight have all been identified as risk factors for having an SGA infant. Conversely, high body mass index and diabetes in pregnancy are protective factors.

The objective of this analysis is to use the random forest probability machine methods presented in Chapters 2 through 6 to identify risk factors for SGA births using data from the Nova Scotia Atlee Perinatal Database (NSAPD).

7.1 Data set

7.1.1 Source

The NSAPD contains information on routine demographic variables, medical conditions, reproductive history, delivery events, and neonatal outcomes for each birth to mothers resident in Nova Scotia. The data is entered by trained coders using information from standardized clinical forms. Nova Scotia uses both a standard Prenatal Record and forms completed at the time of the hospital delivery admission to document prenatal care and information relevant to care and medical research. The NSAPD is administered by the Reproductive Care Program of Nova Scotia which also maintains the coding system, ensures the quality, integrity and security of the data. Using the NSAPD, a retrospective cohort of singleton infants born in Nova Scotia between January 1, 2009 and December 31, 2014 was established.

7.1.2 Outcome and predictors

The primary outcome is SGA ($< 10^{\text{th}}$ percentile birth weight for gestational age and sex), or not SGA ($\geq 10^{\text{th}}$ percentile birth weight for gestational age and sex) relative to the Canadian reference population published by Kramer et al. (2001). The outcome prevalence for the dataset used in this analysis is 0.06.

Demographic and clinical characteristics recorded in the NSAPD that were available at 26 weeks are used as predictors (Table 7.1). The analysis was restricted to women who have previously given birth (multiparous women) as their predictor set, in addition to their sociodemographic and current pregnancy information, also includes information from previous pregnancies.

Area-level household income quintile is calculated from the adjusted annual income based on census data averaged over all households in a postal code area. Area of residence is determined from the mother's postal code at the time of pregnancy. Any smoking at 20 weeks or at the labour admission is used as a proxy for smoking at 26 weeks. Pre-pregnancy body mass index (BMI) is based on height and weight information collected by self-report at the first prenatal visit. Sample characteristics for predictors in Table 7.1 using only complete data on multiparous women (N=15,771) are reported in Table 7.2.

7.2 Analysis

The use of RFPMs to estimate log odds ratios for binary, categorical, and continuous predictors, as well as corresponding confidence intervals, is outlined in Chapters 2 through 6. So far, use of RFPMs has only been demonstrated using simulated data and have yet to be applied to a real life data set. In the following section, RFPMs are used to estimate the association of the potential risk factors described in Table 7.1 with the risk for SGA. Ignoring any possible interaction effects, main effects for all predictors are estimated using the appropriate application of RFPMs. Interactions

Table 7.1: Demographic and clinical characteristics of predictors used in real life data analysis recorded in the Nova Scotia Atlee Perinatal Database (NSAPD).

Description	Type	Identifier	Values
Sociodemographics			
Maternal age	Continuous	matage	
Common-law/married	Binary	clmarried	(1 - common-law/ married, 0 - single)
Area-level income quintile	Categorical	ses5	(Q1, Q2, Q3, Q4, Q5)
Area of residence (rural)	Binary	rural	(1 - rural, 0 - urban)
Pregnancy risk factors			
Smoking before pregnancy	Binary	ppsmk	(1 - yes, 0 - no)
Pre-pregnancy body mass index	Continuous	ppbmi	
Pre-existing hypertension	Binary	prexht	(1 - yes, 0 - no)
Pre-existing diabetes	Binary	prexdm	(1 - yes, 0 - no)
Pregnancy history			
Gravidity	Categorical	gravid	(2, 3, $\geq$ 4)
Parity	Categorical	parity	(1, 2, $\geq$ 3)
Previous gestational diabetes	Binary	prvgdm	(1 - yes, 0 - no)
Previous child birth weight $<$ 2500g	Binary	prvlbw	(1 - yes, 0 - no)
Previous child birth weight $>$ 4080g	Binary	prvbig	(1 - yes, 0 - no)
Previous caesarean section	Binary	prvcs	(1 - yes, 0 - no)
Previous preterm delivery $<$ 29wks	Binary	prvpt29	(1 - yes, 0 - no)
Previous preterm delivery 29-32wks	Binary	prvpt2932	(1 - yes, 0 - no)
Previous preterm delivery 33-36wks	Binary	prvpt3336	(1 - yes, 0 - no)
Previous death of a neonate $\geq$ 500g	Binary	prvnnd	(1 - yes, 0 - no)
Current pregnancy			
Fetal sex	Binary	sex	(1 - male, 0 - female)
Weight gain in pregnancy at 26wks	Continuous	pwtgain26w	
Smoking during pregnancy	Binary	smk	(1 - yes, 0 - no)
Substance use in pregnancy	Binary	chabus	(1 - yes, 0 - no)
Gestational diabetes	Binary	gdm	(1 - yes, 0 - no)
Pregnancy-induced hypertension	Binary	pih	(1 - yes, 0 - no)
Psychiatric disorder	Binary	psych	(1 - yes, 0 - no)

Table 7.2: Sample characteristics for real life data set predictors using only complete information on multiparous women (N=15771). Data are presented as proportions unless otherwise indicated.

Description	Identifier	Sample characteristics
Sociodemographics		
Maternal age	matage	Mean 30.4 (SD 5.1)
Common-law/married	clmarried	0.79
Area-level income quintile	ses5	0.17/0.21/0.23/0.21/0.17
Area of residence (rural)	rural	0.30
Pregnancy risk factors		
Smoking before pregnancy	ppsmk	0.24
Pre-pregnancy body mass index	ppbmi	Mean 26.7 (SD 6.6)
Pre-existing hypertension	prexht	0.01
Pre-existing diabetes	prexdm	0.01
Pregnancy history		
Gravidity	gravid	0.45/0.29/0.26
Parity	parity	0.67/0.23/0.10
Previous gestational diabetes	prvgdm	0.03
Previous child birth weight $<$ 2500g	prvlbw	0.07
Previous child birth weight $>$ 4080g	prvbig	0.12
Previous caesarean section	prvcs	0.25
Previous preterm delivery $<$ 29wks	prvpt29	0.01
Previous preterm delivery 29-32wks	prvpt2932	0.01
Previous preterm delivery 33-36wks	prvpt3336	0.05
Previous death of a neonate $\geq$ 500g	prvnnd	$<$ 0.01
Current pregnancy		
Fetal sex	sex	0.51
Pregnancy Weight gain at 26wks	pwtgain26w	Mean 7.9 (SD 3.3)
Smoking during pregnancy	smk	0.19
Substance use in pregnancy	chabus	0.02
Gestational diabetes	gdm	0.06
Pregnancy-induced hypertension	pih	0.01
Psychiatric disorder	psych	0.12

are then estimated using information provided from the intuitive interaction detection method and a content expert.

7.2.1 Estimating main effects using RFPMs

Ignoring the potential presence of interactions, main effects are estimated using various applications of RFPMs. For binary predictors, the two-machine RFPM method outlined in Chapter 2 is used to obtain odds ratio estimates. For the three categorical predictors, *ses5*, *gravidity*, and *parity*, k RFPMs are used to obtain odds ratios for each level relative to the first, where k is the number of levels of the categorical predictor. For the three continuous predictors, $matage$, $ppbmi$, and $pwtgain26w$, 10 splits are made, and individual log odds ratio bin estimates $\hat{\hat{\beta}}_j$ for $j \in \{1, \ldots, 10\}$ are plotted against bin means to determine the relationship with the outcome. Odds ratio estimates for continuous predictors will be based on 5 bins once a relationship is established. For all odds ratio estimates, 95% confidence intervals are constructed using the bootstrap percentile method.

Odds ratio estimates and corresponding 95% confidence intervals for binary predictors are shown in Table 7.3. A previous child birth weight less than 2500g ($prvlbw$), a previous child birth weight greater than 4080g ($prvbig$), a previous pre-term delivery at 29-32 weeks ($prvpt2932$), a previous pre-term delivery at 33-36 weeks ($prvpt3336$), pregnancy-induced hypertension (pih), pre-existing hypertension ($prexht$), marital status ($clmarried$), smoking before pregnancy ($ppsmk$), smoking during pregnancy (smk), and substance use during pregnancy ($chabus$) are all associated with having a SGA baby.

The most strongly positively associated predictors (odds ratio greater than 3), are $prvlbw$, smk, and $chabus$. This indicates that mothers who have previously given birth to an infant with a birth weight less than 2500g, mothers who reported smoking during pregnancy, and mothers who reported substance use during pregnancy all have greater odds of having a SGA birth. However, mothers who have previously given birth to an infant with a birth weight greater than 4080g have lower odds of having a SGA birth relative to mothers who previously have not given birth to a large infant. Interestingly, the common-law/married predictor is significant, indicating that mothers who are married/common-law have lower odds of having a SGA baby relative to mothers who are single. This predictor may be acting as a proxy for maternal age (*matage*) since it is known that younger (and most likely single) mothers are at a greater risk of having a SGA baby.

Table 7.3: Odds ratios and corresponding 95% confidence intervals for binary predictors constructed using the two-machine RFPM method. Predictors are presented in the same order as in Tables 7.2 and 7.3.

Predictor	**Odds ratio estimate**	**95% confidence interval**
clmarried	0.63	[0.52, 0.74]
rural	1.01	[0.85, 1.20]
ppsmk	2.66	[2.27, 3.15]
prexht	2.06	[1.27, 3.32]
prexdm	0.68	[0.23, 1.34]
prvgdm	0.92	[0.51, 1.40]
prvlbw	4.05	[3.37, 4.82]
prvbig	0.21	[0.13, 0.30]
prvcs	1.02	[0.85, 1.23]
prvpt29	1.90	[0.91, 3.04]
prvpt2932	2.31	[1.19, 3.50]
prvpt3336	1.71	[1.28, 2.35]
prvnnd	1.61	[0.49, 3.07]
sex	1.11	[0.95, 1.25]
smk	3.39	[2.84, 3.86]
chabus	3.27	[2.22, 4.57]
gdm	1.07	[0.80, 1.42]
pih	2.92	[1.35, 4.41]
psych	1.19	[0.92, 1.49]

Odds ratio estimates with corresponding 95% confidence intervals for the three categorical variables are shown in Table 7.4. Mothers who have been pregnant more than once (*gravid*) and mothers who have given birth more than once (*parity*) are significantly more likely to have a SGA baby. The area-level income quintiles predictor (*ses*5) has five levels, which are ordered from lowest to highest income. Quintiles 2 to 5 are all associated with a lower odds of having a SGA baby relative to Quintile 1. The association is statistically significant.

Table 7.4: Odds ratios and corresponding 95% confidence intervals for categorical predictors constructed using RFPMs. Odds ratios are calculated for each level of the categorical predictor relative to the baseline (first level). Predictors are presented in the same order as in Tables 7.2 and 7.3.

Predictor	**Level**	**Odds ratio estimate**	**95% confidence interval**
ses5	Q1	-	-
	Q2	0.81	[0.78, 0.84]
	Q3	0.76	[0.73, 0.78]
	Q4	0.68	[0.67, 0.72]
	Q5	0.74	[0.72, 0.77]
gravid	2	-	-
	3	1.06	[1.02, 1.07]
	≥ 4	1.26	[1.19, 1.26]
parity	1	-	-
	2	1.09	[1.04, 1.10]
	≥ 3	1.54	[1.48, 1.59]

The maternal age predictor should have an approximate U-shaped relationship with having a SGA infant. Young women are at a high risk and then this risk drops until about age 35, after which it increases again because the placenta in older women cannot support the fetus as effectively. With this knowledge, the relationship in Figure 7.1 could be considered U-shaped, but the estimates have a high degree of variability. One could also conclude that there is no association between maternal age and SGA due to the scatter of the estimates. This may also be an indication that more than nine estimates are required to visualize this relationship in a plot. The relationship between pre-pregnancy body mass index and SGA is negative and appears to be linear apart from a few estimates (Figure 7.2). There is also a negative linear relationship between pregnancy weight gain at 26 weeks ($pwtgain26wk$) and SGA (Figure 7.3). Both of these plots are in keeping with the literature as mothers with a larger BMI and mothers who gain more weight during pregnancy are less likely to have a SGA infant.

The odds ratios and 95% confidence intervals for the continuous predictors are shown in Table 7.5. For *matage*, the baseline group consists of mothers between the ages of 16 and 26 years. Mothers between the ages of 26 and 32 years have lower

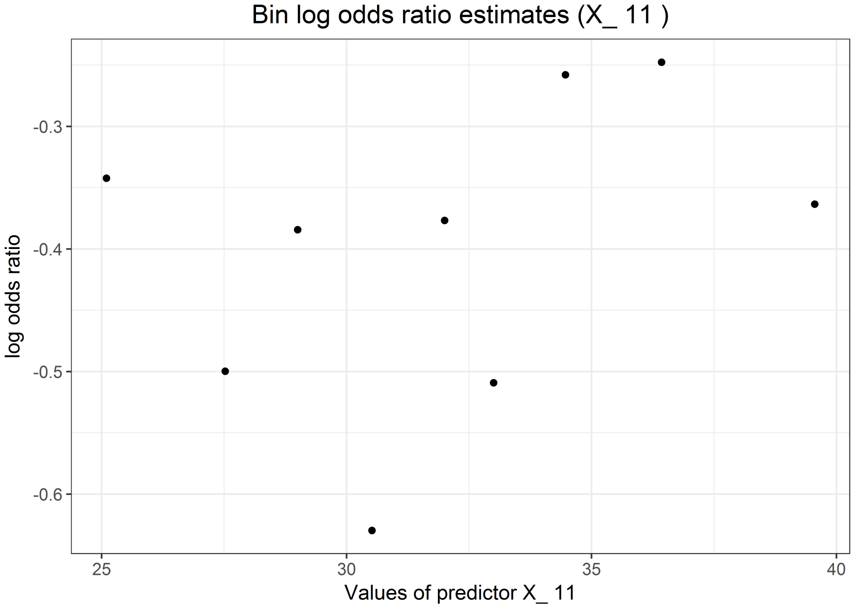


Figure 7.1: Log odds ratio bin estimates $\hat{\hat{\beta}}_j$ calculated relative to the first bin plotted against bin means $\bar{x}_j$ for $j \in \{2, \ldots, 10\}$ for *matage* (maternal age) predictor.

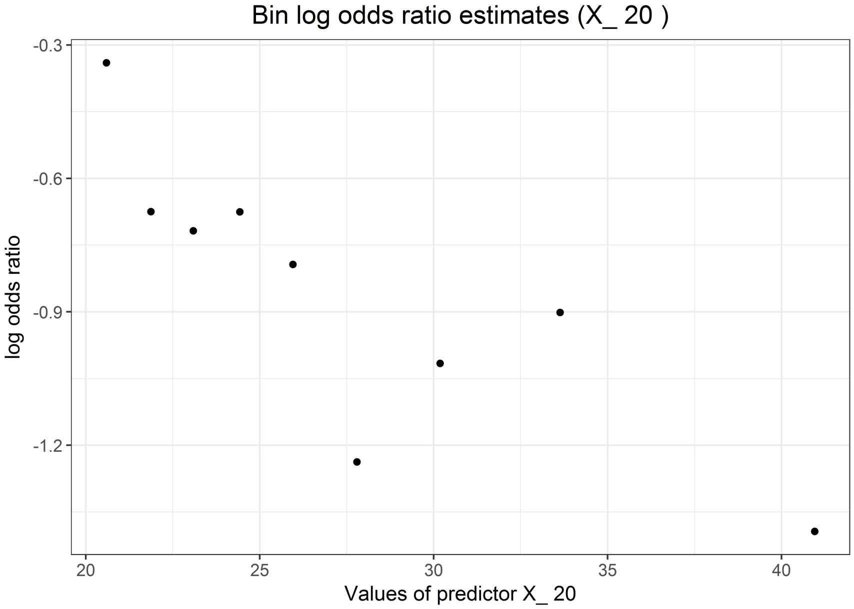


Figure 7.2: Log odds ratio bin estimates $\hat{\hat{\beta}}_j$ calculated relative to the first bin plotted against bin means $\bar{x}_j$ for $j \in \{2, \ldots, 10\}$ for *ppbmi* (pre-pregnancy body mass index) predictor.

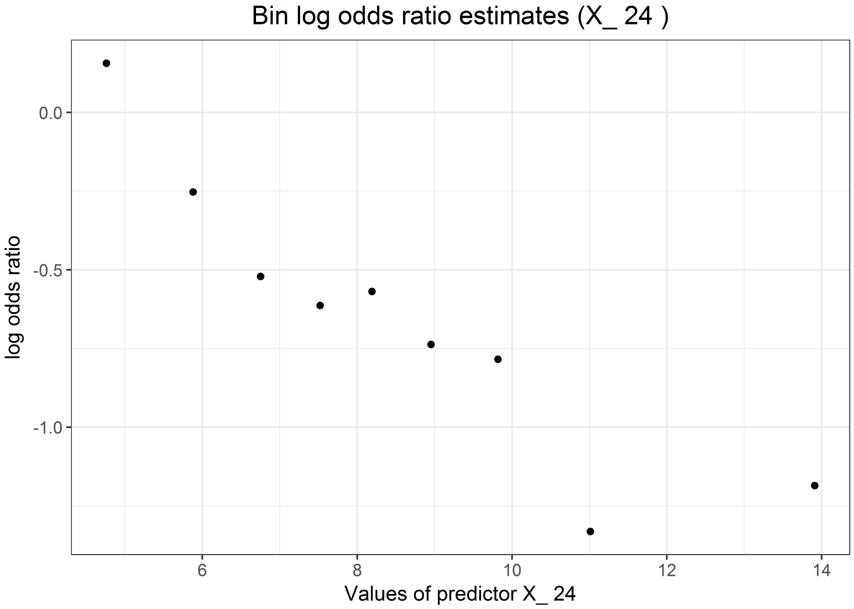


Figure 7.3: Log odds ratio bin estimates $\hat{\hat{\beta}}_j$ calculated relative to the first bin plotted against bin means $\bar{x}_j$ for $j \in \{2, \ldots, 10\}$ for $pwtgain26w$ (weight gain in pregnancy at 26 weeks) predictor.

odds of having a SGA infant, and as mothers get older (ages 32-51), the odds are getting closer to baseline. This relationship coincides with the expected U-shaped relationship between maternal age and SGA. Young mothers and older mothers are at a greater risk for having SGA babies relative to middle-aged mothers. For $ppbmi$, the baseline group consists of mothers with a pre-pregnancy BMI between 15.1 and 21.3 kg/m^2. With increasing pre-pregnancy BMI, the odds of having a SGA baby relative to baseline decreases. For $pwtgain26w$, the baseline group consists of mothers with a weight gain at 26 weeks between -13.7 and 5.41 kg (a negative weight gain indicates mother weighed less at 26 weeks than pre-pregnancy). Similarly to $ppbmi$, with increasing weight gain at 26 weeks, the odds of having a SGA baby relative to baseline decreases.

7.2.2 Detecting and estimating interaction effects using RFPMs

The main effects estimation presented in section 7.2.1 ignores the potential presence of interactions. Using the intuitive interaction detection method and the four-machine RFPM method outlined in Chapter 4, potential interaction effects can be quickly

Table 7.5: Bin odds ratio estimates and corresponding 95% confidence intervals for continuous predictors constructed using four RFPMs. Continuous predictors are split into five bins (quintiles) and odds ratios are obtained for each bin relative to baseline (first bin).

Predictor	Quintile	Odds ratio estimate	95% confidence interval
matage	[16, 26]	-	-
	(26, 29]	0.73	[0.73, 0.77]
	(29, 32]	0.71	[0.70, 0.74]
	(32, 35]	0.87	[0.85, 0.90]
	(35, 51]	0.93	[0.91, 0.96]
ppbmi	[15.1, 21.3]	-	
	(21.3, 23.7]	0.57	[0.56, 0.59]
	(23.7, 26.8]	0.56	[0.54, 0.56]
	(26.8, 31.7]	0.38	[0.35, 0.38]
	(31.7, 67.8]	0.39	[0.37, 0.39]
pwtgain26w	[-13.7, 5.41]	-	
	(5.41, 7.12]	0.60	[0.57, 0.61]
	(7.12, 8.51]	0.50	[0.47, 0.50]
	(8.51, 10.3]	0.39	[0.37, 0.40]
	(10.3, 24.5]	0.25	[0.23, 0.25]

screened for and estimated. The intuitive interaction detection method has only been formalized for interactions between binary predictors, and so any continuous or categorical predictors in Table 7.1 are omitted from the screening process. The interaction between two rare predictors is not entirely relevant for the purpose of this analysis. Thus, any pair of predictors with a product of prevalence rates less than 0.1 are not considered. The combinations of these predictors may create p_{11}, p_{10}, or p_{01} subgroups with very few observations or no observations at all. Since only a single RFPM is fitted in the screening process, misleading results can be obtained if subgroups have very small sample sizes.

To avoid calculating estimates and corresponding confidence intervals for all possible pairs of binary predictors, potential interactions are screened first using the intuitive interaction detection method (Table 7.6). In Simulations 3 and 4, an interaction was deemed relevant if its estimated log odds ratio (ℓROR) was either less than log(1/1.05) or greater than log(1.05) (or an odds ratio (*ROR*) less than 1/1.05 or greater than 1.05). Using these cutoff points, the effect of the interaction between predictors common-law/married (*clmarried*) and smoking status during pregnancy (*smk*); common-law/married (*clmarried*) and pre-pregnancy smoking status (*ppsmk*); common-law/married (*clmarried*) and mother's area of residence (*rural*); and, fetus sex (*sex*) and mother's area of residence (*rural*), are all potentially relevant. Some of these interactions may be spurious. Therefore, interpretation of the results from the screening process requires content expertise.

Table 7.6: Results of the intuitive interaction detection method where only binary predictors are considered. Any predictors with a product of prevalence rates less than 10% are omitted. For each pair of predictors, the multiplicative measure of interaction ROR, odds ratios OR_{01}, OR_{10} and, OR_{11}, and the additive measure of interaction $RERI_{RR}$, are reported. Predictor pairs are sorted by largest $|ROR - 1|$.

Predictors		ROR	OR_{01}	OR_{10}	OR_{11}	$RERI_{RR}$
clmarried	smk	1.40	2.26	0.74	2.27	0.26
clmarried	ppsmk	1.30	2.13	0.75	2.04	0.17
clmarried	rural	1.11	0.95	0.58	0.62	0.08
sex	rural	0.96	1.06	0.99	1.03	-0.03
prvcs	clmarried	0.99	0.60	0.96	0.57	0.01
sex	prvcs	0.99	0.92	0.99	0.90	-0.01
sex	clmarried	1.00	0.60	0.99	0.59	0.00
sex	ppsmk	1.00	2.67	1.00	2.67	0.00

The potential interaction between marital status (*clmarried*) and many other predictors is most likely due to it being a proxy for maternal age, a predictor known to have a strong association with SGA. Since predictors *smk* and *ppsmk* are very similar and highly correlated (correlation coefficient of 0.86), only the interaction between *clmarried* and *smk* is considered on account of smoking during the pregnancy being more relevant biologically. The interaction between *clmarried* and *rural* is not useful, and the latter may be a weak proxy for obesity (obesity rate is higher in women who live in rural areas). Similarly, the interaction between *sex* and *rural* is both spurious and irrelevant biologically, and is ignored.

Based on the results from the intuitive interaction detection method and knowledge from a content expert, only the interaction between *clmarried* and *smk* is further investigated. Using the four-machine RFPM method, odds ratios for main effects (OR_{10} and OR_{01}) and the interaction effect (ROR) are obtained (Table 7.7). The four-machine RFPM method can be used to obtain both main and interaction effects since the highest order term in the fully saturated model containing interacting predictors (only *clmarried* and *smk*) is the 2-way interaction ($2^2 = 4$ machines). All odds ratios reported are significant at the 5% significance level, indicating that *clmarried*, *smk* and their interaction are associated with SGA.

The odds ratios reported are calculated relative to the baseline group (OR_{00}), which is the group of single mothers who do not smoke during pregnancy. The odds ratio $OR_{10} = 0.71$ indicates that common-law/married mothers who do not smoke during pregnancy have lower odds of having a SGA baby relative to the baseline group, whereas $OR_{11} = 2.50$ indicates that common-law/married mothers who smoke during pregnancy have higher odds of having a SGA baby relative to baseline. The odds ratio $OR_{01} = 2.39$ indicates that single mothers who smoke during pregnancy have higher odds of having a SGA baby relative to baseline.

Table 7.7: Odds ratio estimates for main effects (OR_{10} and OR_{01}) and interaction effect (ROR) for predictor pair *clmarried* and *smk*. OR_{ij} is the odds ratio for *clmarried* $= i$ and *smk* $= j$ for $i\,j \in \{0, 1\}$. For each odds ratio, the corresponding 95% confidence interval is reported.

Predictors		Estimate	95% confidence interval
clmarried	smk	$OR_{11} = 2.50$	[1.95, 3.08]
		$OR_{10} = 0.71$	[0.58, 0.91]
		$OR_{01} = 2.39$	[1.89, 3.09]
		$ROR = 1.75$	[1.31, 2.39]

7.3 Conclusion

The current methods for estimating main and interaction effects for predictors associated with SGA have been exclusively based on conventional regression methods. RFPMs provide an alternative non-parametric approach for estimating these effects without imposing any restrictions on the data generating process. In this analysis, RFPMs identified the risk factors for SGA that are known to be associated from the literature. The use of RFPMs to estimate odds ratios for a linear continuous predictor using the concept of binning was helpful in visualizing the relationship as seen in Figure 7.3.

Although RFPMs provided a non-parametric approach to this analysis, several issues with the use of RFPMs in real life data were identified. The first issue is the problem of computation times. For this analysis, a typical 4-core CPU was used to compute all odds ratio estimates and 95% confidence intervals, which proved to be very time consuming. Secondly, in Simulations 9 and 10, the 95% confidence intervals constructed using the bootstrap percentile method were shown to have appropriate coverage probability. However, in the real life data, although all the point estimates are contained inside the 95% confidence interval, some of the estimates are on either the lower or upper bound of the interval. This implies that the distribution of the estimates may not be symmetric and either more bootstrap samples are required (rather than $b = 200$), or an alternative bootstrap method may be more appropriate.

A couple of issues arose with the use of RFPMs for continuous predictors and the concept of binning. The expected relationship between maternal age and SGA is U-shaped, but this relationship is not immediately apparent in Figure 7.1. One could just as easily conclude that there is not association between maternal age and SGA due to the random scatter of the estimates. This may be an indication that more estimates are required, or a different approach to binning is needed to visualize the relationship between predictors and the outcome when the relationship is complex and nonlinear. Using more estimates will increase the computation time required to compute estimates, and could potentially lead to creating bins with very little observations resulting in misleading estimates. Determining the number of bins to split the continuous predictor into requires consideration of both the outcome prevalence and the total sample size of the data set.

The intuitive interaction screening method obviated the need to estimate all possible interactions, but the method had several drawbacks. In the first application of this method, all possible binary predictors were considered (171 possible combinations), regardless of their prevalence rates. This produced 117 potentially relevant interactions using the cutoff points used in Simulations 3 and 4, many of which were either spurious or irrelevant biologically. Many of the rare predictors appeared to interact very frequently, but this may be due to the very small sample sizes of the p_{11}, p_{01}, p_{10} subgroups. Since there is only one RFPM fitted in this method, these estimates may be misleading when the sample sizes are small. Choosing which predictors to further investigate from the 117 proved to be challenging and led us to introduce threshold values with regard to the effect size and the prevalence of the respective predictor pair, and to use content expertise to further screen for relevant interactions.

Due to the complexity of interactions between predictors of different types, the intuitive interaction detection method has yet to be formalized for interactions between these predictors. Also, the estimation of the effect of these interactions becomes increasingly complex with the concept of binning. These interaction effect estimates between continuous and binary or categorical predictors may be difficult to interpret. Overall, the results from the RFPM analysis aligned well with the literature, but additional work is required reduce computation times and improve the estimation of continuous predictors.

Chapter 8

Discussion

The use of RFPMs proposed by Malley et al. (2012) to estimate main and interaction effects for binary predictors has been explored. A technique for estimating the main effects of categorical predictors and of continuous predictors using the concept of binning has been proposed. Using the bootstrap percentile method, confidence intervals for a risk estimate derived using RFPMs were constructed and shown to have appropriate coverage probability. RFPMs were then applied to a real life data set and were found to identify risk factors associated with SGA that aligned with what is known to be associated from the literature. The issues faced when using RFPMs in this analysis were also discussed.

Logistic regression is the most commonly used model for examining the association of potential risk factors with a binary outcome. This method is widely used due to its simplicity, ease of interpretation, and implementation in all major statistical software packages. However, for high dimensional data sets or data with complex relationships between the predictors and outcome (e.g. higher order interactions or non-linear relationships), logistic regression may not be able to fit the model correctly. Another drawback of the logistic regression model is that it produces odds ratio estimates, which may overestimate the relative risk for outcomes with a prevalence $> 10\%$.

RFPM methods deal with several of these issues in that they are completely non-parametric, and are applicable to large data sets and high dimensional problems. RFPM methods require only a specification of which predictors are to be included, rather than any explicit functional form. Since RFPMs estimate predicted counterfactual probabilities of success, individual risk estimates can be calculated; subgroup-specific risk estimates can be obtained by averaging over the appropriate individual estimates.

The two-machine and four-machine RFPM methods are used to estimate main and interaction effects of binary predictors. RFPMs were found to produce estimates with minimal bias and perform almost as efficiently as a correctly specified logistic regression model when the data generating model was logistic. However, correct main effect estimates can be difficult to obtain for predictors involved in many interactions. This requires the use of 2^m machines, where m is the highest order term in the fully saturated model of interacting predictors. This may not always be feasible, as increasing the number of machines may produce subgroups of the data set with very few or no observations.

The intuitive interaction detection method proved to be a relatively quick screening process to identify any potential interaction effects. Rather than estimating all possible combinations of predictors, this method can be used to quickly screen for potentially relevant interactions. However, this method may produce an overwhelming number of irrelevant or spurious interactions, and it must be used with caution and knowledge from a content expert. The interactions between rare predictors may produce subgroups with very few observations, and may produce misleading estimates.

RFPMs can also be used to estimate the effects of categorical predictors by extending the methodology behind the two-machine RFPM. Estimates with minimal bias were produced by using k machines, where k is the number of levels of the predictor, and performed comparably to a correctly specified logistic regression model when the data generating process was logistic. Estimating the effects for a continuous predictor is much more complex and a possible solution involves the concept of binning (described in Chapter 6.2). This method was shown to produce estimates with minimal bias when the effect size was small, but the bias became substantial (around 10%) at an effect size of log(5). Interaction detection and estimation that involves continuous variables is challenging due to the required pre-processing (binning) of the continuous variables.

The relationship between the outcome and a continuous predictor may not always be linear. A possible way to visualize relationships is to plot the individual bin estimates $\hat{\hat{\beta}}_j$ against the bin means, $\bar{x}_j$. This method proved to be valuable when the relationship between the predictor and the outcome was simple and linear, whereas complex and non-linear relationships were less apparent. This may be an indication that more bins are required to examine complex relationships, or an alternative approach may be needed.

The bootstrap percentile method was proposed as an approach to constructing confidence intervals for risk effects derived from RFPMs. In Simulations 6 and 7, this method was shown to produce confidence intervals with appropriate coverage probabilities. However, constructing confidence intervals with a large number of bootstrap samples was very computationally intensive and time consuming. When applied to the real life data set, some of the estimates were on either the lower of upper bound of the interval. This observation suggests that an alternative bootstrap method may be more appropriate or more bootstrap samples are required.

An issue that was not explored in depth is the issue of correlation among predictors. Correlated predictors were briefly explored Simulations 3 and 4 (section 4.1.1) when it was found that complex correlations among predictors led to an increase in false positive rates. Preliminary simulations (data not shown) demonstrated that effect estimates for correlated predictors from RFPMs were consistently biased toward the Null when the correlation was greater than 0.2. The source of this bias is unknown, but may relate to the way trees are built in the random forest. Correlations among predictors are relatively common in health research (e.g. correlations between environmental chemical metabolites), and their effect on RFPM estimates may be substantial.

The use of RFPMs for risk estimation in epidemiological studies still requires additional work. These methods are computationally intensive, and there is no formalized method for estimating interactions between different types of predictors. Future research on the topic should consider developing such methodology, alternative methods for confidence interval construction, and exploring the effect of correlations between predictors on RFPM estimates and determining the source of the observed bias.

Bibliography

Analytics, R. and Weston, S. (2015a). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package.* R package version 1.0.10.

Analytics, R. and Weston, S. (2015b). *foreach: Provides Foreach Looping Construct for R.* R package version 1.4.3.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees.* CRC.

Dasgupta, A., Silke, S., Moore, J. H., Bailey-Wilson, J. E., and Malley, J. D. (2014). Risk estimation using probability machines. *BioData Mining*, 7(2).

Hernan, M. A. (2004). A definition of causal effect for epidemiological research. *J Epidemiol Community Health*, (265-271).

Imbens, G. W. (2004). Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, 86(1):4–29.

Kramer, M. S., Platt, R. W., Wen, S. W., Joseph, K., Allen, A., Abrahamowicz, M., Blondel, B., and Breart, G. (2001). A new and improved population-based canadian reference for birth weight for gestational age. *Pediatrics*, 108.

La Torre, G. (2010). *Analysis of Cost Data Using Bootstrap Technique*, page 249–259. SEEd Medical Publishers, 1 edition.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Malley, J., Kruppa, J., Dasgupta, A., Malley, K., and Ziegler, A. (2012). Probability machines: Consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine*, 51:74–81.

Snowden, J. M., Rose, S., and Mortimer, K. M. (2011). Implementation of g-computation on a simulated data set: demonstration of a causal inference technique. *American Journal of Epidemiology*, page kwq472.

Szklo, M. and Nieto, J. (2014). *Measures of Disease Occurrence and Association*, page 45–101. Jones & Bartlett Publishers Inc, 3 edition.

Wright, M. N. and Ziegler, A. (2015). ranger: A fast implementation of random forests for high dimensional data in c++ and r. *arXiv preprint arXiv:1508.04409*.

Zhang, J. and Yu, K. F. (1998). Whats the relative risk? a method of correcting the odds ratio in cohort studies of common outcomes. *JAMA*, 280(19):1690–1691.

Appendix A

R code

The R code in the following sections is used to implement the intuitive interaction screening method, the two-machine RFPM method, the four-machine RFPM method, use of RFPMs for a categorical predictor with k levels, and use of RFPMs for a discretized continuous predictor split into k bins. The code provided was used to obtain the results from Simulations 1-10 and the results from Chapter 7. The functions provided have a detailed introduction outlining its use, input parameters and output results.

A.1 Intuitive interaction screening method

```
# FUNCTION TO PERFORM DASGUPTA’S "Intuitive Model- Free Interaction Screening Method" IN
# "Risk Estimation using probability machines".
# INTERACTION DETECTION ONLY FOR BINARY PREDICTORS

#INPUT:
#   dataset: dataframe or matrix where each column represents one predictor and the last column is the response vector
#            (each row corresponds to 1 observation)
#   n.tree: number of trees to grow in single RFPM fit to the data
#   m.try: number of predictors randomly sampled as candidates at each split in single RFPM fit to the data
#   thres.p: prevalence cutoff point where if the product of any predictor prevalences is less than p.thres, no interaction
#     estimate will be computed.  If no cutoff point is desired, (ie. compute all interactions regardless of prevalence), set
#     thres.p = 0

#OUTPUT:
#   ’Intuitive Interaction Screening’: matrix indicating potential interactions in the model found by fitting
#           one RFPM.  First two columns are the two predictors involved (x_j, x_k) and remaining are the interaction quantities:
#           1. log ratio of odds ratios: log of the ratio of p_11(1-p_10)/p_10(1-p_11) and p_01(1-p_00)/p_00(1-p_01))
#           2. OR_01: odds ratio p_01(1-p_00)/p_00(1-p_01)
#     3. OR_10: odds ratio p_10(1-p_00)/p_00(1-p_10)
#     4. OR_11: odds ratio p_11(1-p_00)/p_00(1-p_11)
#     5. Relative Excess Risk due to Interaction (RERI): RERI = p_11/p_00 - p_01/p_00 - p_10/p_00 + 1
#
#        *NOTE*:  1 is used to detect multiplicative interactions and 5 is used to detect additive interactions.  ORs (2-4) are
#             provided for insight regarding main effect estimates.  Interactions are sorted by largest lROR in absolute value.

#Libraries
library(randomForest)
library(gtools)
library(foreach)
library(doParallel)

int.detection_final <- function(dataset, n.tree, m.try, thres.p){

  #Set up dataframe
  n.col <- ncol(dataset)
```

```r
data_mat <- data.frame(predictors=dataset[, 1:(n.col-1)], response=dataset[, n.col])

#Set nodesize = 5% of dataset
node.size <- 0.05*nrow(data_mat)

#Determine number of continuous and categorical predictors:
unique.val <- apply(data_mat, 2, unique)
unique.val.length <- as.vector(unlist(lapply(unique.val, length))) #Determine number of unique values in each column of data.mat
which.contcat.preds <- which(unique.val.length > 2)

## Intuitive Model-Free Interaction Detection ------------------------------
#This method is done on dataset containing no continuous or categorical predictors or their "bins" - nonsensical for
#a "bin" of one continuous predictor to interact with another bin and/or a binary predictor.  Only binary
#predictors are considered at this time

#Fit single RFPM to the data (P(Y=1)|X) - only binary predictors
PM_iid <- randomForest(response ~., data=data_mat, ntree=n.tree, mtry=m.try, nodesize=node.size)

#Create vectors to store ratio of odds ratios and RERI for each combination of X_j X_k
comb <- combinations((n.col-1), 2)

#If there are continuous/categorical predictors in the dataset, remove their predictor index from comb
if (length(which.contcat.preds)!=0){
  #Remove any rows of comb that contain index of continuous predictor
  rm.rows <- rep(NA, 0)

  for (i in 1:length(which.contcat.preds)){
    contcat.pred.i <- which.contcat.preds[i]
    rm.rows <- rbind(rm.rows, which(comb==contcat.pred.i, arr.ind = TRUE)[,1])
  }

  rm.rows <- unique(as.vector(rm.rows))

  #Comb only contains binary predictors
  comb <- comb[-rm.rows,]}

odds.iid <- matrix(NA, ncol=4, nrow=0)
RERI.iid <- matrix(NA, ncol=1, nrow=0)

#Compute p00, p01, p10, p11 for combinations of (X_m, X_n) and calculate ratio of odds ratio and RERI
for (r in 1:nrow(comb)){

  print(r)

  m <- comb[r,1]
  n <- comb[r,2]

  x_j <- data_mat[,m]; x_k <- data_mat[,n ]
  prev.x_j <- mean(x_j); prev.x_k <- mean(x_k)

  #Predicted probabilities conditional on the four possible combinations over (Xj, Xk)
  p00 <- predict(PM_iid, newdata=subset(data_mat, data_mat[,m]==0 & data_mat[,n]==0))
  p01 <- predict(PM_iid, newdata=subset(data_mat, data_mat[,m]==0 & data_mat[,n]==1))
  p10 <- predict(PM_iid, newdata=subset(data_mat, data_mat[,m]==1 & data_mat[,n]==0))
  p11 <- predict(PM_iid, newdata=subset(data_mat, data_mat[,m]==1 & data_mat[,n]==1))

  lengths <- c(length(p00), length(p01), length(p10), length(p11))

  if(prev.x_j*prev.x_k < thres.p){

    print("Product prevelances of predictors less than 0.1:")
    print(c(m,n))

    lROR.iid <- NA
    RERI <- NA

    odds.iid <- rbind(odds.iid, lROR.iid)
    RERI.iid <- rbind(RERI.iid, RERI)

  } else if(any(lengths==0)){

    print("One of G_00, G_01, G_10, G_11 contains no observations for predictors:")
    print(c(m,n))

    lROR.iid <- NA
    RERI <- NA

    odds.iid <- rbind(odds.iid, lROR.iid)
    RERI.iid <- rbind(RERI.iid, RERI)

  } else{

  #Mean of the logit of the conditional probabilities
  p00.avlogit <- mean(logit(p00))
  p01.avlogit <- mean(logit(p01))
```

```r
    p10.avlogit <- mean(logit(p10))
    p11.avlogit <- mean(logit(p11))

    lROR.iid <- p11.avlogit - p10.avlogit - p01.avlogit + p00.avlogit

    #Mean of the conditional probabilities
    p00.av <- mean(p00)
    p01.av <- mean(p01)
    p10.av <- mean(p10)
    p11.av <- mean(p11)

    #Odds ratios relative to baseline
    OR_01 <- (p01.av*(1-p00.av))/(p00.av*(1-p01.av))
    OR_10 <- (p10.av*(1-p00.av))/(p00.av*(1-p10.av))
    OR_11 <- (p11.av*(1-p00.av))/(p00.av*(1-p11.av))

    odds.iid <- rbind(odds.iid, c(lROR.iid, OR_01, OR_10, OR_11))

    #Additive Interactions (RERI_RR - relative excess risk due to interaction)
    RERI <- p11.av/p00.av - p01.av/p00.av - p10.av/p00.av + 1
    RERI.iid <- rbind(RERI.iid, RERI)}}

  #Store results in one matrix
  iid.odds.N.RERI.round <- round(cbind(odds.iid, RERI.iid), digits=3)
  iid.odds.N.RERI <- cbind(comb, iid.odds.N.RERI.round)
  colnames(iid.odds.N.RERI) <- c("X_m", "X_n", "lROR", "OR_01", "OR_10", "OR_11", "RERI (add)"); rownames(iid.odds.N.RERI) <- NULL

  #Only show predictors combinations with approporiate prevelances and subgroup sizes
  iid.odds.N.RERI.complete <- iid.odds.N.RERI[complete.cases(iid.odds.N.RERI),]

  #Sort the rows by interactions with largest ratio of odds ratios (in abs value) different from 1
  order.abs <- order(abs(0-iid.odds.N.RERI.complete[,3]), decreasing=TRUE)
  iid.odds.N.RERI_final <- iid.odds.N.RERI.complete[order.abs, ]

  ### Output -----------------------------------------------------------------
  return('Intuitive Model-Free Interaction Screening'= iid.odds.N.RERI_final)}
```

R_code/int.detection_final.R

A.2 Two-machine RFPM method

```
#FUNCTION TO COMPUTE LOG ODDS RATIO ESTIMATES AND CORRESPONDING 95% CONFIDENCE INTERVALS USING DASGUPTA'S
#TWO-MACHINE RFPM METHOD IN "Risk Estimation using Probability Machines" (binary predictors)

# *** Runs in parallel using doParallel and foreach ***

#INPUT:
#   dataset: dataframe or matrix where each column represents one predictor and the last column is the response vector
#            (each row corresponds to 1 observation)
#   n.tree: number of trees to grow in RFPM_0 and RFPM_1
#   m.try: number of predictors randomly sampled as candidates at each split in RFPM_0 and RFPM_1
#   test.main: vector where each element indicates the predictor index for which the main effect estimate
#          is to be calculated (eg. test.main = c(2,6) will calculate main effects for predictors x_2 and
#      x_6 in data set (columns 2&6))

#OUTPUT:
#   'Log odd Ratio Estimates (binary)': two-machine RFPM method used to estimate log odds ratios for any binary predictors
#      not involved in any interactions.  First column is predictor and second column is the following log odd ratio estimate:
#          1. lOR = log (p_1(1-p_0)/p_0(1-p_1))
#
#   '95% Confidence interval for log odd ratio estimates (binary)': two-machine RFPM method and bootstrapping (b=200) used to
#      construct 95% confidence intervals for lOR estimates from 'Log odd Ratio Estimates'.  First column is predictor
#      and columns 2 & 3 correspond to lower and upper endpoints for lOR estimate.

#Libraries
library(randomForest)
library(foreach)
library(doParallel)

RF2PM.CI_final <- function(dataset, n.tree, m.try, test.main){

  #Set up dataframe
  n.col <- ncol(dataset)
  data_mat <- data.frame(predictors=dataset[, 1:(n.col-1)], response=dataset[, n.col])

  #Set nodesize = 5% of dataset
  node.size <- 0.05*nrow(data_mat)
```

```r

#Determine number of main effects to estimate
n.main <- length(test.main)

### Main effect Estimation -------------------- --------------------------------
#Use two-machine RFPM method to obtain subject-specific lOR estimates and take mean for overall lOR estimate
main.estimates_binary <- foreach (r=1:n.main, .combine=rbind) %dopar% {

  #Set X_m to desired predictor for main effect estimation
  m = test.main[r]

  #Split dataset into 2 groups G0 and G1 based on whether Xi=0 or Xi=1 and remove Xi
  G_0 <- data_mat[data_mat[,m] == 0, -m]
  G_1 <- data_mat[data_mat[,m] == 1, -m]

  #Train identically specified RFPMs on each group, PM_0 and PM_1
  PM_0 <- randomForest(response ~., data=G_0, ntree=n.tree, mtry=m.try, nodesize=node.size)
  PM_1 <- randomForest(response ~., data=G_1, ntree=n.tree, mtry=m.try, nodesize=node.size)

  #Obtain predictions for observed and counterfactual probabilities of success for each individual
  p_0 <- predict(PM_0, newdata=data_mat)
  p_1 <- predict(PM_1, newdata=data_mat)

  #Calculate subject-specific lOR estimates
  OR.1 <- (p_1*(1-p_0))/(p_0*(1-p_1))
  lOR.1 <- log(OR.1)

  #Calculate sample lOR estimate by taking the mean over all subject-specific estimates
  lOR <- mean(lOR.1)
  lOR}

results.main_binary <- cbind(test.main, main.estimates_binary)
colnames(results.main_binary) <- c("X_j", "lOR"); rownames(results.main_binary) <- NULL

### 95% Confidence Intervals for Main Effect Estimates (b=200) --------------
#Calculate 95% confidence intervals for each of the above log odds ratio estimates
main.estimates_binary.boot <- foreach(y=1:200, .combine=rbind) %dopar% {

  #Create boostrap sample with n=nrow(dataset)
  boot.bin.sample <- sample(nrow(data_mat), size=nrow(data_mat), replace=TRUE)
  sample.bin.b <- data_mat[boot.bin.sample, ]

  #Calculate OR for each predictor for each bootstrap sample
  main.estimate_binary.b <- foreach(z=1:n.main, .combine = cbind) %dopar% {

    #Set X_m to desired predictor for main effect estimation
    m.b = test.main[z]

    #Split dataset into 2 groups G0 and G1 based on whether Xi=0 or Xi=1 and remove Xi
    G_0b <- sample.bin.b[sample.bin.b[,m.b] == 0, -m.b]
    G_1b <- sample.bin.b[sample.bin.b[,m.b] == 1, -m.b]

    #Train identically specified RFPMs on each group, PM_0 and PM_1
    PM_0b <- randomForest(response ~., data=G_0b, ntree=n.tree, mtry=m.try, nodesize=node.size)
    PM_1b <- randomForest(response ~., data=G_1b, ntree=n.tree, mtry=m.try, nodesize=node.size)

    #Obtain predictions for observed and counterfactual probabilities of success for each individual
    p_0b <- predict(PM_0b, newdata=sample.bin.b)
    p_1b <- predict(PM_1b, newdata=sample.bin.b)

    #Calculate subject-specific OR estimates
    OR.1b <- (p_1b*(1-p_0b))/(p_0b*(1-p_1b))
    lOR.1b <- log(OR.1b)

    #Calculate sample OR estimate by taking the mean over all subject-specific estimates
    lORb <- mean(lOR.1b)
    lORb}

  main.estimate_binary.b
}

#Extract 5th and 195th ordered elements to construct 95% confidence interval for each main effect OR estimate
lOR1.sort <- apply(main.estimates_binary.boot, 2, sort)
lOR.main.lower_binary <- lOR1.sort[5,]
lOR.main.upper_binary <- lOR1.sort[195,]
results.main.CI_binary <- cbind(test.main, lOR.main.lower_binary, lOR.main.upper_binary)
colnames(results.main.CI_binary) <- c("X_j", "LB lOR", "UB lOR"); rownames(results.main.CI_binary) <- NULL

### Output ---------------------------------------------- -------------------
return(list('Log odds ratio estimates (binary)'=results.main_binary, '95% Confidence interval for log odds ratio estimates (binary)'=results.main.CI_binary))}
```

R_code/RF2PM.CI_final.R

A.3 Four-machine RFPM method

```
#FUNCTION TO COMPUTE LOG/ODDS RATIO ESTIMATES (OR_11, OR_10, OR_01, ROR, lROR) AND CORRESPONDING 95% CONFIDENCE INTERVALS
#USING DASGUPTA'S FOUR-MACHINE RFPM METHOD IN "Risk Estimation using Probability Machines" (binary predictors)

# *** Runs in parallel using doParallel and foreach ***

#INPUT:
#   dataset: dataframe or matrix where each column represents one predictor and the last column is the response vector
#          (each row corresponds to 1 observation)
#   n.tree: number of trees to grow in RFPM_00, RFPM_01, RFPM_10, RFPM_11
#   m.try: number of predictors randomly sampled as candidates at each split in RFPM_00, RFPM_01, RFPM_10, RFPM_11
#   test.int: matrix with two columns (X_j and X_k) where each row indicates the interaction estimate to be calculated
#   (eg: test.int <- matrix(c(1,2,2,3), nrow=2, byrow=T) indicates interaction estimation for predictors (x_1, x_2)
#   and (x_2, x_3), corresponding to columns (1,2) and (2,3) in dataset)

#OUTPUT:
#   'Interaction Odds Ratio Estimates': four-machine RFPM method used to estimate log/odd ratios for given combinations of
#          binary predictors in test.int.  First two columns are the two predictors involved and remaining 4 columns are
#          means of the following odd ratio estimates:
#          1. OR_11 = p_11(1-p_00)/p_00(1-p_11)
#          2. OR_10 = p_10(1-p_00)/p_00(1-p_10)
#          3. OR_01 = p_01(1-p_00)/p_00(1-p_01)
#          4. ROR = OR_11/OR_10*OR_01
#          5. lROR = log(ROR)
#          *NOTE* Quantity described in Dasgupta is defined as OR_11/OR_10*OR_01
#
#   '95% Confidence interval for interaction odd ratio estimates': four-machine RFPM method and bootstrapping (b=200) used to construct
#          95% confidence intervals for OR_11, OR_10, OR_01, ROR and lROR estimates from 'Interaction Odd Ratio Estimates'.  Output is a
#        list where first component is 95% lower bounds for all estimates and second list component is 95% upper bounds for all estimates
#      (labled accordingly)

```

```
#Libraries
library(randomForest)
library(gtools)
library(foreach)
library(doParallel)

RF4PM.CI_final<- function(dataset, n.tree, m.try, test.int){

  #Set up dataframe
  n.col <- ncol(dataset)
  data_mat <- data.frame(predictors=dataset[, 1:(n.col-1)], response=dataset[, n.col])

  #Set nodesize = 5% of dataset
  node.size <- 0.05*nrow(data_mat)

  ### Interaction Estimation -----------------------------------------------------------------------------
  #Use 4 machine RFPM to estimate following odd ratios:
  #   OR_11 = p_11(1-p_00)/p_00(1-p_11)
  #   OR_10 = p_10(1-p_00)/p_00(1-p_10)
  #   OR_01 = p_01(1-p_00)/p_00(1-p_01)
  #   ROR = OR_11/(OR_10*OR_01)
  #   lROR = log(ROR)

  #Determine number of interactions to estimate
  n.int <- nrow(test.int)

  #For each interaction combination, use four-machine RFPM method to obtain subject-specific estimates and take mean
  #for overall estimate
  interaction.estimates <- foreach (r=1:n.int, .combine=rbind) %dopar% {

    #Set X_m and X_n to desired predictors for interaction estimation
    m = test.int[r,1]
    n = test.int[r,2]

    #Split dataset into 4 groups based on combinations (X_m, X_n)
    G_00 <- data_mat[data_mat[,m] == 0 & data_mat[,n] == 0, c(-m, -n)]
    G_01 <- data_mat[data_mat[,m] == 0 & data_mat[,n] == 1, c(-m, -n)]
    G_10 <- data_mat[data_mat[,m] == 1 & data_mat[,n] == 0, c(-m, -n)]
    G_11 <- data_mat[data_mat[,m] == 1 & data_mat[,n] == 1, c(-m, -n)]

    #Train identically specified RFPMs on each group, PM_00, PM_01, PM_10, PM_11
    PM_00 <- randomForest(response ~., data=G_00, ntree=n.tree, mtry=m.try, nodesize=node.size)
    PM_01 <- randomForest(response ~., data=G_01, ntree=n.tree, mtry=m.try, nodesize=node.size)
    PM_10 <- randomForest(response ~., data=G_10, ntree=n.tree, mtry=m.try, nodesize=node.size)
    PM_11 <- randomForest(response ~., data=G_11, ntree=n.tree, mtry=m.try, nodesize=node.size)

    #Observed probability and counterfactual probabilities for each observation
    p_00 <- predict(PM_00, newdata=data_mat)
    p_01 <- predict(PM_01, newdata=data_mat)
    p_10 <- predict(PM_10, newdata=data_mat)
    p_11 <- predict(PM_11, newdata=data_mat)

    #Calculate odds ratios:
    OR.11 <- (p_11*(1-p_00))/(p_00*(1-p_11))
    OR.10 <- (p_10*(1-p_00))/(p_00*(1-p_10))
    OR.01 <- (p_01*(1-p_00))/(p_00*(1-p_01))
    ROR <- OR.11/(OR.10*OR.01)
    lROR <- log(ROR)

    estimates.combined <- c(mean(OR.11), mean(OR.10), mean(OR.01), mean(ROR), mean(lROR))
    estimates.combined}

  #Bind ORs with test.int for output
  results.interaction <- cbind(test.int, matrix(interaction.estimates, nrow=n.int))
  colnames(results.interaction) <- c("X_j", "X_k", "OR_11", "OR_10", "OR_01", "ROR", "lROR"); rownames(results.interaction) <- NULL

  ### 95% Confidence Intervals for Interaction Estimates (b=200) ----------------------------
  #Calculate 95% confidence intervals for each of the above odds ratio estimates
  interaction.estimates_boot <- foreach(b=1:200, .combine=rbind) %dopar% {

    #Create boostrap sample with n=nrow(data_mat)
    boot.int.sample <- sample(nrow(data_mat), size=nrow(data_mat), replace=TRUE)
    sample.int.b <- data_mat[boot.int.sample, ]

    #Calculate lOR_11, lOR_10, lOR_01 and lROR for each interaction in test.int for each bootstrap sample
    interaction.estimate_b <- foreach(z=1:n.int, .combine = cbind) %dopar% {

      #Set X_m and X_n to desired predictors for interaction estimation
      m.b = test.int[z,1]
      n.b = test.int[z,2]

      #Split dataset into 4 groups based on combinations (X_m, X_n)
      G_00b <- sample.int.b[sample.int.b[,m.b] == 0 & sample.int.b[,n.b] == 0, c(-m.b, -n.b)]
      G_01b <- sample.int.b[sample.int.b[,m.b] == 0 & sample.int.b[,n.b] == 1, c(-m.b, -n.b)]
      G_10b <- sample.int.b[sample.int.b[,m.b] == 1 & sample.int.b[,n.b] == 0, c(-m.b, -n.b)]
      G_11b <- sample.int.b[sample.int.b[,m.b] == 1 & sample.int.b[,n.b] == 1, c(-m.b, -n.b)]

      #Train identically specified RFPMs on each group, PM_00, PM_01, PM_10, PM_11
      PM_00b <- randomForest(response ~., data=G_00b, ntree=n.tree, mtry=m.try, nodesize=node.size)
      PM_01b <- randomForest(response ~., data=G_01b, ntree=n.tree, mtry=m.try, nodesize=node.size)
      PM_10b <- randomForest(response ~., data=G_10b, ntree=n.tree, mtry=m.try, nodesize=node.size)
      PM_11b <- randomForest(response ~., data=G_11b, ntree=n.tree, mtry=m.try, nodesize=node.size)

      #Observed probability and counterfactual probabilities for each observation
      p_00b <- predict(PM_00b, newdata=sample.int.b)
      p_01b <- predict(PM_01b, newdata=sample.int.b)
      p_10b <- predict(PM_10b, newdata=sample.int.b)
      p_11b <- predict(PM_11b, newdata=sample.int.b)

      #Calculate odds ratios:
      OR.11b <- (p_11b*(1-p_00b))/(p_00b*(1-p_11b))
      OR.10b <- (p_10b*(1-p_00b))/(p_00b*(1-p_10b))
      OR.01b <- (p_01b*(1-p_00b))/(p_00b*(1-p_01b))
      RORb <- OR.11b/(OR.10b*OR.01b)
      lRORb <- log(RORb)

      #Calculate odds ratio by taking the mean over all subject specific estimates
      estimates.combinedb <- c(mean(OR.11b), mean(OR.10b), mean(OR.01b), mean(RORb), mean(lRORb))
      estimates.combinedb <- matrix(estimates.combinedb, nrow=1)
      estimates.combinedb}

    interaction.estimate_b}

  if(sum(!is.finite(interaction.estimates_boot)) != 0){
    sum.inf.na.nan <- sum(!is.finite(interaction.estimates_boot))
    print(c("Inf/NA/NaN bootstrap estimates produced:", sum.inf.na.nan,))
    interaction.estimates_boot[which(!is.finite(interaction.estimates_boot))] <- 0
  }

  #Extract the 5th and 195th ordered elements
  int.ORs.sort_binary <- apply(interaction.estimates_boot, 2, sort)
  int.ORs.lower_binary <- matrix(int.ORs.sort_binary[5,], nrow=n.int, byrow=T)
  int.ORs.upper_binary <- matrix(int.ORs.sort_binary[195,], nrow=n.int, byrow=T)

  #Bind with predictor index
  results.int.LCIs_binary <- cbind(test.int, int.ORs.lower_binary)
```

```
  colnames(results.int.LCIs_binary) <- c("X_j", "X_k", "LB OR_11", "LB OR_10", "LB OR_01", "LB ROR", "LB lROR"); rownames(results.int.LCIs_binary) <- NULL
  results.int.UCIs_binary <- cbind(test.int, int.ORs.upper_binary)
  colnames(results.int.UCIs_binary) <- c("X_j", "X_k", "UB OR_11", "UB OR_10", "UB OR_01", "UB ROR", "UB lROR"); rownames(results.int.UCIs_binary) <- NULL

  #Create list containing upper and lower 95% confidence bounds
  results.int.CIs_binary <- list("95% lower bounds"=results.int.LCIs_binary, "95% upper bounds"= results.int.UCIs_binary)

  ### Output -------------------------------------------------------------------
  return(list("Interaction Odd Ratio Estimates"=results.interaction, "95% Confidence intervals for interaction odd ratio estimates"=results.int.CIs_binary))}
```

R_code/RF4PM.CI_final.R

A.4 RFPMs for categorical predictors

```
#FUNCTION TO COMPUTE LOG ODDS RATIO ESTIMATES AND CORRESPONDING 95% CONFIDENCE INTERVALS USING DASGUPTA'S
#RFPM METHODS IN "Risk Estimation using Probability Machines" (categorical predictors)

# *** Runs in parallel using doParallel and foreach ***

#INPUT:
#   dataset: dataframe or matrix where each column represents one predictor and the last column is the response vector
#          (each row corresponds to 1 observation)
#   n.tree: number of trees to grow in the RFPM_1, ..., RFPM_k where k is the number of levels of categorical predictor
#   m.try: number of predictors randomly sampled as candidates at each split in RFPM_1, ..., RFPM_k where k is the
#      number of levels of categorical predictor
#   test.cat: vector where each element indicates the categorical predictor column index. (eg. test.cat = c(2,4), indicates
#         predictors x_2 and x_4 corresponding to columns 2 and 4 in dataset are categorical)

#OUTPUT:
#   'Log odds ratio estimates (categorical)': n.levels RFPMs used to estimate log odds ratios for each level of the
#          categorical predictor relative to the first level.  First column is categorical predictor index and following
#          columns correspond to the (n.levels - 1) estimates.  Each row is one predictor.
#
```

```
#   ’95% Confidence interval for log odds ratio estimates (categorical)’: n.levels RFPMs and bootstrapping (b=200) used to
#          construct 95% confidence intervals for lOR estimates from ’Log ddds Ratio Estimates’.  List object where first
#          component corresponds to lower bounds and second component corresponds to upper bounds.  Formats of individual
#          components analogous to ’Log odds ratio estimates’ result

#Libraries
library(randomForest)
library(foreach)
library(doParallel)

cat.CI_final<- function(dataset, n.tree, m.try, test.cat){

  #Set up dataframe
  n.col <- ncol(dataset)
  data_mat <- data.frame(predictors=dataset[, 1:(n.col-1)], response=dataset[, n.col])

  #Set nodesize = 5% of dataset
  node.size <- 0.05*nrow(data_mat)

  #Determine number of categorical predictors
  n.cat <- length(test.cat)

  #Determine maxmimum number of levels for any categorical predictor
  data_cat <- data_mat[,c(test.cat)]

  if(length(test.cat) == 1){max.levels = length(unique(data_cat))
  } else {unique.levels <- apply(data_cat, 2, unique)
  unique.levels.length <- as.vector(unlist(lapply(unique.levels, length)))
  max.levels <- max(unique.levels.length)}

  ###OR Estimates relative to first level for each categorical predictor -------------
  #Use n.levels RFPMs to obtain (n.levels-1) ORs for each categorical predictor relative to the
  #first level
  level.estimates_categorical <- foreach(c=1:n.cat, .combine=rbind) %dopar% {

    #Extract categorical predictor c
    index.pred.c <- test.cat[c]
    pred.c <- data_mat[,index.pred.c]

    #Determine number of levels
    n.levels <- length(unique(pred.c))

    #Split dataset into groups based on different levels of categorical predictor
    split.c <- split(data_mat, data_mat[, index.pred.c])

    #Create matrix to store all counterfactual and observed probabilities of success for each n.level
    probs.pred.c.level.g <- matrix(NA, nrow=nrow(data_mat), ncol=n.levels)

    #Fit a n.levels RFPMs using observations in each level and predict observed and counterfactual probabilities for
    #each observation in the dataset
    for (g in 1:n.levels){

      #Extract all observations in level_g and remove categorical predictor
      level.g <- split.c[[g]]
      level.g.wo.c <- level.g[,-index.pred.c]

      #Fit a RFPM to the remaining predictors
      PM_g <- randomForest(response ~., data=level.g.wo.c, ntree=n.tree, mtry=m.try, nodesize=node.size)

      #Predict counterfactual and observed probabilities for each observation in the dataset
      p_g <- predict(PM_g, newdata=data_mat)

      #Store in matrix
      probs.pred.c.level.g[,g] <- p_g
    }

    #Compute log odds ratios of each level for each observation and take the mean (using lOR.calc function)
    level.lORs.pred.c <- apply(probs.pred.c.level.g[,-1], 2, lOR.calc, p0.vec=probs.pred.c.level.g[,1])
    level.lORs.pred.c_mat <- matrix(c(level.lORs.pred.c, rep(NA, times=(max.levels-n.levels))), nrow=1)
    level.lORs.pred.c_mat}

  level.estimates_categorical <- matrix(cbind(test.cat,level.estimates_categorical), nrow=n.cat)
  colnames(level.estimates_categorical) <- c("X_j", paste("lOR lvl", seq(2, max.levels)))

  ### 95% Confidence Intervals for level lOR Estimates (b=200) --------------
  #Calculate 95% confidence intervals for each of the above odds ratio estimates
  level.estimates_categorical.boot <- foreach(y=1:200, .combine=rbind) %dopar% {

    #Create boostrap sample with n=nrow(dataset)
    boot.cat.sample <- sample(nrow(data_mat), size=nrow(data_mat), replace=TRUE)
    sample.cat.b <- data_mat[boot.cat.sample, ]

    #Calculate OR for each predictor for each bootstrap sample
    level.estimate_categorical.b <- foreach(cb=1:n.cat, .combine = cbind) %dopar% {

      #Extract categorical predictor c
      index.pred.cb <- test.cat[cb]
      pred.cb <- data_mat[,index.pred.cb]

      #Determine number of levels
      n.levels <- length(unique(pred.cb))

      #Split dataset into groups based on different levels of categorical predictor
      split.cb <- split(data_mat, data_mat[, index.pred.cb])

      #Create matrix to store all counterfactual and observed probabilities of success for each n.level
      probs.pred.cb.level.gb <- matrix(NA, nrow=nrow(data_mat), ncol=n.levels)

      #Fit a n.levels RFPMs using observations in each level and predict observed and counterfactual probabilities for
      #each observation in the dataset
      for (gb in 1:n.levels){

        #Extract all observations in level_gb and remove categorical predictor
        level.gb <- split.cb[[gb]]
        level.gb.wo.cb <- level.gb[,-index.pred.cb]

        #Fit a RFPM to the remaining predictors
        PM_gb <- randomForest(response ~., data=level.gb.wo.cb, ntree=n.tree, mtry=m.try, nodesize=node.size)

        #Predict counterfactual and observed probabilities for each observation in the dataset
        p_gb <- predict(PM_gb, newdata=data_mat)

        #Store in matrix
        probs.pred.cb.level.gb[,gb] <- p_gb
      }

      #Compute log odds ratios of each level for each observation and take the mean (using lOR.calc function)
      level.lORs.pred.cb <- apply(probs.pred.cb.level.gb[,-1], 2, lOR.calc, p0.vec=probs.pred.cb.level.gb[,1])
      level.lORs.pred.cb_mat <- matrix(c(level.lORs.pred.cb, rep(NA, times=(max.levels-n.levels))), nrow=1)
      level.lORs.pred.cb_mat}

    level.estimate_categorical.b}

  #Sort bootstrap estimates for each level of each predictor
  level.lORs.sort_categorical <- apply(level.estimates_categorical.boot, 2, sort)
```

```

  #If NA vector, convert its sorted values to NAs
  for(u in 1:length(level.lORs.sort_categorical)){
    estimate.sort.u <- level.lORs.sort_categorical[[u]]
    if(length(estimate.sort.u) == 0){level.lORs.sort_categorical[[u]] <- rep(NA, 200)}
  }

  level.lORs.sort_categorical <- matrix(unlist(level.lORs.sort_categorical), nrow=200)

  #Extract 5th and 195th ordered elements
  level.lORs.lower_categorical <- matrix(level.lORs.sort_categorical[5,], nrow=n.cat, byrow=T)
  level.lORs.upper_categorical <- matrix(level.lORs.sort_categorical[195,], nrow=n.cat, byrow=T)

  #Bind with predictor index
  results.level.LCIs_categorical <- cbind(test.cat, level.lORs.lower_categorical)
  colnames(results.level.LCIs_categorical) <- c("X_j", paste("LB lOR lvl", seq(2, max.levels))); rownames(results.level.LCIs_categorical) <- NULL
  results.level.UCIs_categorical <- cbind(test.cat, level.lORs.upper_categorical)
  colnames(results.level.UCIs_categorical) <- c("X_j", paste("UB lOR lvl", seq(2, max.levels))); rownames(results.level.UCIs_categorical) <- NULL

  #Create list containing upper and lower 95% confidence bounds
  results.level.CIs_categorical <- list("95% lower bounds"=results.level.LCIs_categorical, "95% upper bounds"= results.level.UCIs_categorical)

  ### Output ----------------
  return(list(’Log odds ratio estimates (categorical)’=level.estimates_categorical, ’95% Confidence interval for log odds ratio estimates (categorical)’= results.level.CIs_categorical))}

lOR.calc <- function(pa.vec, p0.vec){

  #Individual odds ratios and log odds ratios
  OR_i <- (pa.vec/(1-pa.vec))/(p0.vec/(1-p0.vec))
  lOR_i <- log(OR_i)

  #Sample log odds ratios
  lOR <- mean(lOR_i)
  return(lOR)}
```

R_code/cat.CI_final.R

A.5 RFPMs for continuous predictors

```
#FUNCTION TO COMPUTE LOG ODDS RATIO ESTIMATES AND CORRESPONDING 95% CONFIDENCE INTERVALS USING DASGUPTA'S
#RFPM METHODS IN "Risk Estimation using Probability Machines" (continuous predictors)

#************ RUNS IN PARALLEL ******************

#INPUT:
#   dataset: dataframe or matrix where each column represents one predictor and the last column is the response vector
#          (each row corresponds to 1 observation)
#   n.tree: number of trees to grow in the RFPM s
#   m.try: number of predictors randomly sampled as candidates at each split in RFPMs
#   n.break: number of bins to split continuous predictor into
#   test.cont: vector where each element indicates the continuous predictor column index that will be split.  (eg:
#    test.cont = c(2,4), indicates predictors x_2 and x_4 corresponding to columns 2 and 4 in dataset are continuous
#    and will be split into n.break bins and (n.break-1) log odds ratios will be estimated)

#OUTPUT:
#   'Log odds ratio estimates (continuous)': n.break RFPMs used to estimate log odds ratios for each bin of continuous predictor
#          relative to first bin (produces n.break-1 lORs).  First column is continuous predictor index and
#          columns correspond to the (n.break-1) lOR bin estimates.  Each row is one predictor
#
#   '95% Confidence interval for log odds ratio estimates (continuous)': n.break RFPMs and bootstrapping (b=200) used to
#          construct 95% confidence intervals for lOR estimates from 'Log odd Ratio Estimates'.  List object where first
#          component corresponds to lower bounds and second component corresponds to upper bounds.  Formats of individual
#          components analogous to Log odds ratio estimates (continuous)' result
#
# 'Continuous predictor bin means': individual n.break bin means of continuous predictor \bar{x}_j for each predictor
#     in test.cont.  First column is test.cont and reamaining n.break columns correspond to bin means
#
# Plots: plot for each continuous predictor will also be produced.  Plotting bin log odds ratio estimates against bin means
#     and corresponding 95% confidence intervals

#Libraries
library(randomForest)
library(foreach)
library(doParallel)
library(ggplot2)

cont.CI_final <- function(dataset, n.tree, m.try, n.break, test.cont){

  #Set up dataframe
  n.col <- ncol(dataset)
  data_mat <- data.frame(predictors=dataset[, 1:(n.col-1)], response=dataset[, n.col])

  #Set nodesize = 5% of dataset
  node.size <- 0.05*nrow(data_mat)

  #Determine number of continuous predictors
  n.cont <- length(test.cont)

  ###Bin OR Estimates for each continuous predictor -------------
  #Use n.break RFPMs to obtain (n.break-1) ORs for each continuous predictor relative to the
  #first bin
  bin.estimates_continuous <- foreach(c=1:n.cont, .combine=rbind) %dopar% {

    #Extract continuous predictor c
    index.pred.c <- test.cont[c]
    pred.c <- data_mat[,index.pred.c]

    #Break continuous predictor into n.break quantiles and split dataset into n.break groups
    #based on category of continuous predictor
    break.c <- quantcut(pred.c, n.break)
    bins.c <- split(data_mat, break.c)

    #Store the mean of the continuous predictor in each bin
    bins.pred.c <- split(pred.c, break.c)
    mean.bins.pred.c <- unlist(lapply(bins.pred.c, mean))

    #Create matrix to store all counterfactual and observed probabilities of success for each n.break
    probs.pred.c.bin.g <- matrix(NA, nrow=nrow(data_mat), ncol=n.break)

    #Fit a n.break RFPMs using observations in each bin and predict observed and counterfactual probabilities for
    #each observation in the dataset
    for (g in 1:n.break){

      #Extract all observations in bin_g and remove continuous predictor
      bin.g <- bins.c[[g]]
      bin.g <- bin.g[,-index.pred.c]

      #Fit a RFPM to the remaining predictors
      PM_g <- randomForest(response ~., data=bin.g, ntree=n.tree, mtry=m.try, nodesize=node.size)

      #Predict counterfactual and observed probabilities for each observation in the dataset
      p_g <- predict(PM_g, newdata=data_mat)

      #Store in matrix
      probs.pred.c.bin.g[,g] <- p_g
    }

    #Compute odds ratios of each bin for each observation and take the mean (using lOR.calc function)
    bin.lORs.pred.c <- apply(probs.pred.c.bin.g[,-1], 2, lOR.calc, p0.vec=probs.pred.c.bin.g[,1])

    #Bind estimates and means of bins
    bin.lORs.N.means.pred.c <- c(bin.lORs.pred.c, mean.bins.pred.c)
    bin.lORs.N.means.pred.c}

  bin.estimates_continuous <- cbind(test.cont, matrix(bin.estimates_continuous, nrow=n.cont))
  colnames(bin.estimates_continuous) <- c("X_j", paste("OR Bin", seq(2, n.break)), paste("Mean bin", seq(1, n.break)))
  rownames(bin.estimates_continuous) <- NULL

  ### 95% Confidence Intervals for Bin OR Estimates (b=200) --------------
  #Calculate 95% confidence intervals for each of the above odds ratio estimates
  bin.estimates_continuous.boot <- foreach(y=1:200, .combine=rbind) %dopar% {

    #Create boostrap sample with n=nrow(dataset)
    boot.cont.sample <- sample(nrow(data_mat), size=nrow(data_mat), replace=TRUE)
    sample.cont.b <- data_mat[boot.cont.sample, ]

    #Calculate OR for each predictor for each bootstrap sample
    bin.estimate_continuous.b <- foreach(cb=1:n.cont, .combine = cbind) %dopar% {

      #Extract continuous predictor c
      index.pred.cb <- test.cont[cb]
      pred.cb <- data_mat[,index.pred.cb]

      #Break continuous predictor into n.break quantiles and split dataset into n.break groups
      #based on category of continuous predictor
      break.cb <- quantcut(pred.cb, n.break)
      bins.cb <- split(data_mat, break.cb)

      #Create matrix to store all counterfactual and observed probabilities of success for each n.break
      probs.pred.cb.bin.gb <- matrix(NA, nrow=nrow(data_mat), ncol=n.break)

      #Fit a n.break RFPMs using observations in each bin and predict observed and counterfactual probabilities for
      #each observation in the dataset
      for (gb in 1:n.break){

        #Extract all observations in bin_g and remove continuous predictor
        bin.gb <- bins.cb[[gb]]
        bin.gb <- bin.gb[,-index.pred.cb]

        #Fit a RFPM to the remaining predictors
        PM_gb <- randomForest(response ~., data=bin.gb, ntree=n.tree, mtry=m.try, nodesize=node.size)

        #Predict counterfactual and observed probabilities for each observation in the dataset
        p_gb <- predict(PM_gb, newdata=data_mat)

        #Store in matrix
        probs.pred.cb.bin.gb[,gb] <- p_gb
      }

      #Compute odds ratios of each bin for each observation and take the mean (using lOR.calc function)
      bin.lORs.pred.cb <- apply(probs.pred.cb.bin.gb[,-1], 2, lOR.calc, p0.vec=probs.pred.cb.bin.gb[,1])
      bin.lORs.pred.cb <- matrix(bin.lORs.pred.cb, nrow=1)
      bin.lORs.pred.cb}

    bin.estimate_continuous.b}

  #Extract the 5th and 195th ordered elements
  bin.lORs.sort_continuous <- apply(bin.estimates_continuous.boot, 2, sort)
  bin.lORs.lower_continuous <- matrix(bin.lORs.sort_continuous[5,], nrow=n.cont, byrow=T)
  bin.lORs.upper_continuous <- matrix(bin.lORs.sort_continuous[195,], nrow=n.cont, byrow=T)

  #Bind with predictor index
  results.bin.LCIs_continuous <- cbind(test.cont, bin.lORs.lower_continuous)
  colnames(results.bin.LCIs_continuous) <- c("X_j", paste("LB lOR Bin", seq(2, n.break))); rownames(results.bin.LCIs_continuous) <- NULL
  results.bin.UCIs_continuous <- cbind(test.cont, bin.lORs.upper_continuous)
  colnames(results.bin.UCIs_continuous) <- c("X_j", paste("UB lOR Bin", seq(2, n.break))); rownames(results.bin.UCIs_continuous) <- NULL

```

```
  #Create list containing upper and lower 95% confidence bounds
  results.bin.CIs_continuous <- list("95% lower bounds"=results.bin.LCIs_continuous, "95% upper bounds"= results.bin.UCIs_continuous)

  #Create bin means
  bin.means_continuous <- bin.estimates_continuous[, c(1, (n.break+1):ncol(bin.estimates_continuous))]

  ### Plotting estimates and confidence intervals for each continuous predictor -----------------------
  #Plot estimates of continuous variables at the point (mean(bin), estimate) with corresponding error bars (95% confidence interval)
  for (p in 1:n.cont){

    #Extract bin lOR estimates corresponding to continuous predictor p
    bin.lOR.pred.p <- bin.estimates_continuous[p,2:n.break]
    bin.means.pred.p <- bin.estimates_continuous[p, (n.break+1):ncol(bin.estimates_continuous)]
    pred.p_continuous <- bin.estimates_continuous[p,1]

    bin.lORs.means_mat <- cbind(bin.lOR.pred.p, bin.means.pred.p[-1])
    colnames(bin.lORs.means_mat) <- c("lOR", "Mean")

    #Plot estimates over means
    plot <- ggplot(as.data.frame(bin.lORs.means_mat), aes(x=Mean, y=lOR)) + geom_point(shape=19) + scale_x_continuous(paste("Values of predictor X_", pred.p_continuous)) + scale_y_continuous(name="log odds ratio") + ggtitle(paste("Bin log odds ratio estimates and corresponding 95% confidence intervals (X_", pred.p_continuous, ")")) + theme_bw() + theme(legend.position="none", plot.title = element_text(hjust=0.5, size=15), axis.title = element_text(size=12), axis.text= element_text(size=10)) + geom_errorbar(aes(x=bin.means.pred.p[-1], ymin=bin.lORs.lower_continuous[p,], ymax=bin.lORs.upper_continuous[p,]), width=0.1)
    print(plot)}

  ### Output ----------------
  return(list('Log odds ratio estimates (continuous)'=bin.estimates_continuous[,-c((n.break+1):ncol(bin.estimates_continuous))], '95% Confidence interval for log odds ratio estimates (continuous)'= results.bin.CIs_continuous, "Continuous predictor bin means" = bin.means_continuous))}

lOR.calc <- function(pa.vec, p0.vec){

  #Individual odds ratios and log odds ratios
  OR_i <- (pa.vec/(1-pa.vec))/(p0.vec/(1-p0.vec))
  lOR_i <- log(OR_i)

  #Sample log odds ratios
  lOR <- mean(lOR_i)
  return(lOR)}
```

R_code/cont.CI_final.R