PREDICTIVE VARIABLE SELECTION FOR SUBGROUP IDENTIFICATION

A thesis submitted to the University of Manchester
for the degree of Master of Philosophy
in the Faculty of Science and Engineering

2017

Emily Turner
School of Computer Science

Contents

Word count 11,479

List of Tables

List of Figures

The University of Manchester

Emily Turner
Master of Philosophy
Predictive Variable Selection for Subgroup Identification
December 16, 2017

The problem of exploratory subgroup identification can be broken down into three steps. The first step is to identify predictive features, the second is to identify the interesting regions on those features, and the third is to estimate the properties of the subgroup region, such as subgroup size and the predicted recovery outcome for individuals belonging to this subgroup. While most work in this field analyses the full subgroup identification procedure, we provide an in-depth examination of the first step, predictive feature identification. A feature is defined as predictive if it interacts with a treatment to affect the recovery outcome.

We compare three prominent methods for exploratory subgroup identification: Virtual Twins (Foster et al. 2011), SIDES (Subgroup Identification based on Differential Effect Search, Lipkovich et al. 2011) and GUIDE (Generalised, Unbiased Interaction Detection and Estimation, Loh et al. 2015).

First, we provide a theoretical interpretation of the problem of predictive variable selection and connect it with the three methods. We believe that bringing different approaches under a common analytical framework facilitates a clearer comparison of each. We show that Virtual Twins and SIDES select interesting features in a theoretically similar way, so that the essential difference between the two is in the way in which this selection mechanism is implemented in their respective subgroup identification procedures.

Second, we undertake an experimental analysis of the three. In order to do this, we apply each method to return a predictive variable importance measure (PVIMs), which we use to rank features in order of their predictiveness. We then evaluate and compare how well each method performs at this task.

Although each of Virtual Twins, SIDES and GUIDE either output a PVIM or require minor adaptations to do so, their strengths and weaknesses as PVIMs had not been explored prior to this work. We argue that a variable ranking approach is a particularly good solution to the problem of subgroup identification. Because clinical trials often lack the power to identify predictive features with statistical significance, predictive variable scoring and ranking may be more appropriate than a full subgroup identification procedure. PVIMs enable a clinician to visualise the relative importance of each feature in a straightforward manner and to use clinical expertise to scrutinise the findings of the algorithm.

Our conclusions are that Virtual Twins performs best in terms of predictive feature selection, outperforming SIDES and GUIDE on every type of data set. However, it appears to have weaknesses in distinguishing between predictive and prognostic biomarkers.

Finally, we note that there is a need to provide common data sets on which new methods can be evaluated. We show that there is a tendency towards testing new subgroup identification methods on data sets that demonstrate the strengths of the algorithm and hide its weaknesses.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and she has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Policy on Presentation of Theses.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Gavin Brown, for expert input on the substance and form of the dissertation. He gave me a lot of his time as we edited the work together, and through this, he helped me to develop a keener eye for structure and style, and a critical eye for well-focused research. I also thank him for the kind, helpful support he gave me as I was making the decision to convert from the PhD to the MPhil.

I would also like to thank my co-supervisor, Prof. Robert Stevens, and Prof. Andy Brass for providing feedback on the my progress of my work and the direction of my research at key stages in the degree.

To Dr. Kostas Sechidis, I would like to say thank you for invaluable guidance, and for kindly answering the many questions I threw at him at all sorts of convenient and inconvenient times. His breadth and depth of knowledge around this topic made for enlightening and clarifying conversations about the work.

I want to thank my friends for their support and encouragement as I was writing up. This includes my research group: Georgiana Neculae, Nikos Nikolaou, Henry Reeve, Kostas Papangelou, Tameem Hesham, and Sarah Nogueira; the Lunch Gang: Cameron Shand, Danny Wood, Rob James, Amy Pollard, Lara Mihaela and Alex Bogatu; Hannah Lowry OReilly and Lianora Bermingham, who may both be in Ireland, but are nevertheless a constant WhatsApp presence. To Dr. Javier Caballero, thank you for the proof reading and generous support as I worked, and for devising fun excursions and distractions in my time off :)

To my sister, Lisa Turner, and my parents, Margaret and Eamonn Turner, I owe the biggest thanks of all. Thank you for the encouragement you have always provided. I would not be where I am today without it.

Chapter 1

Introduction

1.1 Background

Different patients will respond to a treatment in different ways. Factors to do with an individual's lifestyle, history and genetic make-up can all affect how well a patient will receive the treatment [1]. For example, the presence of a certain genetic mutation could mean that a medicine is metabolised particularly well. In this example, a *biomarker* is a binary variable indicating whether the mutation is present or not. In general, a biomarker is a molecule, gene, or characteristic by which a physiological process can be identified [2]. A subgroup is then defined by the region on the variable that corresponds to the presence of the mutation, so that a patient falls into the subgroup if they have the mutation that makes them particularly responsive to the treatment.

The process of subgroup identification occurs in three main steps:

1. identify features that interact with the treatment to affect recovery outcome,
2. locate the relevant region on the feature, and
3. estimate properties of the subgroup region.

The first step is the focus of this study. If suboptimal features are found at this point, true subgroups may be overlooked or poorly defined. Therefore, it is important to understand how a subgroup identification algorithm performs in this task. The second step is to identify the values of the feature that correspond to a better (or worse) recovery outcome. In the example of the previous paragraph, the region of interest on

the binary feature is the indication that the mutation is present as this is associated with a strong, positive response to the treatment. This step is often combined with the first so that a feature is selected because the best split on that feature yields a high importance score. The third and final step is to estimate properties, such as the number of patients that we expect to have the characteristics of the subgroup and the expected treatment effect for those patients, where *treatment effect* is the extent to which receiving the treatment improves recovery prospects. Additional metrics calculated at this stage can include standard deviation and bias of the estimated treatment effect.

We are looking for *predictive* as opposed to *prognostic* biomarkers. Predictive biomarkers are associated with a large treatment effect. Prognostic biomarkers are associated with enhanced recovery prospects regardless of whether the treatment is received or not [3]. In other words, predictive biomarkers are important for personalised medicine, while prognostic biomarkers provide insights for general patient care.

The nature of the search is *exploratory*, which means we are *generating* hypotheses about predictive biomarkers [4], not *testing* hypotheses that have already been derived. The latter is *confirmatory* subgroup analysis and its purpose is to verify the existence of a small group of well-defined subgroups [5]; the important features are already known and subgroups are defined before looking at the data. It is not the focus of this study. In exploratory subgroup identification, we don't predefine the subgroups, because we do not yet know what they are and they are what we are looking for. Instead, we predefine the method that will be used to search for them. Following the same justification for requiring a *pre*definition of subgroups in the confirmatory process, the predefinition of the method in the exploratory process mitigates against a biased analysis of the data [6].

Exploratory subgroup analysis is commonly used in Phase III clinical trials [7], but can be applied in Phase IV too [8]. Both of these stages of clinical trials are larger than Phase I and II trials, the safety of the new treatment having been established in the earlier phases. Phase III trials are carried out to compare a new drug against the best currently available treatment, and Phase IV is for learning more about the side effects and long-term risks associated with it [1]. Any compelling discoveries made during exploratory analysis are examined by clinicians for their plausibility. They will need

to be verified in the more statistically rigorous setting of confirmatory analysis where the discovered subgroups become the well-defined subgroups that are to be tested.

This is a feature selection problem in so far as we wish to find a small subset of important features in a potentially large feature set. However, it can also be descried as an interaction detection problem, because a feature is defined as important if the influence of both the feature and the treatment on the outcome is not additive but, rather, the combination of the treatment with the patient characteristic is required for there to be a differential treatment effect.

In this study we examine three methods of subgroup identification: the Virtual Twins [9], SIDES [10], and GUIDE [11] methods. Each uses recursive partitioning to model the data. The hierarchical tree structure of recursive partitioning, or, in other words, models that are variations on the decision tree, are well suited to modelling higher-order interactions because the process by which they add nodes to a tree implicitly searches for interactions between features without first requiring that these interactions are specified, as is the case in, say, linear or logistic regression models [6]. We evaluate and compare them in terms of their power to identify predictive features. In Chapter 2 each of the methods is outlined in full.

We apply the subgroup identification methods to return predictive variable importance measures (PVIMs), which are used to rank features in terms of their predictiveness. By shifting from a binary feature selection procedure to a ranking system, we can then chooose a threshold on the PVIM with reference to clinical, practical or business considerations. The relative importance of features in a data set can also be visualised in a way that is easy for clinicians to interpret. We cover the ways in which each method returns an importance score in a second literature review of Section 2.2.

In Chapter 3, we analyse each method from a theoretical standpoint, looking past the particularities of each method to focus on the way by which each makes the biomarker selection. Using a single theoretical framework, we compare how each identifies predictive biomarkers and distinguishes them from prognostic biomarkers. Then, in Chapter 4, we compare the performance of each method experimentally, and examine how the theoretical conclusions play out in practice. Final discussions and conclusions are provided in Chapter 5.

In each of the three works, [9], [10], [11], the authors propose a synthetic data

set on which they test their method. The data sets are described in Section 4.1. We perform a *cross-comparison* to assess the performance of each of the three methods on each of the three synthetic data sets.

1.2 Contributions

- We re-express the three methods in a common analytical framework, which permits a better understanding of how each method functions and facilitates a comparison of the three. This approach brings additional clarity to our understanding of the mechanism by which each method is picking out predictive features.

- We apply the Virtual Twins, SIDES and GUIDE methods to return predictive feature importance scores and analyse the resultant feature rankings. Although each method can be used as a PVIM, none had been evaluated in this capacity before.

- We show that the Virtual Twins method outperforms SIDES and GUIDE in identifying predictive features. However, it is weaker than the other two in distinguishing predictive from prognostic features, consistently scoring prognostic features higher than irrelevant features. This indicates that the Virtual Twins PVIM is not solely a detector of predictiveness and is also picking up prognostiveness.

- We show that there is a need to provide data sets on which all subgroup identification methods can be evaluated and benchmarked. We identify a trend of bias in the choice of model on which a new method is evaluated. From our study of Virtual Twins, SIDES and GUIDE, we notice that the data model that was selected demonstrates each method in the best light, to the detriment of providing insight on the weaknesses of each method.

Chapter 2

Literature Review

2.1 Subgroup Identification

The problem of subgroup identification has been approached in diverse ways. We give a brief overview of the variety of approaches that have been used, and then describe the Virtual Twins, SIDES and GUIDE methods in detail. In particular, we focus on the mechanism by which each method identifies predictive features. We will show that predictive feature identification is essentially a problem of feature selection based on interaction detection. We focus on binary and continuous response variables in this work, although there are also procedures for identifying subgroups with longitudinal and censored data, for example censored survival data [12], [13] and longitudinal and multiresponse features [14].

To take a traditional, parametric statistical approach to subgroup identification, a global regression model is fitted to the data. Then, in order to test a variable-treatment interaction, it must be explicitly specified in the model along with the main effects of the treatment and the variable in question. Pocock [15] notes that these tests are likely to be underpowered because most clinical trials tend to only have sufficient statistical power to detect main effects alone. Another drawback of having to specify each interaction to be tested is that in data sets with many features a large number of interaction terms will need to be tested. Multiple significance tests increases the risk of *multiplicity*: the more interaction terms that are tested, the greater the probability that one term will look significant *by chance*. One way to reduce the risk of false positives is to use a family-wise error rate control such as the Bonferroni correction. However, this

is not ideal. Although this reduces the risk of false positives, it necessarily increases the risk of false negatives. In this setting, penalisation methods such as LASSO [16] and elastic net [17] can be employed to handle the unwieldy number of features that may be present in a data set.

A Bayesian model averaging approach is proposed by [18] as an alternative means of reducing the burden of multiplicity by assigning prior probabilities to the models. It is possible to incorporate expert knowledge into the modelling process via the priors with this approach. A drawback of this approach is that it is computationally intense to construct all possible models in the model space for large data sets. This is further complicated when dealing with continuous data - only binary predictors were considered in [18]. Berger uses tree-based methods to construct main-effects models and subgroup models following the procedure described by Wang [19], limiting tree depth to one split only. Thus, it is a partitioning method, but it is computationally expensive to split recursively for a tree depth greater than one.

Recursive partitioning in general has become a popular approach to detecting subgroups. Tree structured models are well-suited to detecting interactions, including non-linear interactions, between features without first having to be specified. They are nonparametric and can readily handle any variable type.

Interaction Trees [6], implement the CART regression tree with a new splitting criterion. The criterion proposed by Su et al. separates the data based on the split that yields the greatest heterogeneity in treatment effect. An integration of parametric models within tree-structured models is developed in the Model-based Recursive Partitioning method [20]. At each node in a tree, the model must be specified (eg: a logistic or linear regression model) along with the variables to be used in the partitioning. The QUalitative INteraction Trees (QUINT) method [21] is motivated by the idea that it is important to identify patients for which treatment A is better than B, those for which B is better than A, and those for which there is no great difference in outcome regardless of whether they receive A or B. The authors use a splitting criterion that maximises the conjunction of two criteria. The first criterion is that in the two subgroups where one treatment is more effective than the other, the difference in outcome between treatments A and B is maximised. The second criterion is that

membership size in both of these groups should be maximised. This method has application in an area that is closely related to subgroup identification, namely, optimal treatment allocation.

The three methods that are the focus of this study are Virtual Twins [9], SIDES [22] and GUIDE [11]. They are chosen because they are prominent, recently proposed methods, and because they have a diversity strengths and weaknesses in their approaches which can be compared and contrasted. Each method is fully detailed in the next three sections.

The notation that will be used is outlined now. Let N be the total number of patients in a clinical trial and $X = X^1, ..., X^m$ be the measurements that have been recorded for each patient. These are potential 'biomarkers' or, in statistical terminology, they are the 'features' or 'variables' that will be used to model the outcome. The recovery outcome, Y, may be binary or continuous. The treatment indicator T indicates whether the new treatment was received, $T = 1$, or not, $T = 0$. The treatment effect, Z, is the difference in recovery outcome when the treatment is received versus when it is not received. We are looking for regions of the biomarker space, $S_x \in X$, where there is a large treatment effect.

We use a toy data set to illustrate the differing ways in which Virtual Twins, SIDES and GUIDE operate when selecting a predictive feature. The simulation is based on the example used by Loh et. al [11].

$$Y = 1.9 - 1.8I(X_1 > 0) + 0.2T + 3.6TI(X_1 > 0) + \epsilon, \tag{2.1}$$

where $I(X_1 > 0)$ is the indicator function that takes the value 1 if $x_1 > 0$ and 0 otherwise. There are twenty features in total, $X = X_1, ..., X_{20}$, each following a normal distribution, $N(0, 1)$. The Gaussian noise in the data set is represented by ϵ. There is one predictive feature, X_1, and the rest are irrelevant. The subgroup signal in this example is strong for illustrative purposes. The methods are demonstrated in the best-case scenario. In reality, and in the data sets that are used for the cross-comparison of Chapter 4, the subgroup signal is weaker and disguised by a larger amount of noise.

2.1.1 Virtual Twins

The Virtual Twins method, as outlined by the authors, proceeds in three main steps. First, the treatment effect for each individual is estimated. Second, the data subspaces associated with large treatment effect are identified. At this point, the first and second steps of subgroup identification, as outlined in the introduction, are performed, so that both the predictive features and the interesting regions on those features are selected. Third, in subgroups that look promising, the treatment effect is evaluated and subgroup size estimated. The details of the Virtual Twins method [9] are now set out.

In calculating the individual treatment effect z_i, the authors focus on a binary outcome and calculate z_i as the difference in recovery prospects for individual i with biomarker characteristics $X_i = x_i^1, ..., x_i^m$, given they receive the new treatment versus their prospects of recovery given they receive the control treatment instead,

$$z_i = \hat{p}_{1X_i}(y_i = 1) - \hat{p}_{0X_i}(y_i = 1), \tag{2.2}$$

where $\hat{p}_{1X_i}(y_i = 1)$ is the estimated probability of recovery given that the new treatment is received and the patient has characteristics X_i. Similarly, $\hat{p}_{0X_i}(y_i = 1)$ is the probability of recovery given the reference treatment is received. We only observe the outcome for individual i under one treatment setting, because a patient can receive either the new or reference treatment, but not both. The Virtual Twins method overcomes this problem using counterfactual modelling. So if, say, a patient receives the new treatment, $T = 1$, then $p_{1X_i}(y_i = 1)$ is *estimated* from the data, while $p_{0X_i}(y_i = 1)$ is the counterfactual outcome that is *predicted* from the data. The authors use a random forest for this step.

If, the outcome is not binary but, rather, on the continuous scale, for example, if we are measuring the impact of a new treatment on insulin levels where the outcome is measured in microunits per millilitre, then individual treatment effect is

$$z_i = y_{1i} - y_{0i},$$

the difference between the outcome when the new treatment is received, y_{0i}, and the outcome when the reference treatment or placebo is received, y_{0i}.

The next step is to determine whether any of the features, $X = X^1, ..., X^m$, are

predictive of heterogeneities in the newly created latent feature, Z, where such heterogeneities exist. A decision tree is used for this step. The tree is grown with two stopping conditions, the minimal terminal node size and complexity parameter. The minimal node size parameter ensures that there are enough observations to calculate reliable statistics and also ensures that we only look for subgroups that are big enough to be worthwhile from a commercial perspective. The complexity parameter ensures that a new split on a branch is only added to the tree if the decrease in model lack-of-fit by doing so is enough of an improvement on the previous split on that branch.

If the predicted treatment effect, $\hat{Z}$, in a terminal node of a fully grown tree is greater than some threshold c, then it is determined to be an interesting subgroup S_x^*:

$$S_x^* = \{S_x : \hat{Z}_{terminalnode} > c\}$$

Threshold c is a hyperparameter that needs to be specified. The authors suggest setting it as $c = \delta + 0.05$ or $c = \delta + 0.1$, where δ is the overall treatment effect. Thus terminal nodes are selected where the predicted treatment effect exceeds the overall treatment effect by either 0.05 or 0.1 The predictive features are then identified as the variables that define the path down the tree to the terminal nodes corresponding to the interesting subgroups.

We use the toy data described by Equation 2.1 to illustrate the workings of the Virtual Twins method. In Figure 2.1, plot a), we plot predictive feature X_1 against the recovery outcome Y. There is a clear difference in treatment outcome depending on whether a patient has $X_1 > 0$ or not. A random forest is used to model the counterfactual outcome and then the individual treatment effect Z_i is calculated for each patient. The resultant Z_i is significantly different for patients having $X_1 > 0$ as can be seen from plot b). The Virtual Twins algorithm identifies this interesting region by minimising the mean squared error in two child nodes that are created by splitting on X_1, which would be somewhere close to 0 in this clearcut example. As a comparison, similar plots are also provided for an irrelevant feature, X_2, in c) and d) of Figure 2.1. We can see that the individual treatment effect is distributed randomly with respect to X_2 in plot d).

Once the potentially important subgroups have been identified, the properties of each are estimated, including the *differential treatment effect*, $Q(S_x^*)$, which is the

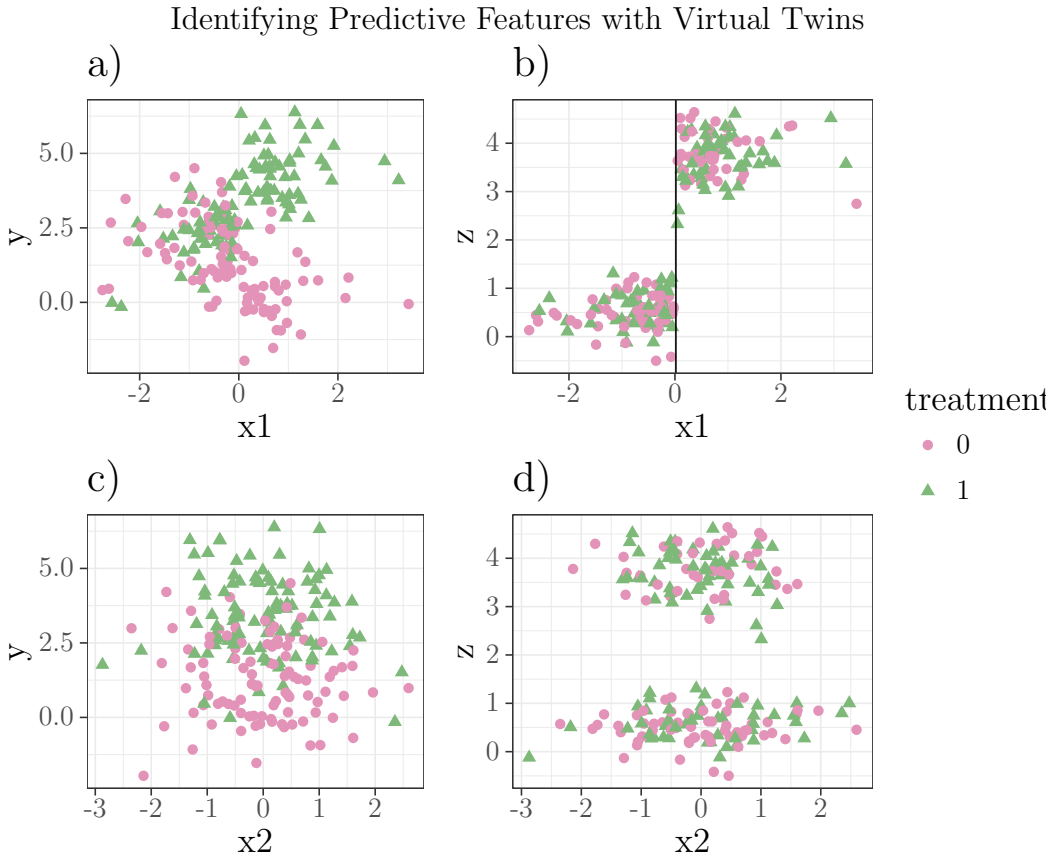

Figure 2.1: a) plots the outcome Y against the predictive feature X_1 of Equation 2.1; b) plots Z against X_1, where Z is the latent feature that has been induced by taking the difference between the counterfactual outcomes (the virtual twins) for each individual.

magnitude of the difference in treatment effect between the subgroup and the full group

$$Q(S_x^*) = \underbrace{\left[p_{1S_x^*}(y_i = 1) - p_{0S_x^*}(y_i = 1)\right]}_{\text{subgroup } S^* \text{ treatment effect}} - \underbrace{\left[p_1(y_i = 1) - p_0(y_i = 1)\right]}_{\text{overall treatment effect}}. \tag{2.3}$$

The number of individuals falling into the subgroups and therefore experiencing an enhanced treatment effect is also estimated. A bootstrap bias corrected approach was observed to yield the most accurate results [9]. These properties are only estimated for the subgroups of interest and not the whole sample. It is in this sense that the Virtual Twins method can be considered a local modelling method.

For the variable importance score used in this study, we implement the first step in its original form and use a random forest instead of a decision tree in the second step, in order to obtain a PVIM. This is outlined fully in Subsection 2.2.1 on the Virtual Twins predictive variable importance score.

2.1.2 SIDES

SIDES is a recursive partitioning method that also uses a local modelling approach. At each step in the search, it identifies potentially interesting regions of the data, discontinuing the subgroup search in the data falling outside of these regions. SIDES is based on the idea of bump-hunting in high-dimensional data as proposed by Friedman and Fisher [23] and developed by Kehl and Ulm [24]. The bump-hunting method is applicable where we are only interested in local features of the variable space, such as regions with strong treatment effect [23], and we are not interested in modelling the outcome for every individual. The 'uninteresting' subspaces of the data are peeled away, and excluded from the rest of the modelling procedure.

SIDES searches *directly* for predictive biomarkers by choosing the split in the data to maximise the differential treatment effect between the left and the right child nodes. If a feature is categorical, all ways of dividing the data in two must be considered. For example, a categorical feature with three levels, A, B, C, can be grouped in 3 ways: $\{A, BC\}$, $\{AB, C\}$, $\{AC, B\}$. If the feature is continuous, we limit the split search space by choosing ten evenly spaced split-points for evaluation. The authors use a Šidák-based multiplicity adjustment to reduce the selection bias between features that arises from the fact that some features can be split in more ways than others. For

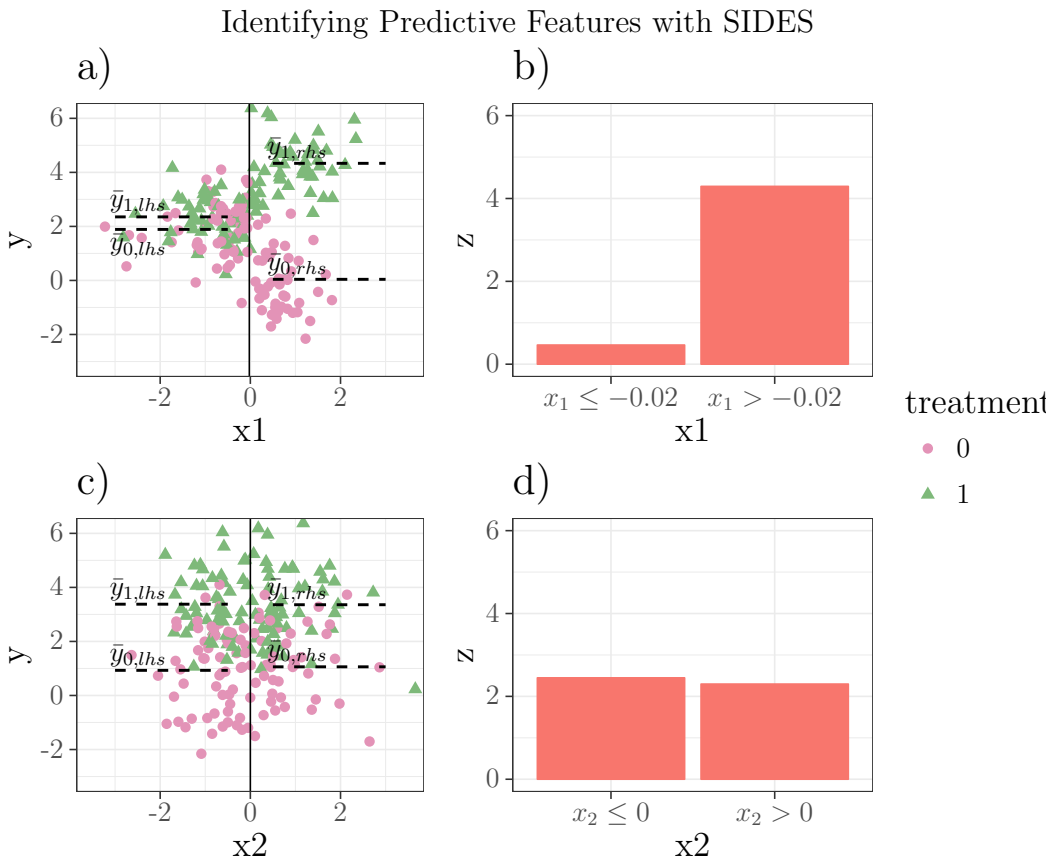

Figure 2.2: a) plots the outcome Y against the predictive feature X_1 of Equation 2.1; b) displays the treatment effect of both child nodes; c) and d) display similar plots for irrelevant feature X_2.

example, a categorical feature taking three or more different values can be split in more ways than a binary variable can. This makes it more likely to be selected for spurious reasons only, because with more ways to split a feature comes an increased risk that a feature will look significant by chance alone. Multiplicity adjustment reduces this false-positive risk.

There are three hyperparameters that need to be set. First, the maximum depth of the branches is set by L. This has the effect of limiting the maximum number of variables that are permitted to define a single subgroup to L. Second, is the minimum subgroup size, S. Similar to the Virtual Twins minimum node size parameter, this can be determined based on clinical trial or marketing considerations. Third, the maximum number of split-feature candidates considered at each node M is limited, so that the algorithm runs more quickly.

A *splitting criterion* is used to select the M best split-feature combinations, subject

to the condition that the size of the better subgroup is greater than S. The authors list three examples of criterion that can be used at this point. We focus on the first criterion [22] because it utilises a search strategy that maximises the differential treatment effect between two child nodes of the parent node k. The splitting criterion is a p-value p_k which represents the difference in test statistics of the left and right arms,

$$p_k = 2\left[1 - \Phi\left(\frac{|t_{kS_x} - t_{kS_x^c}|}{\sqrt{2}}\right)\right]. \tag{2.4}$$

Each test statistic, t_k, is a one-sided significance test for treatment effect, calculated in each of the child node subpopulations represented by S_x and S_{x^c}. The data subsets do not overlap, $S_x \cap S_x^c = \emptyset$, and $S_x \cup S_x^c$ encompasses the full space of a variable. An individual i is allocated to the left arm if the value of biomarker $x_i \in S_x$ and to the right otherwise. The test statistic represents treatment effect magnitude (the larger the treatment effect, the larger the test statistic), so that a significant difference between the statistics of the two child nodes is indicative of a large differential treatment effect. In this way, the SIDES algorithm *directly* searches for biomarkers that are predictive.

The best M split-feature combinations are identified using this criterion. Then, for each of these, the child node data corresponding to the largest test statistic is selected and the split-feature selection process is repeated in this subset unless maximum depth, L, has been reached. If a feature is already used in the definition of the node, it cannot be used again as a splitting variable.

Referring back to the toy example outlined by Equation 2.1, we illustrate how SIDES operates to identify predictive feature X_1 in Figure 2.2. Plot a) reveals that the subgroup experiencing a large differential treatment effect is distinct from the rest of the sample. The data subsets of the child nodes are defined by $S_x : x_1 > -0.02$ and $S_x^c : x_1 \leq -0.02$, where the threshold of -0.02 is chosen because, as said, a split at this point yields the largest splitting criterion p-value. We can see from plot b) that $\bar{y}_{1,rhs} - \bar{y}_{0,rhs}$ is much larger in magnitude than $\bar{y}_{1,lhs} - \bar{y}_{0,lhs}$. Plots c) and d) illustrate that, for irrelevant feature X_2, a similar split on the feature does not yield a subgroup with a large differential treatment effect.

A *continuation criterion* is applied to splits, which compares the treatment effect p-value of the parent group, p_P, with that of the child node, p_C. The child is only kept if it provides enough of an improvement on the parent: $p_C < \gamma p_P$, where $\gamma \in [0, 1]$

defines the extent of improvement (reduction) on the parents p-value that the child node must achieve in order to be retained.

The finalised set of subgroups is defined by the *selection criterion*. The nature of the SIDES subgroup search procedure is such that where multiple subgroups are identified in a data set, they may be overlapping. A candidate subgroup is selected if it has a p-value smaller than α. A resampling-based method is used to set the adjusted significance level α. It controls the false-positive rate in the weak sense, in that it ensures the probability of incorrectly selecting a promising subgroup is no greater than the nominal level, say 0.05, when no differential treatment benefit is present in *any* of the subgroups.

2.1.3 GUIDE

While Virtual Twins and SIDES couple together the variable selection step with the split selection step, choosing the variable if a split on it maximises an objective function, GUIDE [11] separates these out. The splitting variable is chosen first and then best split on that variable is found. The authors show that this prevents selection bias that arises when there are a variety of variable types in a data set. It has the same penalising function as the Šidák-based multiplicity adjustment of the SIDES method.

For each biomarker X, predict Y with a simple linear model, $Y \sim T + X$. Then obtain the residuals from the model, $R = Y - \hat{Y}$. These residuals can be interpreted as the variance in outcome Y that is not explained by the main effects of either the treatment or the biomarker. The authors then binarise the residual values $R_b = \delta(R > 0)$. If a biomarker X is categorical, it is unchanged, and, if it is ordinal, it is binarised as $X_b = \delta(X > \bar{x})$, where $\bar{x}$ is the mean value of the observations for that biomarker.

Then, at each treatment level, the v degrees of freedom (df) chi-square test statistic for testing the independence between the residuals and the biomarker is calculated,

$$W_t(X) = \chi(R_b; X_b|T = t), t \in (0, 1), \tag{2.5}$$

where v df are calculated as $(\mathcal{R}_b - 1)^*(\mathcal{X} - 1)$, $\mathcal{R}$ is the number of levels of the binarised residuals, which will always be two, and $\mathcal{X}$ is the number of levels of feature X. Equation 2.5 is for the case when X is ordinal. When X is binary, we instead calculate $\chi(R_b; X|T)$.

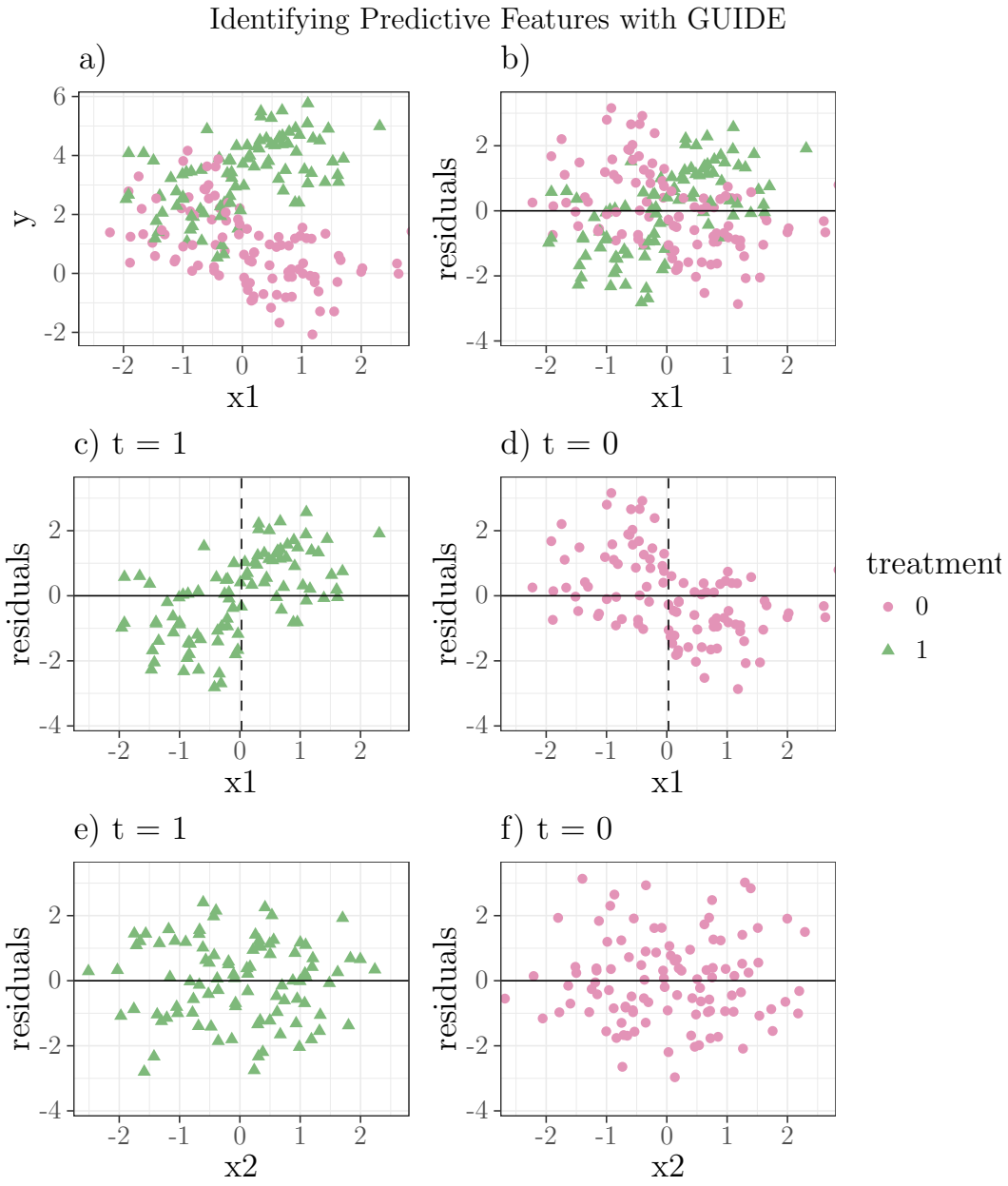

Figure 2.3: a) plots the outcome Y against the predictive feature X_1 of Equation 2.1; b) plots the residuals against X_1; c) and d) split out the points of plot b) according to whether the treatment is received or not; e) and f) plot irrelevant feature X_2 against residuals, split out by treatment level.

To combine the chi-square statistics for a feature across each treatment level, the v-df statistic $W_t(X)$ is first converted to a one-df chi-square quantile using the Wilson-Hilferty approximation [25]:

$$y \approx \mu\Big[1 - 2/(9\mu) + \sqrt{(}v/\mu)\big\{(x/v)^{1/3} - 1 + 2/(9v)\big\}\Big]^3. \tag{2.6}$$

Here, x with v-df is converted into a μ-df quantile y. We set $\mu = 1$ to achieve the one-df chi-square quantile, so we calculate

$$r_t(X) = max\bigg(0, \Big[7/9 + \sqrt{(}v_t)\big\{(W_t(X)/v_t)^{1/3} - 1 + 2/(9v_t)\big\}\Big]^3\bigg). \tag{2.7}$$

Then we sum over the treatment levels, $\sum_{t=0}^{1} r_t(X)$. This sum is a chi-squared variable with L-df, where L is 2 in our case. We use the Wilson-Hilferty approximation a second time to convert it to a one-df chi-square quantile

$$q(X) = max\bigg((0, \Big[7/9 + \sqrt{(}L)\big\{(L^{-1}\sum_{t=0}^{1} r_t(X))^{1/3} - 1 + 2/(9L)\big\}\Big]^3\bigg) \tag{2.8}$$

This provides a ranking of the biomarkers in terms of their relative predictiveness. However, this only works when Y is continuous. If Y is a binary outcome, we use logistic regression instead and calculate the chi-square test statistic via the likelihood ratio test, comparing the models with and without the interaction term, $Y \sim T + X + TX$ and $Y \sim T + X$ respectively.

The next step is to select the feature with the largest chi-square test statistic and search for the best split on it. If X^* is ordinal, choose the split on X^* that minimises the sum of squared residuals in the model $Y \sim T$. If X^* is categorical search over all possible splits in the same way that we do for SIDES.

We demonstrate these principles by applying the GUIDE method to the toy data set. The results are plotted in Figure 2.3. After modelling the main effects of predictive feature X_1, the residuals are *not* distributed randomly with respect to X_1 and T, as can be observed in plots c) and d), indicting that there is an interaction between X_1 and T that is affecting the outcome. This will yield a large chi-square statistic. Plotting the irrelevant feature of the toy data set, X_2, against residuals in plots e) and f), we can see that there is no discernible pattern in the distribution of the residuals with respect to either X_2 or T. The chi-square statistic of this feature will be lower than that of feature X_1. Feature X_1 would be selected and the next step is to find the best split on it.

The data is recursively split in this way until some stopping conditions are met. The authors do not specify these conditions but maximum tree depth, minimal terminal node size or chi-square p-value threshold settings could be used. In [26] the maximum depth is set to 4.

Treatment means and differences are estimated using bootstrap confidence intervals. The naive point and interval estimates are calculated from the training data and then the bootstrap method is applied to estimate standard deviations on these point estimates.

2.2 Predictive Variable Importance Scores

The Virtual Twins, SIDES and GUIDE methods are proposed as full subgroup identification procedures in so far as they provide solutions for the three steps as outlined in section 1.1. The first step is to identify predictive features, the second is to find the interesting regions on those features and the third is to estimate the characteristics of the subgroup. For this work, we have used them as predictive feature selection methods, making minor adaptations where necessary. In this section, the predictive variable importance measures (PVIMs) that are used in the empirical analysis of Chapter 4 are outlined and discussed. A PVIM should identify predictive features and it should not differentiate between prognostic or irrelevant features.

2.2.1 Virtual Twins

The original Virtual Twins method estimates individual treatment effect by modelling counterfactuals using random forests and then using a decision tree to identify the subgroups. To obtain a PVIM, the decision tree is replaced by a second random forest. The random forest will output a variable importance measure for each feature $X = X^1, ..., X^m$ that can be used to rank the features in terms of their subgroup predictiveness.

Every tree in the forest is built using a bootstrap sample of the data and the variable importance score is computed using the out-of-bag (OOB) data. The OOB error is recorded for each tree. As treatment effect Z is continuous, the error is calculated as the mean squared error in the prediction of Z. Then, one at a time, each feature is

permuted and the OOB error recalculated. The difference between the error rate of the non-permuted data and that of the data with permuted feature X is an indication of the extent to which X is important for predicting Z. This is done for each feature in every tree of the random forest, and the average score of each feature across all trees is taken, normalised by the standard deviation of the differences. So a predictive feature should have a PVIM value, whereas prognostic and irrelevant features should have lower values.

2.2.2 SIDES

The authors provide a variable importance score as part of the SIDESscreen method [10]. To calculate the score, grow the SIDES branches by recursively partitioning the data, as specified by the original SIDES algorithm [22]. The output is a selection of branches, each of which represents a subspace of the data that has a large predicted treatment effect.

The more important a feature, the more final subgroups we expect it to appear in. Another indicator of importance is the splitting criterion p-value, D. A smaller p-value is indicative of a larger differential treatment effect. Both of these factors are accounted for by finding the average contribution of each variable across the total number of final subgroups. For each of the final subgroups, if variable X_j appears in a subgroup, then its contribution for that branch is $-log(D_j)$. If it does not appear, its contribution to that subgroup is set to 0.

In the current study we use bagging with the variable importance scoring algorithm. For each bootstrap, the SIDES model is built on the bootstrap sample and the variable importance score is then calculated from the OOB sample, much like the process that is used to calculate the variable importance score in the Virtual Twins method. This reduces the false positive rate. An additional benefit is that this approach aligns more closely to the variable importance score calculation of Virtual Twins, making the important differences between the two methods more easily comparable.

2.2.3 GUIDE

A variable importance score for the GUIDE method is described in [11] with further details provided in [26]. The recursive model is built by following the process that is described in Subsection 2.1.3. At each node k of the fully grown tree we then calculate $q(X)$ for each feature. The variable importance measure (VIM) for feature X^* is then calculated by summing up the $q(X^*)$ scores at each of the nodes weighted by the node size:

$$V_G(X^*) = \sum_{k=1}^{K} n_k \chi^2_{1,k}(X^*) \tag{2.9}$$

Note that here k represents the node, not a non-centrality parameter of a chi-square distribution. In this study we take a simpler approach and rank the features based on one-df chi-square quantiles on the full sample only. We do not use recursive partitioning. In experiments, this approach worked as well for the purpose of ranking predictive features. This aligns with the observations of Hooker [27] that a d-way interaction need only be checked for if all the $(d-1)$-interactions of which it is comprised are significant.

GUIDE could be improved by allowing for non-linear main effects. However, the data sets on which it will be tested have linear main effects, so it suffices to use the simple approach of modelling the main effects as linear for now. The results can be interpreted as being attained in the optimum setting as far as the main effects are concerned.

2.3 Subgroup Identification Algorithms

Virtual Twins, Subsection 2.1.1

1. Estimate the true and counterfactual outcomes for each individual under the treatment received and the alternative treatment respectively (random forest).
2. Calculate individual treatment effect Z as the difference between the outcome when the treatment is received and when it is not received.
3. Predict Z using the X features (decision tree) and select terminal with a prediction $\hat{Z} > c$.
4. Estimate the differential treatment effect and subgroup membership in the selected subgroups.

SIDES, Subsection 2.1.2

1. Select the best M split-feature combinations with splitting criterion, Eq. 2.4.
2. For each split-feature combination, select the child node data with the larger treatment effect test statistic.
3. Recursively split the data in this way until max depth of branch is reached or another split does not provide sufficient improvement on the p-value of the parent node.
4. The final set of subgroups is identified by the selection criterion. A candidate subgroup is selected in this way if $p < \alpha$ where α is set using a resampling-based method.

GUIDE, Subsection 2.1.3

1. For each feature X_i, $i \in 1, ..., m$, model the main effects of X_i and T on Y. Extract the residuals from this model and binarise as $r_b = 1$ if $r > 0$ else 0.
2. Calculate the one-df chi-square statistic testing for independence between X_i and R_b with Eqs. 2.7 and 2.8.
3. Select feature with the largest chi-squared statistic and find the best split on it that minimises the sum of squared residuals of the model $Y \sim T$.
4. Repeat the process in each child node, recursively splitting the data until stopping conditions are met.
5. Estimate subgroup treatment effect using bootstrap confidence intervals.

2.4 PVIM Algorithms

Virtual Twins, Subsection 2.2.1

1. Estimate the true and counterfactual outcomes for each individual under the treatment received and the alternative treatment respectively (random forest).
2. Calculate individual treatment effect Z as the difference between the outcome when the treatment is received and when it is not received.
3. Predict Z using the X features (random forest) and extract random forest variable importance measure as the Virtual Twins PVIM.

SIDES, Subsection 2.2.2

1. Select the best M split-feature combinations by using the splitting criterion of Eq. 2.4.
2. For each split-feature combination, select the child node data with the larger treatment effect test statistic.
3. Recursively split the data in this way until max depth of branch is reached or another split does not provide sufficient improvement on the p-value of the parent node.
4. Once all stopping conditions are met, calculate the SIDES PVIM for each feature as the average contribution of that variable across the total number of final subgroups. If variable X_j appears in a subgroup, then its contribution for that branch is $-log(D_j)$, where D_j is the p-value of that feature for that branch.

GUIDE, Subsection 2.2.3

1. For each feature X_i, $i \in 1, ..., m$, model the main effects of X_i and T on Y. Extract the residuals from this model and binarise as $r_b = 1$ if $r > 0$ else 0.
2. Calculate the one-df chi-square statistic testing for independence between X_i and R_b with Eqs. 2.7 and 2.8. This statistic is the GUIDE PVIM.

Chapter 3

Theoretical Analysis

In Subsections 2.1.1, 2.1.2, and 2.1.3 of the previous chapter, it was shown that Virtual Twins, SIDES, and GUIDE are algorithmically complex, making it difficult to compare them. We strip back the modelling idiosyncrasies of each method to isolate the particular mechanism by which each identifies predictive features. Our strategy is to represent the key dependency measures employed by each in abstract terms in order to understand the fundamental similarities and differences between the methods. We choose notation D to represent the conditional dependencies used by each method in detecting predictive biomarkers. For example, we express the conditional relationship between A and B given C as $D(A; B|C)$.

3.1 Subgroup Identification Methods

3.1.1 Virtual Twins

In Subsection 2.1.1 we saw that the treatment effect for individual i having feature characteristics $X_i = x_i^1, ..., x_i^m$ is described as $z_i = p_{1X_i}(y_i = 1) - p_{0X_i}(y_i = 1)$, where $p_{1X_i}(y_i = 1)$ is the probability of recovery given that the new treatment is received and $p_{0X_i}(y_i = 1)$ is the probability of recovery given that the alternative treatment is received. One of these probabilities is estimated from the data and the other must be predicted. If, say, a patient receives the new treatment, $T_i = 1$, we estimate $\hat{p}_{1X_i}$; and then we flip the treatment indicator and predict the counterfactual outcome, $\tilde{p}_{0X_i}$, for patient i from the same model that is used to estimate $\hat{p}_{1X_i}$.

Alternatively, treatment effect can be expressed on the logit scale instead of the probability scale. This can be arranged as the log-odds ratio:

$$\begin{aligned} z_i &= \text{logit}\big(\hat{p}_{1X_i}(y_i = 1)\big) - \text{logit}\big(\hat{p}_{0X_i}(y_i = 1)\big) \\ &= \frac{\hat{p}_{1X_i}(y_i = 1)\hat{p}_{0X_i}(y_i = 0)}{\hat{p}_{1X_i}(y_i = 0)\hat{p}_{0X_i}(y_i = 1)} \\ &= \text{L}(Y; T|X_i), \end{aligned} \tag{3.1}$$

where $L(Y; T|X_i)$ is the log-odds ratio of Y and T given X_i.

Then, analogous with the original method of Subsection 2.1.1, we search for regions of the data set where the new expression for Z exceeds some threshold c,

$$S_x^* = \{S_x : \text{L}(Y; T|S_x) > c\} \quad \Rightarrow \quad S_x^* = \{S_x : D(Y; T|S_x) > c\}. \tag{3.2}$$

Here, we capture the essential operation of the log-odds ratio by using notation D to represent the conditional relationship of Y and T in the predictive region S_x of the biomarker. So we are looking for subspaces of the data, S_x, where the treatment indicator is highly predictive of recovery outcome.

3.1.2 SIDES

The mechanism by which the SIDES algorithm determines the predictiveness of a feature is described in Equation 2.4. We use the D-notation to represent this functionality as

$$S_x^* = \arg\max_{S_x} \left(t_{S_x} - t_{S_x^c}\right) \quad \Rightarrow \quad S_x^* = \arg\max_{S_x} \big(D(Y; T|S_x) - D(Y; T|S_x^c)\big), \tag{3.3}$$

where $S_x \cap S_x^c = \emptyset$ and $S_x \cup S_x^c$ encompasses the full space of a biomarker.

From this viewpoint, the approaches to predictive biomarker detection taken by both Virtual Twins and SIDES are premised on similar rationales. The difference is that SIDES implements this criterion at each node of the tree, whereas Virtual Twins grows the decision tree in the typical way, by minimising the within node sum of squared errors, SSE $= \sum_{i \in d}(z_i - \bar{z}_d)^2$, where $\bar{z}_d$ is the maximum likelihood estimate of the treatment effect in node d, and then identifies the predictive biomarkers using the terminal node predictions of the finalised tree. This amounts to a less direct way to search for biomarkers associated with a differential treatment effect, but it is still calibrated to identify predictive and not prognostic biomarkers.

Both the Virtual Twins and SIDES methods are local, but SIDES is more extensively so. The Virtual Twins method models individual treatment effect Z by minimising the sum of squared errors in both child nodes, so that it is agnostic to the magnitude of treatment effect in either node when choosing the best split on a feature. It is only when the interesting subgroups are selected from the terminal nodes that a local modelling technique is used. It is local in the sense that the properties such as treatment effect and subgroup size are only estimated for the subgroups as defined by the terminal nodes that have a predicted treatment effect in excess of a threshold, c.

In contrast, SIDES maximises the difference in treatment effect between two child nodes, and, therefore, at each split point it is explicitly searching for predictive features. Local modelling is applied at every node in the sense that only the subgroup of the child node with the larger treatment effect is kept as a prospective subgroup of interest. The other node is peeled away so that it is no longer included in the modelling task. In this way, the SIDES algorithm searches for interesting subspaces and avoids the burden of having to model the whole of the data set. Theoretically, this is well suited to the task of subgroup identification, where we are not interested in the treatment effect of the whole sample, but, rather, only a small portion of it.

3.1.3 GUIDE

The GUIDE variable importance score described in Equation 2.5 can be rewritten as

$$X^* = \max_X D(R; X|T). \tag{3.4}$$

GUIDE relies on the conditional dependence of the *whole* space of the X feature on the variation in Y that is unexplained by main effects.

3.2 Summary

We can see that each method is, in theory, calibrated to identify predictive biomarkers and exclude prognostic and irrelevant biomarkers from selection. However, each method does it in a different way. Both Virtual Twins and SIDES partition variable X then measure the conditional relationship of Y and T in the potentially predictive subspace S_x. They differ in two important ways. The former uses a fixed threshold c

(see Eq. 3.2) to identify important predictor variables and this is only applied to the terminal nodes of the tree. On the other hand, the threshold of the latter, $D(Y;T|S_x^c)$ (see Eq. 3.3), is data dependant, and it is implemented at each node in the tree.

In contrast to Virtual Twins and SIDES, GUIDE avoids selecting features based on split-feature combinations. By doing this, GUIDE mitigates against the risk of false positives inherent in selecting features based on feature splits. It is effectively ranking the features and selecting the highest ranking result as the most promising feature. In this way it is *explicitly* searching for predictive features, a characteristic that it shares with SIDES.

Chapter 4

Empirical Analysis

The theoretic analysis of Chapter 3 reveals the central predictive feature search mechanism of each method. Beyond theoretical differences in their predictive feature selection strategies, there are also significant differences in implementation, as was outlined in Chapter 2.2. In this section, we explore the behaviour of each method empirically, making a cross-comparison of each. The data sets are described in Section 4.1. The experimental design and evaluation methods are outlined in Section 4.2. This is followed by results in Section 4.3.

4.1 Data Sets

Each method was originally tested using simulated data sets. For this empirical study we use the same simulation settings to generate three data sets, namely the Virtual Twins, SIDES and GUIDE simulated data. We wish to compare the performance of each method on each of these data sets.

The original **Virtual Twins model** has a binary response and is defined as

$$\text{logit}(P(Y=1)) = -1 + 0.5(X_1 + X_2 - X_3 + X_2X_3) + 0.1T + 0.9TI(X_1 > 0 \cap X_2 < 0), \tag{4.1}$$

where the features, X_j, $j = 1, ..., M$ are independent and normally distributed with $\mu = 0$ and $\sigma^2 = 1$. The total number of features is denoted M. Both of the predictive biomarkers *are also prognostic*. For our experiments we alter the model specification so that only one feature, X_1, is both predictive and prognostic, and the other, X_2 is predictive only. To make the comparison clearer, we also remove the X_2X_3 term.

The new model is

$$\text{logit}(P(Y=1)) = -1 + 0.5(X_1 - X_3) + 0.1T + 0.9TI(X_1 > 0 \cap X_2 < 0). \quad (4.2)$$

With this model we test the ability of each method to find a *three-way interaction*, $T \times X_1 \times X_2$.

For ease of reference, we define a feature that is both prognostic and predictive as 'progpredictive'. A feature is described as 'predictive' if it is *only* predictive and not prognostic also. The data_{VT} is the only data set that contains a progpredictive feature. Both the predictive and progpredictive features are *predictive* in the same way. In other words, both make identical contributions to the subgroup definition. As the features are symmetrically distributed with a mean of 0, $P(X_1 > 0) = P(X_2 < 0) = 0.5$.

The **SIDES model** is the only model with a continuous response. It is

$$Y = 0.6T(-2f + (X_1 = 1) + (X_2 = 1)) + \epsilon, \quad (4.3)$$

where the random noise $\epsilon \sim N(0, \sigma^2)$ and f is set to simulate a subgroup that covers $f^2 = 0.5$ of the observations [10]. There are no prognostic effects. Unlike the Virtual Twins and GUIDE data models, there are two subgroups present and they overlap. The subgroup effects are additive so that individuals that have the feature values $X_1 = 1$ *and* $X_2 = 1$ are members of both subgroups, whereas individuals that have either $X_1 = 1$ *or* $X_2 = 1$ are members of one subgroup only.

The **GUIDE model** is

$$P(Y=1) = 0.3 + 0.2(I(X_1 \neq 0) + I(X_2 \neq 0)) + 0.2(2T-1)I(X_3 \neq 0 \cap X_4 \neq 0). \quad (4.4)$$

It has three-level categorical features. These simulate genetic markers with genotypes AA, Aa and aa. Features $X_1, ..., X_4$ follow the same distribution with $P(X = AA) = 0.4$, $P(X = \text{Aa}) = 0.465$ and $P(X = \text{aa}) = 0.135$. The remaining of the features, $X_5, ..., X_M$, where M is the total number of features in the data, have the distribution $P(X = \text{AA}) = (1-\pi)^2$, $P(X = \text{Aa}) = 2\pi(1-\pi)$ and $P(X = \text{aa}) = \pi^2$. Thus, π defines the distribution of genotype values for each of the features, and each π_j, $j = (5, ..., M)$, is independent and simulated from a beta distribution with density $f(x) \propto x(1-x)^2$. Similarly to the Virtual Twins data model, the GUIDE data model will test the ability of each subgroup identification method to distinguish between

	Y	X	OTE	subgroup size	STE	STE - OTE
data$_{VT}$	binary	continuous	0.07	$0.25N$	0.23	0.16
data$_{SIDES}$	continuous	binary	0	$0.50N^*$	0.35^*	0.35
data$_{GUIDE}$	binary	categorical	0.14	$0.36N$	0.40	0.26

Table 4.1: The three data models are summarised in terms of overall and subgroup treatment effect (OTE and STE) and subgroup size as a proportion of total number of observations, N. *In the SIDES data model, $0.50N$ of the observations have full subgroup membership, but $0.91N$ have *partial or full* membership; full subgroup members have a treatment effect of 0.35, while the average treatment effect across the group that includes both partial and full members is 0.08.

predictive and prognostic features; and it also contains a three-way interaction, $T \times X_3 \times X_4$.

Data generated from these models will be referred to as data$_{VT}$, data$_{SIDES}$, and data$_{GUIDE}$ respectively. Within each data set, the X-features are all of the same type: data$_{VT}$ features are all continuous, in data$_{SIDES}$ they are all binary and they are all categorical in data$_{GUIDE}$. While this means that we are not testing the ability of each method to identify predictive features in the presence of a variety of data types, it permits the observation that some methods are better suited to a particular class of features than others. This is discussed further in Section 4.3.

The data sets are summarised in Table 4.1. They are highly dissimilar in several ways. Rather than making them more alike, they are left in this form in order to preserve the data generation methods as they were used in the original papers [28], [22], [11].

4.2 Experimental Method

For ethical, financial, and practical reasons, clinical trials are limited in the number of patients they include. We compare the performance of each method on each of the Virtual Twins, SIDES and GUIDE data sets, varying the number of observations, N, between 200 and 1000 to understand the relationship of model performance to clinical trial size. The number of features is fixed at $M = 20$.

As more patient data becomes available, for example, there may be thousands of genetic features of unknown importance, subgroup identification models will need

to have the power to detect important features while also minimising the risks of multiplicity. To test each method in this respect, we vary the number of features M from 20 to 100, while keeping the number of observations fixed at $N = 500$. Note that, as M increases, the number of predictive and prognostic features stays the same. It is only the number of irrelevant features, those that have no bearing on the outcome Y, that increase with M. For example, data_{VT} has one each of predictive, progpredictive and prognostic features. So if $M = 20$, then there are $M - 3 = 17$ irrelevant features in the data set.

Each method is, in theory, tuned to rank predictive over prognostic features. We compare the performance of each method on data_{VT} and data_{GUIDE}. As data_{SIDES} does not contain prognostic features, it is excluded from this part of the analysis. For data_{VT} we split the analysis of the progpredictive feature out from that of the predictive feature. Thus, for each of the methods, we make three comparisons in all: on data_{VT}, we compare the ranking of the progpredictive to the prognostic feature, we also compare the solely predictive feature to the prognostic one, and on data_{GUIDE} we compare the ranking of the predictive features against the prognostic ones.

Subgroup identification in clinical trial data is challenging. It is often the case that just enough individuals are recruited to a trial to test the main effects on the outcome, whereas predictive feature identification involves the detection of treatment-variable interactions. Thus, many clinical trials are not powered for interaction detection. This leads to a high rate of false positives and false negatives when detecting important features.

Due to this inability to arrive at strong statistical conclusions, these methods should be used in conjunction with the insight of medical experts. Converting a subgroup method to a variable importance score gives clinicians the flexibility to set the significance threshold on how many features they are willing to examine manually. We examine the performance of each method on the Receiver Operating Characteristic (ROC) curve [29] and explore means of making the final selection of important features using the Virtual Twins PVIM.

The Virtual Twins method has one hyperparameter, the number of trees in each random forest. This is set to 1000. SIDES has several hyperparameters: the maximum depth of the branches, $L = 3$, limits the complexity of the subgroup definition; the

minimum subgroup size, $S = 40$, ensures that the calculation of the one-sided test statistic is based on an adequate sample size; the maximum number of candidate splits considered at each node, $M = 5$, limits the search space so that the algorithm runs efficiently; and the continuation criterion $\gamma = 0.5$ ensures that a subgroup defined by a child node has a splitting criterion p-value that is, at most, 0.5 times the p-value of the parent subgroup. There are no parameters that need to be set for the GUIDE method.

For Virtual Twins and SIDES, the models are built on bootstrapped samples and the variable importance scores are calculated on the out-of-bag samples. This happens by default in the random forest of the Virtual Twins method. We implement this approach in SIDES in order to make the two methods more comparable. Bootstrapping also mitigates against the risk of multiplicity. The GUIDE method is not improved with bootstrapping because calculating the chi-square statistics is a high bias, low variance approach.

All experimental results are based on 500 iterations of each data set. In one iteration of one data model, we generate a data set and apply each of the Virtual Twins, SIDES and GUIDE methods to it. For that same data model, we repeat this process of sampling a data set and applying the subgroup identification methods for 500 iterations.

4.3 Results

4.3.1 ROC curves

We use ROC curves to evaluate the ranking performance of each method on each of the data sets. The methods return a PVIM for each feature and the ROC curves plot the true positive false positive trade-off for every possible threshold setting. The true positive rate, or sensitivity, for a given PVIM threshold represents the proportion of predictive features that are identified out of the total number of predictive features for all iterations. Conversely, the false positive rate, or (1− sensitivity), is the proportion of unimportant features that were incorrectly identified as being significant out of the total number of unimportant features. The further a curve is into the top left corner, the better the method, because it indicates that a high true positive rate and low

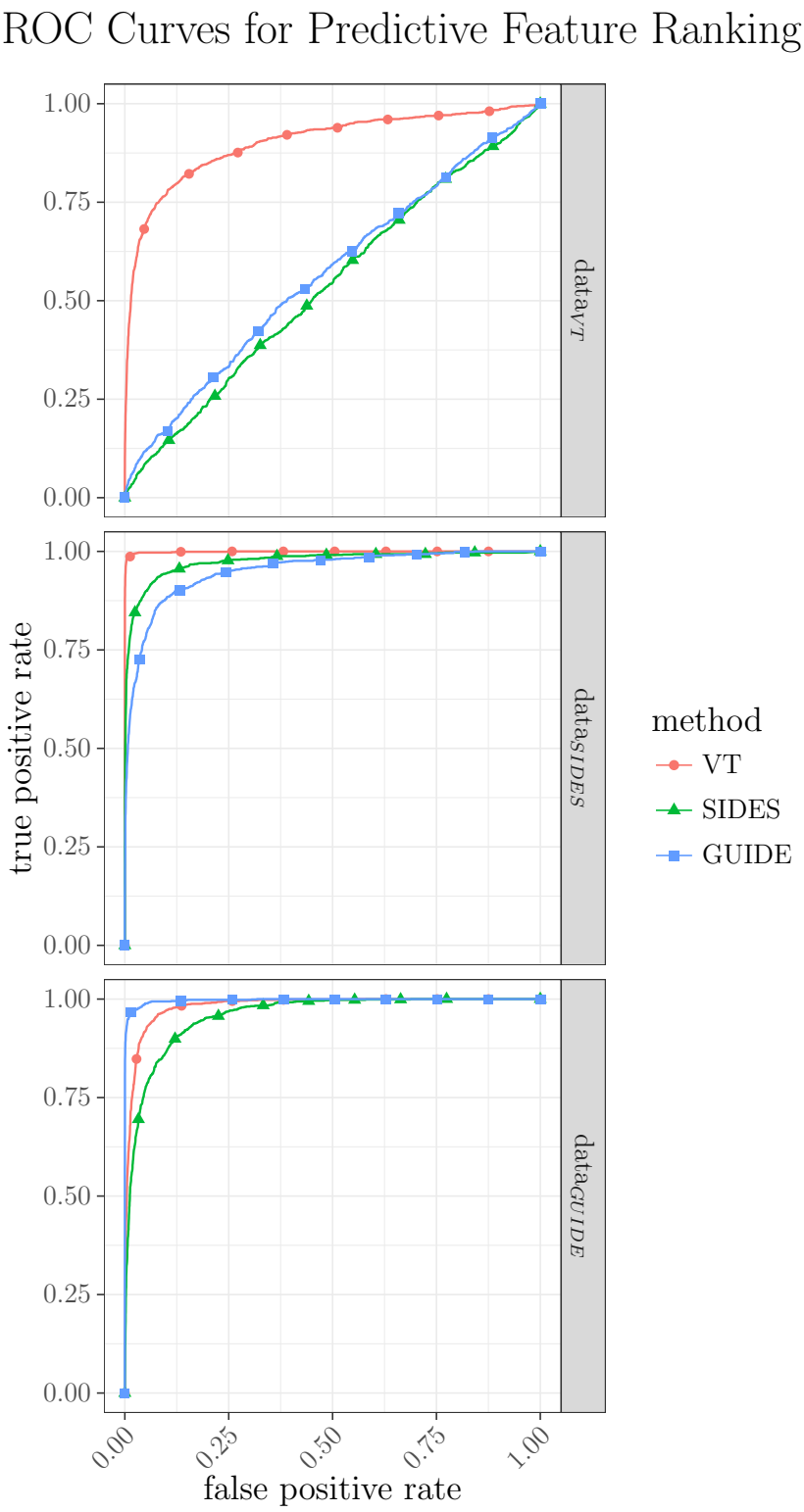

Figure 4.1: ROC curves compare the true/false positive trade-off for different thresholds of the variable importance score. Results are based on 1000 observations and 20 features in each of the data sets.

false positive rate has been achieved. The best case scenario is if a method ranks all the predictive features higher than all the rest every time. In that case, the curve is composed of a straight line from 0 to 1 on the y-axis and a straight line from 0 to 1 on the x-axis. The true positive rate in that case would be 1 and the false positive rate 0. An example of this in Figure 4.1 is the Virtual Twins method when applied to data_{SIDES} - it achieves almost perfect ranking of the predictive features. If, on the other hand, the curve is close to the diagonal from the bottom left to the top right, this indicates that the model is no better than random in the binary classification setting. We can see from Figure 4.1 that this is the case for the SIDES and GUIDE methods when applied to data_{VT}.

The subgroups in data_{SIDES} and data_{GUIDE} are easiest to find, as can be seen in Figure 4.1. Recalling the data set summary of Table 4.1, the subgroup population in data_{SIDES} is the largest, with $\sim 50\%$ of the population having full subgroup membership and a further $\sim 41\%$ having partial membership, and this makes it easier to identify the predictive features. The only data set with additive subgroup effects is data_{SIDES}. Partial subgroup membership is not possible for either of the data_{VT} or data_{GUIDE}.

The Virtual Twins method is the best of the three. Even though it performs worse than GUIDE on data_{GUIDE}, it has the best overall performance across the three varieties of data. In particular, on the data set with the weakest subgroup signal, data_{VT}, it outperforms SIDES and GUIDE by a long way for all levels of threshold setting.

The ROC curve results are based on data sets with $M = 20$ features and $N = 1000$ individuals. In the following three subsections, we examine the performance of each method as M and N change.

4.3.2 Varying the number of samples

The less data there is available, the more difficult it is to detect the subgroup signal. We compare each method in terms of its ability to rank both of the true predictive features in the top K, $K \in 2, 5$ as N varies. For all levels of N the Virtual Twins method outperforms SIDES and GUIDE on data_{VT} and data_{SIDES} but performs worse than GUIDE on data_{GUIDE}. From Figure 4.2 it seems that, to achieve good results

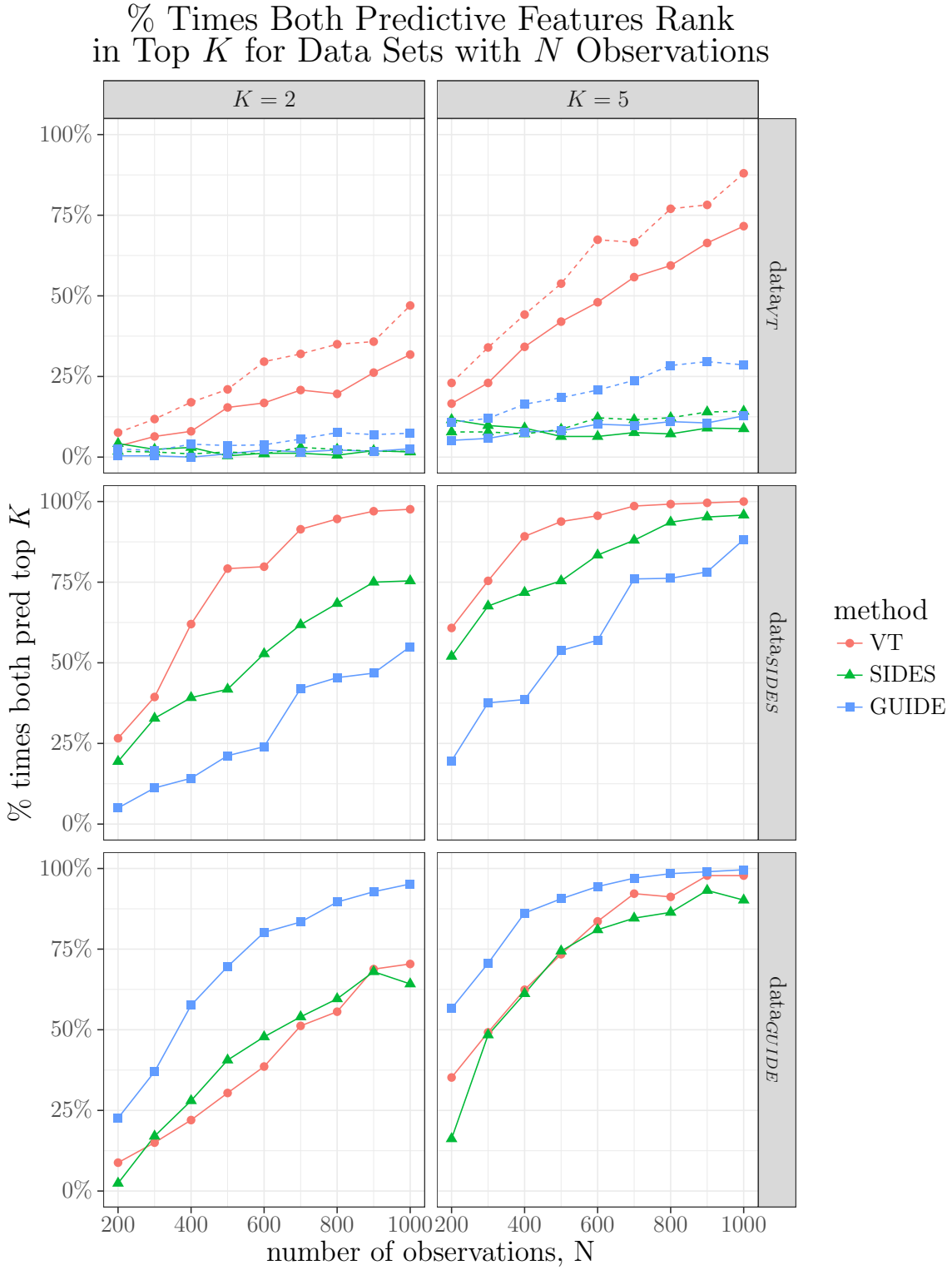

Figure 4.2: The % times both predictive features (pred) are ranked in the top K, $K \in (2, 5)$, is evaluated for each model as sample size, N, varies. The dashed lines in the data_{VT} correspond to the results that are obtained when the original Virtual Twins data simulation model as specified in Equation 4.1 is used.

with this method, data sets of at least 500 observations are required and performance degrades significantly on smaller data sets.

The dashed lines in the top two plots of Figure 4.2 corresponding to data_{VT} represent the results for each method when the original data_{VT} simulation of Equation 4.1 is used. The Virtual Twins method performs better on this data set than it does on the altered specification that is described in Equation 4.2. The key difference between these two is that, in the former, bother the predictive features are also prognostic, whereas, in the latter, only one feature is progpredictive and the other is solely predictive. This indicates that the Virtual Twins method is not only detecting predictive signal, but also prognostic signal. In other words, its success at detecting a predictive feature is affected by the extent to which that feature is also prognostic. We explore this further in Subsections 4.3.4 and 4.3.5. To a lesser extent, there is also an improvement in the results of the SIDES and GUIDE methods when applied to the original data_{VT}.

The data_{GUIDE} has the largest subgroup treatment effect of the three methods, which also makes the subgroup signal more easily detectable. The Virtual Twins data has the smallest subgroup membership and subgroup treatment effect of the three data sets. For $N = 1000$, the Virtual Twins method ranks both predictive features the highest 27.8% of the time.

The SIDES and GUIDE methods are not agnostic to data type. There is a qualitative variance in each method's performance across the data sets which differ in outcome type (binary or continuous) and predictor feature type (binary, continuous or categorical). Both the Virtual Twin and GUIDE data simulations use a binary outcome, but the features of the former are continuous whereas they are categorical for the latter. GUIDE outperforms the Virtual Twins method on its own data set but underperforms relative to Virtual Twins on data_{VT}. This suggests that both the Virtual Twins and GUIDE methods are sensitive to feature type. Similarly, data_{SIDES} and data_{GUIDE} have comparable features, binary and three-level categorical, yet the SIDES method outperforms GUIDE on its own data set but performs significantly worse than GUIDE on data_{GUIDE}.

There is a general remark to be made about benchmarking methods which will become increasingly important as the subgroup identification literature expands. Each

method performs best on its own data set which suggests that there is a biased approach to method evaluation. It indicates that there is a need to provide benchmark data sets against which subgroup identification methods can be compared, a point that has already been raised by Shen et al. [30]. The best results from the SIDES method are when it is applied to data$_{SIDES}$. Similarly, GUIDE outperforms the other two on its own data set. However, it is ineffective on the Virtual Twins data set and falls far short of the Virtual Twins and SIDES methods on the data$_{SIDES}$ set. The original Virtual Twins data model, as specified by [28] and described in Equation 4.1, only contains progpredictive features, which disguises the aforementioned weakness of the Virtual Twins method in identifying features that are solely predictive.

4.3.3 Varying the number of irrelevant features

Similarly to the previous subsection, we compare the ability of each method to rank the predictive features in the top 2 and top 5, this time varying the total number of features, M. The simulated data model specifications remain the same, so that, by increasing the number of features, we increase the number of irrelevant features only. Thus, in Figure 4.3 we are evaluating the false positive rate of each subgroup identification method as more irrelevant features are added. The number of observations is fixed at $N = 500$.

The trends in performance decay as M increases is linear in most cases. The GUIDE method has a slower rate of diminishing performance on its own data set as we add more irrelevant features. Similarly, the performance of the Virtual Twins method degrades more slowly that SIDES and GUIDE when applied to data$_{SIDES}$, however, it is badly affected by the addition of features on data$_{VT}$.

4.3.4 Distinguishing predictive from prognostic features

Although prognostic features provide useful insights for general patient care, at times we are exclusively interested in finding predictive features. We evaluate each method in terms of its ability to specifically identify features that *interact* with the treatment to affect recovery outcome as opposed to identifying features that directly affect the outcome.

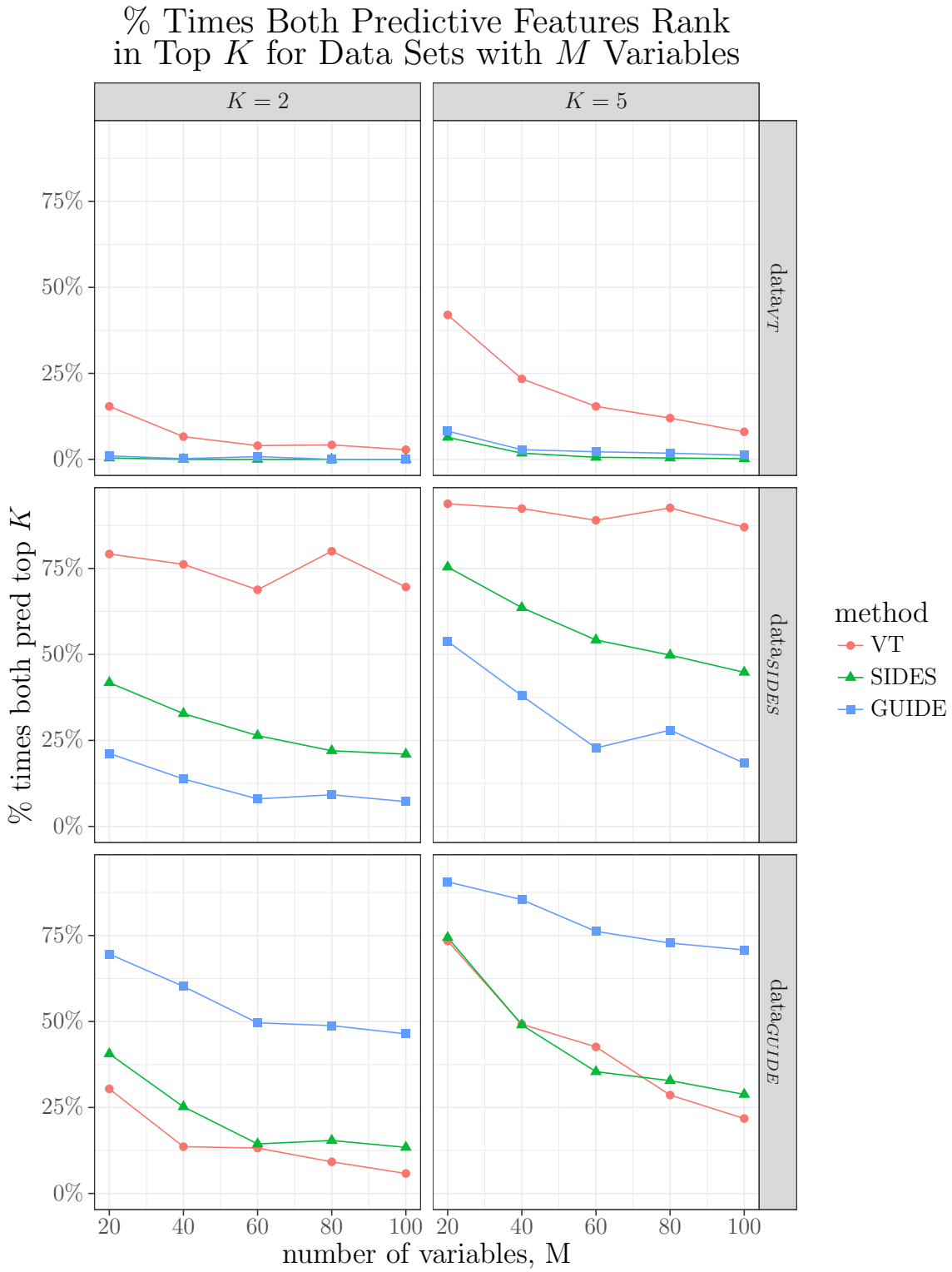

Figure 4.3: The % times both predictive features (pred) are ranked in the top K is evaluated for each model as the number of irrelevant features, M, varies.

The Virtual Twins data set contains two predictive features. Feature X_1 is progpredictive, whereas X_2 is predictive only. Both are *equivalent* in terms of their predictiveness in so far as they follow the same distribution, $\sim N(0,1)$, and have identically sized regions of $X_1 > 0$ and $X_2 < 0$ that define the subgroup membership. Feature X_3 is prognostic only. The analyses of both predictive features are split out so that we compare the performance of each method, first, in ranking feature X_1 over X_3, and, second, in ranking X_2 over X_3. In Figure 4.4 we can see that each method performs similarly when ranking the predictive feature that is progpredictive, X_1, over the prognostic feature X_3. The Virtual Twins method consistently outperforms the other two by a narrow margin. However, Virtual Twins performs much worse in ranking X_2, a solely predictive feature, above X_3. Interestingly, the more observations there are, the worse it becomes at ranking the predictive over prognostic feature. This indicates that what the Virtual Twins method is really detecting is prognostic feature signal.

SIDES and GUIDE perform very consistently across both predictive and progpredictive feature ranking. Although both methods perform weakly on the Virtual Twins data set in terms of ranking both the predictive features in the top K (for example, the best performance in either when $M = 20$ and $N = 1000$ is 2.6%, whereas the probability of selecting the two predictive features in a random sampling of the X_j, $j = 1, ..., 20$ is 0.53%) the fact that they nevertheless perform strongly in ranking the predictive over prognostic suggests that their poor performances overall are due to unusually high scores on the irrelevant features rather than low scores on the predictive features. In other words, despite the multiplicity controls, there is a high false positive rate associated with these methods.

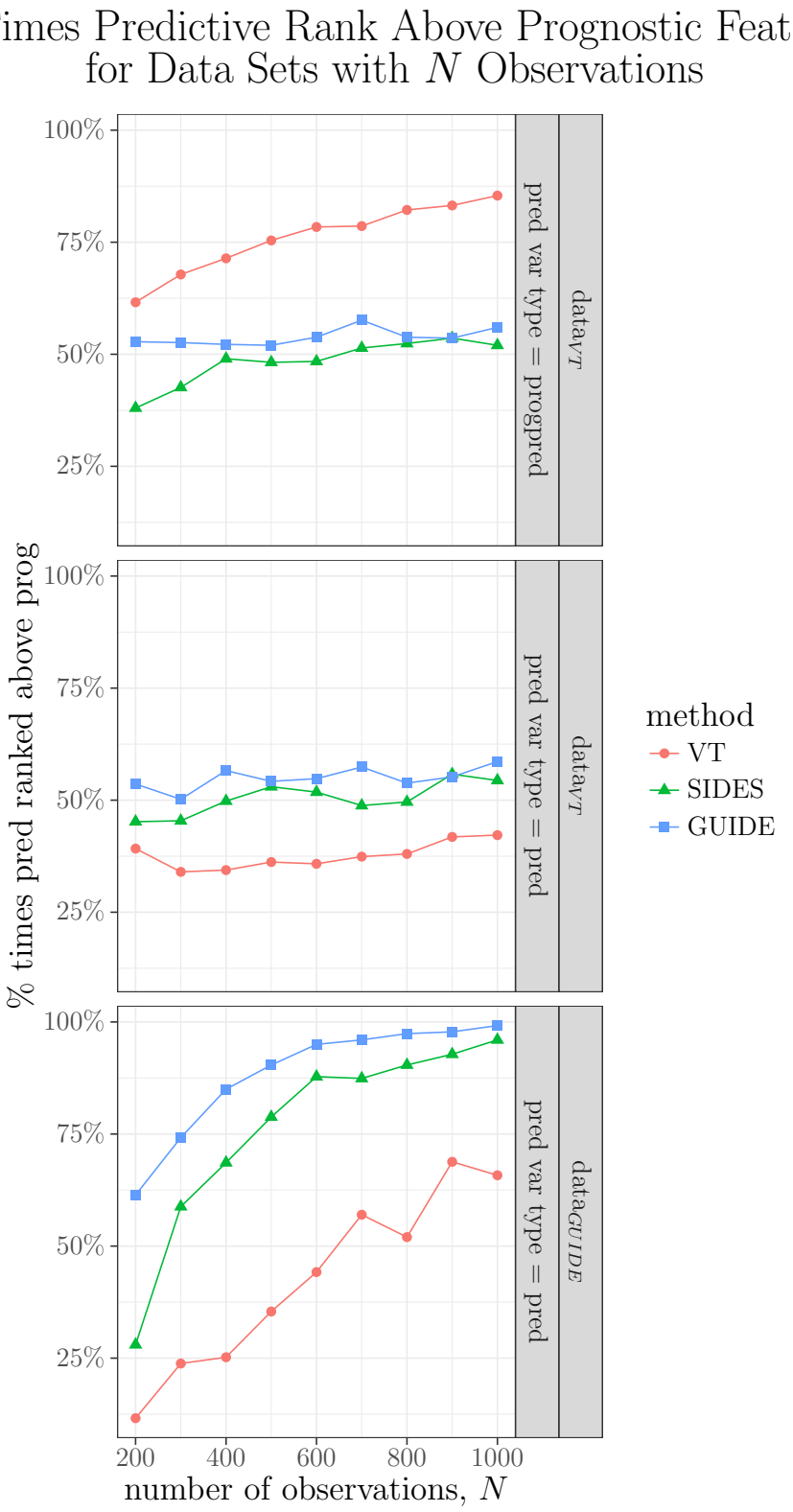

Figure 4.4: Testing the ability of each model to rank predictive (pred) above prognostic (prog) features as the number of observations, N, varies. As the Virtual Twins data has two different types of predictive features, one that is both prognostic and predictive (progpred) and one that is just predictive, we analyse each separately.

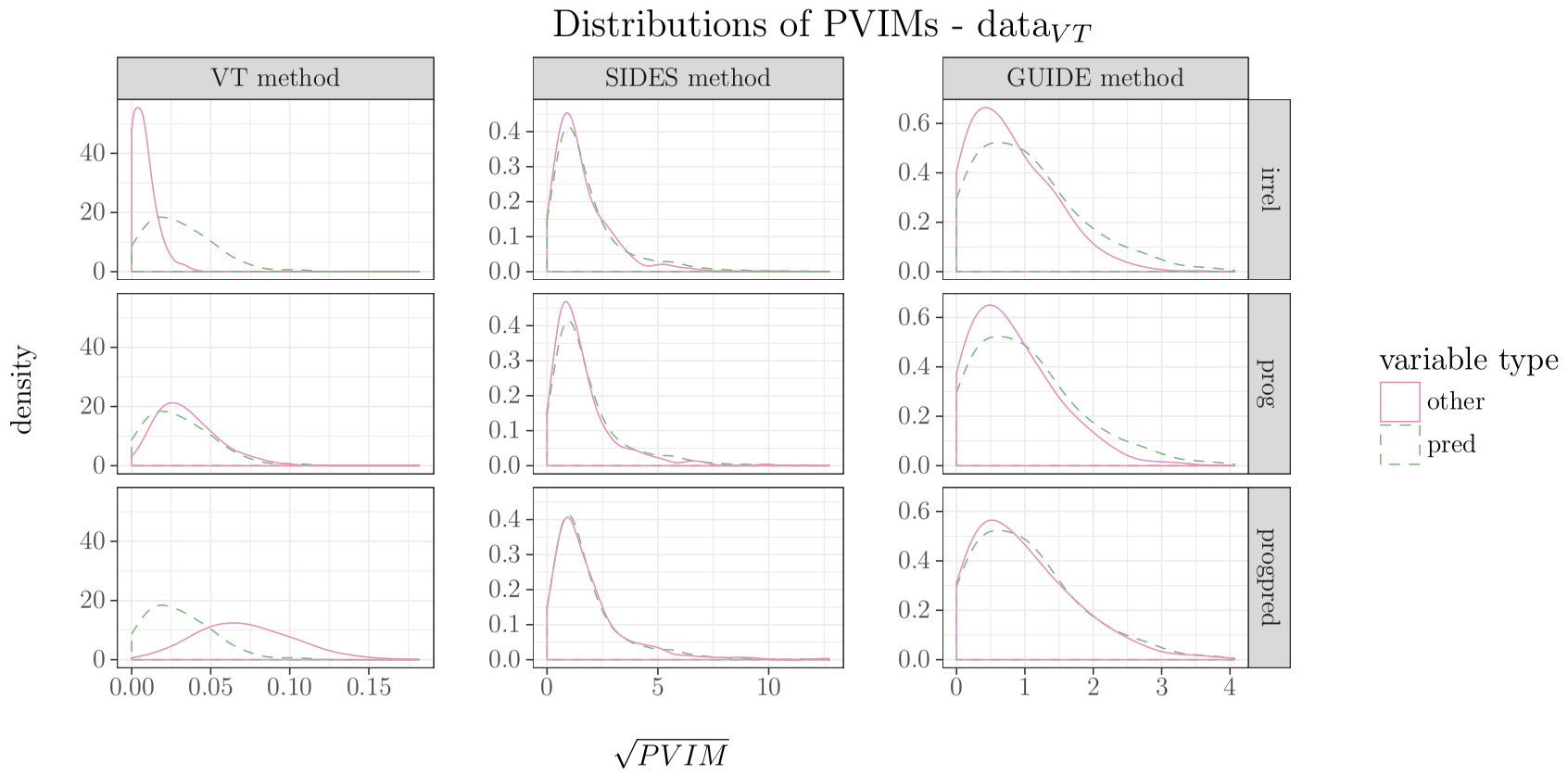

Figure 4.5: For each method, Virtual Twins, SIDES and GUIDE we observe the distribution of the square root of the predictive variable importance (PVIM) scores split out by the variable types: predictive (pred), prognostic (prog), both prognostic and predictive (progpred), and irrelevant (irrel). This pertains to data$_{VT}$ only. Each of the PVIMs are calculated on data sets of size $N = 1000$. Each distribution is based on 500 samplings of a PVIM.

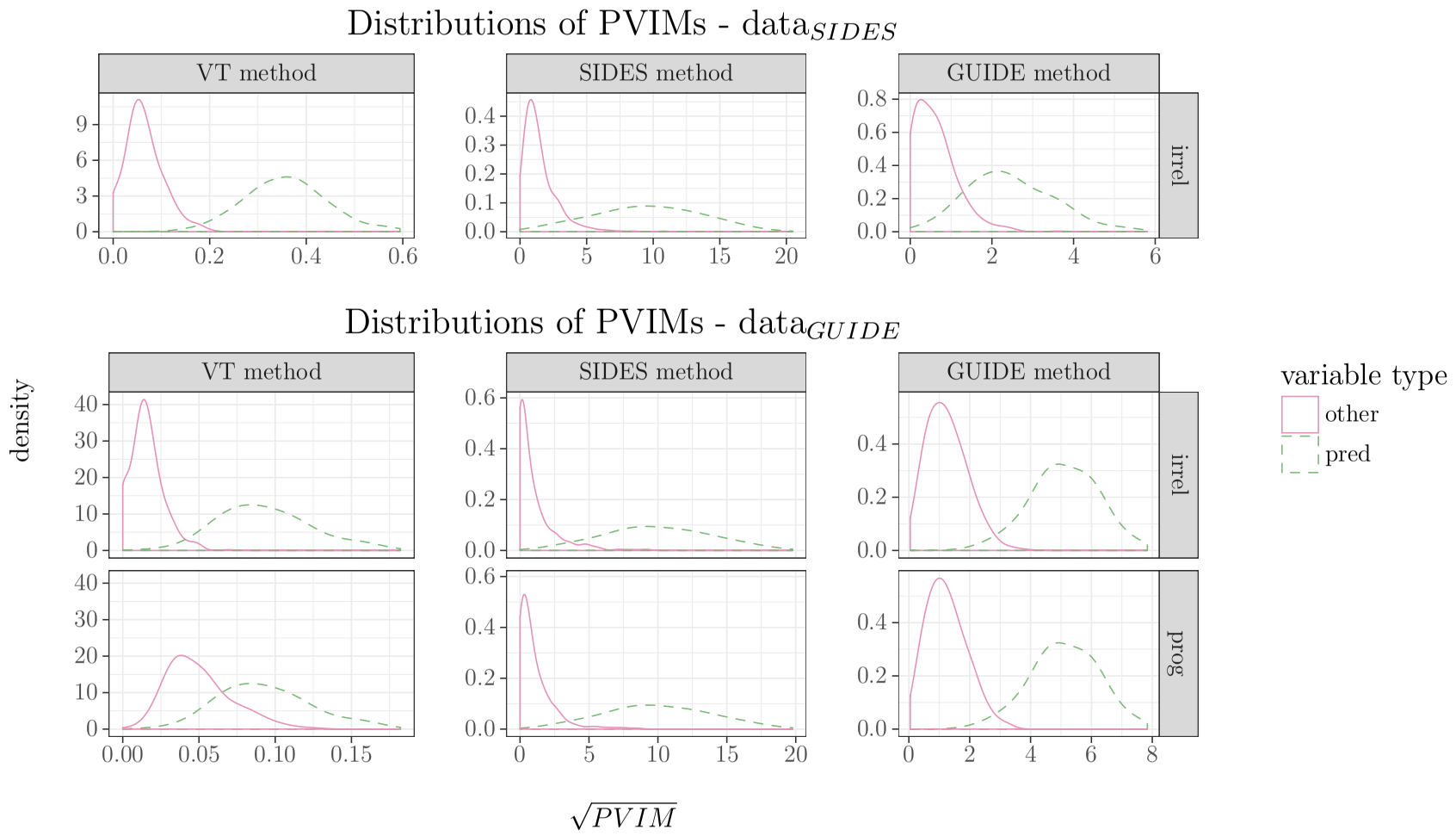

Figure 4.6: For each method, Virtual Twins, SIDES and GUIDE, on data sets data$_{SIDES}$ and data$_{GUIDE}$ only, we observe the distribution of the square root of the predictive variable importance (PVIM) scores split out by the variable types: predictive (pred), prognostic (prog), both prognostic and predictive (progpred), and irrelevant (irrel). Each of the PVIMs are calculated on data sets of size $N = 1000$. Each distribution is based on 500 samplings of a PVIM.

4.3.5 PVIM Distributions of Each Feature Type

For each method we compare the distributions of the PVIMs for predictive and irrelevant features in each of the data sets. Where prognostic or progpredictive features are also present, we examine these distributions too. In Figures 4.5 and 4.6 we plot the square root of the PVIM distributions. These are based on data_{VT}, data_{SIDES}, and data_{GUIDE}, each of size $N = 1000$ with $M = 20$ features. The distributions are obtained by calculating the PVIMs for each feature on 500 data sets resampled from each of the Virtual Twins, SIDES, and GUIDE simulation models. So, for example, the PVIM on the progpredictive feature in data_{VT} is based on 500 PVIM values of feature X_2.

Where a PVIM value is less than 0 it is set to 0 before taking the square root. The negative values occur when the results obtained on the permuted feature are *better* than those obtained on that feature when it is not permuted. This occurs for features that have no bearing on the outcome, nor any correlations with the other features, so that, when they appear in a tree, it is because noise has erroneously been modelled as signal. There is a chance that, when the irrelevant feature is permuted, it will yield a better result than the unpermuted version. As we don't expect the difference in predictibility between the permuted and unpermuted feature to be large, one will only be better than the other by a narrow margin.

Where there are multiple instances of a feature type, for example, there is always more than one irrelevant feature in a data set, we plot the distribution of one irrelevant feature only, say X_9. As these features are drawn from the same distribution, X_9 is representative of the rest.

The data_{VT} has the weakest signal and each method performs poorly on it, Figure 4.5. The Virtual Twins method consistently gives the prognostic and progpredictive features higher PVIMs, as can be deduced from the fact that the distributions for these two feature types are shifted *to the right* of the predictive feature distribution. A further indication that it's mostly detecting the *prognostiveness* of the progpredictive feature.

There is much stronger subgroup signal in data_{GUIDE} than there is in data_{VT} (see Table 4.1) and the Virtual Twins method performs well in identifying the predictive feature in this setting. However, when a method is successfully callibrated to search for

predictiveness only, we expect the PVIM distributions of the irrelevant and prognostic features to be the same, as neither are predictive. Both SIDES and GUIDE do this very well, as can be deduced from comparing performances across irrelevant and prognostic features on data_{GUIDE} in Figure 4.6. When irrelevant and prognostic features are present in a data set, the PVIM distributions of each are near identical. On the other hand, the distribution of the data_{GUIDE} prognostic feature is shifted to the right of the irrelevant feature when the Virtual Twins method is used. It is erroneously detecting predictiveness in a feature that has no predictive properties.

One caveat in the evaluation of GUIDE is that it uses a linear model to account for prognostic effects of a feature, and it is only tested on data sets with linear prognostic effects. Further analysis of GUIDE could evaluate its performance on a non-linear prognostic feature. Since the Virtual Twins and SIDES methods use partitioning, they should perform similarly in the non-linear setting.

4.3.6 Interpreting the Output of Virtual Twins

Although the Virtual Twins method is prone to detecting subgroup predictiveness where it shouldn't, in prognostic features, it is the best method overall. In particular, on the data set with the weakest subgroup signal, data_{VT}, it outperforms SIDES and GUIDE by a significant margin. Thus, we conclude that it is the best method for the purpose of ranking features in order of subgroup predictiveness. The final set of experiments focus on the Virtual Twins method only and demonstrates how the method can be used in practice. We use plots of the PVIMs, Figure 4.7, and partial dependence plots, Figures 4.8 and 4.8. Although results are based on data sets with $M = 20$ features and $N = 1000$ observations, only the results for the first 8 features are displayed.

Plots of the PVIMs for each feature provide an easily interpretable visualisation of the features that have been found to be important by Virtual Twins. Following the approach of Genuer et al. [31], the Virtual Twins method is rerun 50 times on the same data set. Because the random forest algorithm is not deterministic, the results vary a little each time, and this enables us to obtain a mean and variance on the PVIM of each feature. The upper plot in Figure 4.7 demonstrates what these look like in the case where the ranking is successful and the lower plot provides an example of what

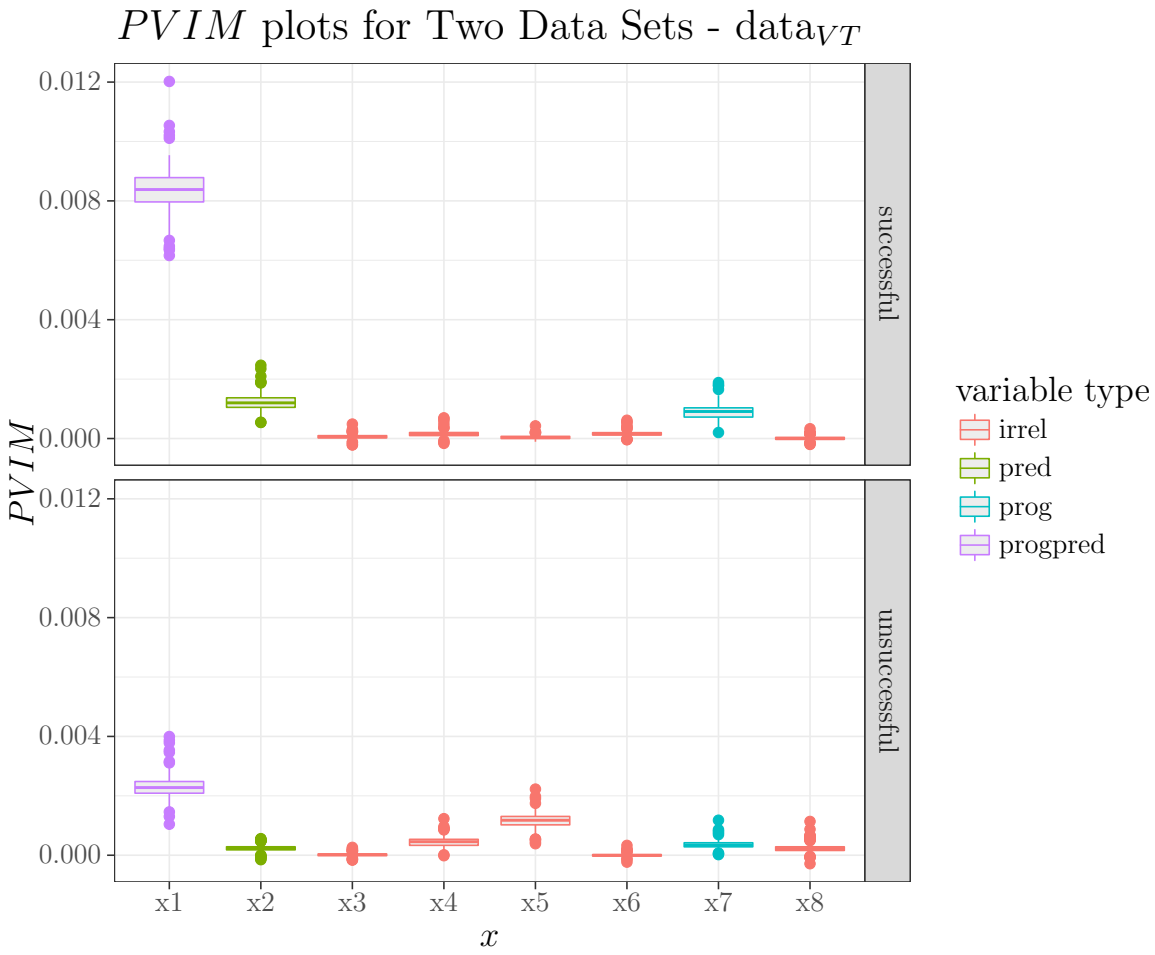

Figure 4.7: PVIM plots for two data sets, both of which have been sampled from data$_{VT}$ model, Eq. 4.2. The top is an example of a successful ranking, as both predictive features, X_1 and X_2 rank highest. The bottom plot is an example of an unsuccessful predictive feature ranking because feature X_2 has been ranked lower than irrelevant features X_4 and X_5, and prognostic feature X_7.

these plots looks like when the Virtual Twins method does not rank the predictive features highest.

The PVIM plots are useful for understanding the relative magnitude of PVIM ranking across the features and for picking out interesting features for closer examination. However, these do not provide an understanding about *how* a feature relates to the differential treatment effect, Z. For this we use partial dependence plots [32]. The partial dependence plots visualise the *marginal* effects of a variable on an outcome. We demonstrate how they are applied on data_{VT} only, and look at the clear cut case in Figure 4.8 first, where the variable importance plots conclusively identify the predictive features, examining the partial dependence plots for the predictive, prognostic, and a selection of the irrelevant features. For completeness, we compare this with a case where the predictive features are not found and false positives are present, Figure 4.9. The data for both sets of plots correspond to the successful and unsuccessful examples of data_{VT} that are used in Figure 4.7.

In Figure 4.8 we can see that the partial dependence plots correctly represent the *nature* of the effect of X_1 and X_2 on treatment effect Z, although the results are more pronounced for the progpredictive feature X_1. The ticks at the bottom of each plot represent the density of X_i for $i \in 1, ..., 8$. We can see that the important trends in the plots of X_1 and X_2 are based on regions of each feature where there are plenty of observations, making them more reliable than would be the case if we made the observation in sparser regions.

From the data_{VT} PVIM distributions of Figure 4.7 corresponding to the case when the ranking is *not* successful, we can see that features X_4 and X_5 received a higher PVIM than predictive feature X_2. A clinician can look at the results of the partial dependence plots for these features in Figure 4.9 and evaluate whether the discovered relationship makes biological sense and warrants further experimentation, perhaps in another clinical trial which would test for these effects in a more rigorous, rather than exploratory, statistical setting.

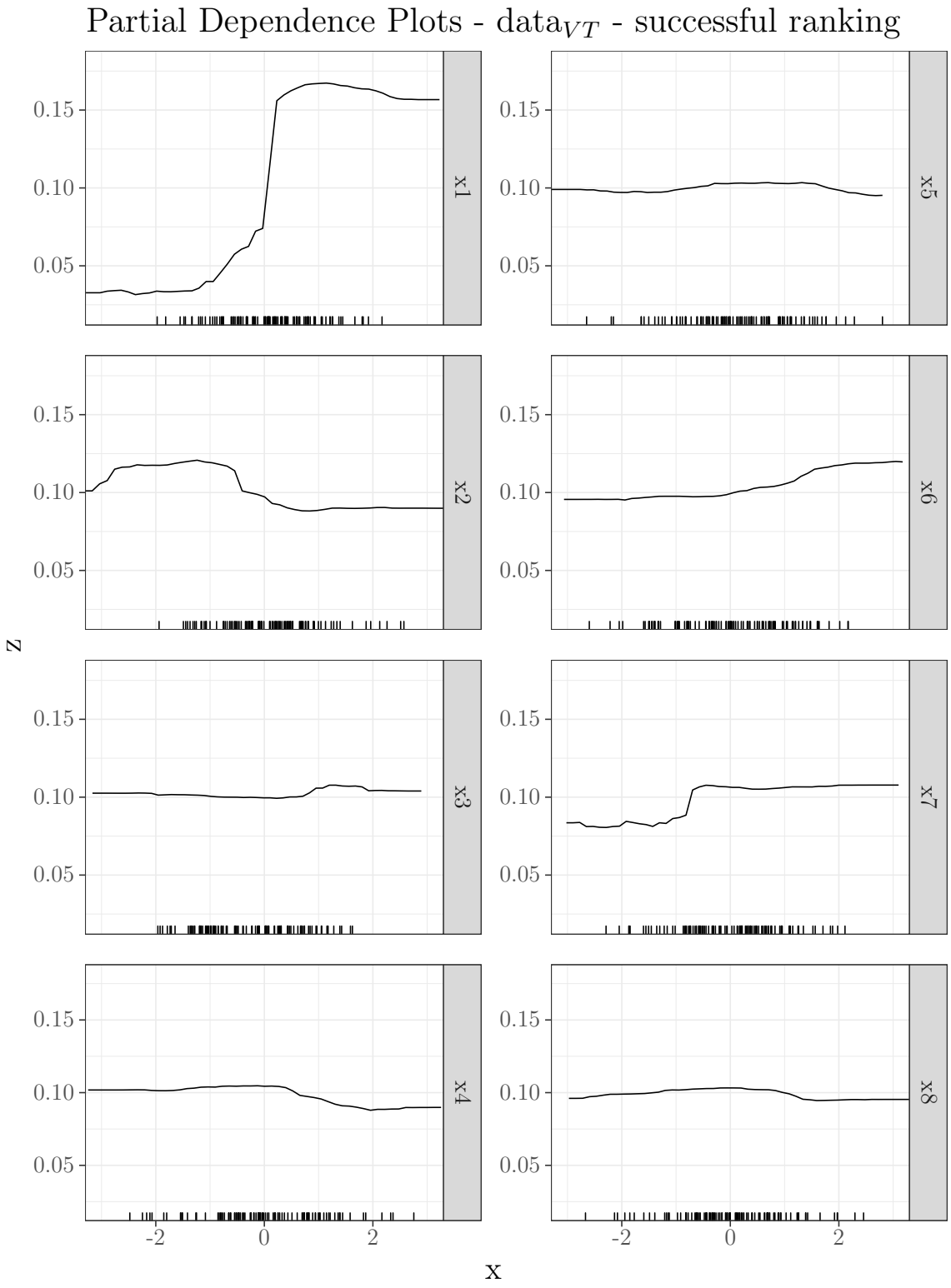

Figure 4.8: Partial dependence plots of Z on X_i for $i \in 1, ..., 8$ on a single sampling of data$_{VT}$ of size $N = 1000$ with $M = 20$ features. The rug at the bottom of each graph indicates the density of feature X.

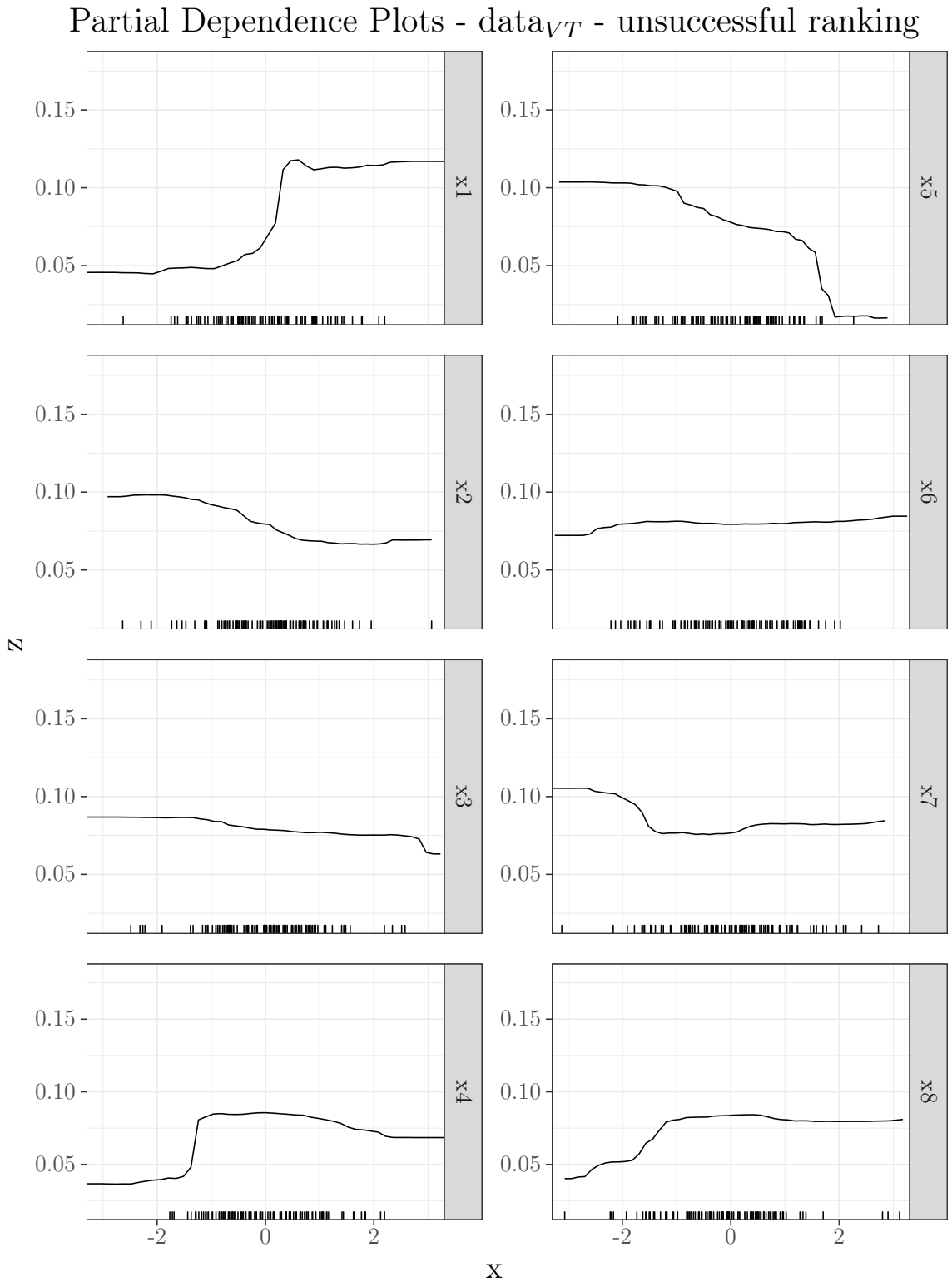

Figure 4.9: Partial dependence plots of Z on X_i for $i \in 1, ..., 8$ on a single sampling of data$_{VT}$ of size $N = 1000$ with $M = 20$ features. The rug at the bottom of each graph indicates the density of feature X.

Chapter 5

Conclusion

We have studied three methods of subgroup identification that use recursive partitioning, Virtual Twins, SIDES, and GUIDE. Recursive partitioning-based methods are popular in this area because they are non-parametric, so that they readily handle different types of variables (binary, continuous, etc) that are frequently encountered in complex medical data sets. We analyse and compare the three methods from a theoretical, and then an empirical, point of view.

The theoretical analysis of Virtual Twins, SIDES, and GUIDE reveals that each method has a different mechanism for selecting predictive features, but that Virtual Twins and SIDES are closely related:

$$\underbrace{D(Y;T|S_x) > c}_{\text{Virtual Twins}} \qquad \text{versus} \qquad \underbrace{\left(D(Y;T|S_x) - D(Y;T|S_x^c)\right)}_{\text{SIDES}}.$$

Yet empirical analysis reveals that the former consistently outperforms the latter. This implies that the more important difference between the two is in the implementation of the feature selection mechanism. The Virtual Twins method identifies interesting features as those associated with *terminal node* predictions of treatment effect that exceed a certain threshold. On the other hand, SIDES applies the predictive variable selection mechanism *at every node split in the tree*. SIDES explicitly searches for predictive features at each node split whereas Virtual Twins implicitly does so by minimising the mean squared error in predicting latent variable Z at each node split.

A theoretical analysis of GUIDE demonstrates that it uses the different strategy of selecting features X that are conditionally informative about the outcome Y given treatment T. Unlike Virtual Twins and SIDES, it does not choose features based on

the best feature-split combination. Instead, it looks at the full space of a feature. By taking this approach the authors endeavour to avoid predictive feature selection bias that occurs when there are different types of features in a data set, such as categorical and continuous features (see Subsection 2.1.2 for further description of this type of bias). However, GUIDE had a poor performance on data_{VT} and data_{SIDES}, and this may be attributable to the fact that it is examining all of the feature $D(Y; X|T)$ as opposed to an interesting region of it. Thus, honing in on an interesting feature subspace and then testing the relationship between the outcome and treatment in that subspace $D(Y; T|S_x)$, as Virtual Twins and SIDES do, may be a better strategy for detecting subgroup signal that may be too weak to detect on the full variable space [23].

Empirical analysis shows that the Virtual Twins method outperforms SIDES and GUIDE in ranking predictive features. It performs well on all three data sets, unlike SIDES and GUIDE which appear not to be agnostic to data type. SIDES performs better on data_{SIDES} with a continuous outcome and binary features, whereas GUIDE works better on data_{GUIDE}, which is comprised of a binary outcome and three-level categorical features. One drawback of Virtual Twins is that it struggles to differentiate between predictive and prognostic features. Although the algorithm is calibrated to find variables associated with a differential treatment effect, i.e. *predictive* variables, it nevertheless gives a high score to prognostic variables too.

An opportunity for future work could involve looking at using the Virtual Twins method as a predictive *and* prognostic feature detector instead, as opposed to a solely predictive feature identifier. However, it should not yet be relied upon in this additional capacity. Restraint in this respect is particularly important because it is not clear how the detection of prognostic variables is occurring. The method is callibrated to identify predictive variables to the exclusion of prognostics, as is made clear, both in the author's statement of purpose [9], and by examining the mathematics that describe the calibration of the model for identifying *predictive* features only, Eq. 2.2. Although there is evidence that it is capable of detecting prognosticness in Normally distributed and three-level categorical variables when the outcome is binary, it would need to be tested on other types of variable-outcome combinations before concluding that it is suitable for this additional task.

We evaluate the strategy of using each method as a predictive variable importance score. This approach has advantages over implementing full subgroup identification algorithms in that it provides a variable ranking. We envision that this can be used by clinicians to filter out unimportant features, and subject the remaining few to medical expertise and scrutiny. Given the fact that clinical trials are often too small to test for variable-treatment interactions in a rigorous statistical setting, and given the risk of false positives that is associated with multiple hypothesis testing, we believe that a variable scoring and ranking approach, combined with clinician judgment, is a more sensible approach to conducting exploratory subgroup identification than a full subgroup identification procedure is. Thus, this pertains to another point on results interpretation which is to emphasise that, although Virtual Twins outperforms SIDES and GUIDE, it nevertheless is prone to inaccuracies in predictive feature ranking. Where the Virtual Twins method identifies predictive features, this on its own can only lead to tentative conclusions that they are actually present. Further validation is required to reliably conclude that the variables in question are predictive.

We have conducted this comparative study in an artificial environment. The Virtual Twins method was only evaluated on data sets with homogeneous features: continuous, binary and three-level categorical features in data_{VT}, data_{SIDES} and data_{GUIDE} respectively. In the clinical trial setting, there will typically be a variety of predictor variable types in the data set. Loh et al. [11] establish that the Virtual Twins method has substantial selection bias towards selecting variables with fewer possible splits, so that, for example, it favours binary over continuous features. Strobl et al. [33] confirm the existence of this bias more generally in the random forest algorithm that is influenced by the number of categories and scale of measurement of the predictive features. They propose an unbiased variable selection method to be implemented in each of the individual trees that make up the forest. Bias is overcome by using conditional inference trees [34] and sampling without replacement instead of bootstrapping. Conditional inference trees use separate criterion to select the feature and the split on that feature, which would amount to using a key idea of the GUIDE algorithm in the Virtual Twins method.

An additional artificiality is that we use simulated, not real, clinical trial data sets because the latter were unobtainable for the study. Foster et al. [9] did test the Virtual

Twins method on a real data set, but found no subgroups present. A better validation of the model would be to test predictive feature ranking performance on a real data set with known subgroups present. Thus, results must be interpreted in light of this restriction.

The critique of not having an adequate real data set is not only applicable to the Virtual Twins method. This is a problem for most exploratory subgroup identification algorithms: they can't be benchmarked on a common, easily obtainable data set. There are hurdles involved in making a clinical trial data set available. However, it is worth overcoming these in order to encourage more robust algorithm testing. It was observed that for each method, the data set on which it was evaluated in the original paper may have been chosen to show off the best parts of the algorithm and mask its weaknesses. The only two predictive features of the Virtual Twins data set were also prognostic. And both SIDES and GUIDE performed much better on their respective simulated data sets than they did on the others. It signifies the need for a systematic approach to method evaluation, which would involve analysing all proposed methods on benchmark data sets, much like the widespread practice in the machine learning community of evaluating new image processing methods on the MNIST data set.

Since Su et al. [6] first proposed Interaction Trees, there has been a steadily increasing amount of research conducted in the area of exploratory subgroup identification. Yet widespread adaptation of these methods in clinical trials has not followed in the wake of these developments. One probable reason for this is a fear that subgroup identification will be used to misrepresent the therapeutic capabilities of a new treatment. This might explain the reluctance amongst practitioners to switch from traditional subgroup analysis, which requires subgroups of interest to be specified *before* looking at the data, to exploratory post-hoc analysis. Critics cite the increased risk of false positives as a reason to avoid testing for many subgroups. However, because there are many ways of mitigating against the increased risk of multiplicity, these objections seem unreasonable. It is possible that attitudes will shift as policy in this area modernises to incorporate the new goals of personalised medicine.

The guidelines on statistical principals for clinical trials that were recommended by the ICH (International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use) working group for adoption by

the regulatory bodies of the EU, USA, and Japan [35] state that exploratory analyses "should be interpreted cautiously; any conclusion of treatment efficacy (or lack thereof) or safety based solely on exploratory subgroup analayses are unlikely to be accepted." This conservative attitude may have disincentivised exploratory analysis.

However, recent guidelines give greater consideration to exploratory subgroup identification and the language is couched in less reluctant terms. The European Medicines Agency recommends a cautious use of post-hoc subgroup investigation, particularly when there is a significant degree of heterogeneity in the study population [7]. "There is a tension therefore between the widely appreciated statistical phenomenon related to multiplicity and the issues [...] relating to the potential heterogeneity of a target population and potential heterogeneity of response to treatment. Despite the statistical limitations, not investigating, or ignoring results of subgroups could also lead to incorrect decisions." Now that computers have the power to implement exploratory subgroup identification procedures in a way that was not conceivable for much of the history of randomised controlled clinical trials, there is a moral imperative to try to extract more insight from the data.

Bibliography

[1] L. M. Friedman, C. D. Furberg, and D. L. Demets, *Fundamentals of clinical trials*. Springer International Publishing, 2010.

[2] K. Strimbu and J. a. Tavel, "What are Biomarkers?," *Curr Opin HIV AIDS*, vol. 5, no. 6, pp. 463–466, 2011.

[3] K. V. Ballman, "Biomarker: Predictive or prognostic?," *Journal of Clinical Oncology*, vol. 33, no. 33, pp. 3968–3971, 2015.

[4] I. Lipkovich, A. Dmitrienko, and R. B. D'Agostino, "Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials," *Statistics in Medicine*, vol. 36, no. 1, pp. 136–196, 2017.

[5] J. Tanniou, I. V. D. Tweel, S. Teerenstra, and K. C. B. Roes, "Subgroup analyses in confirmatory clinical trials : time to be specific about their purposes," *BMC Medical Research Methodology*, pp. 1–15, 2016.

[6] X. Su, D. M. Nickerson, C.-L. Tsai, H. Wang, and B. Li, "Subgroup analysis via recursive partitioning," *Journal of Machine Learning Research*, vol. 10, pp. 141–158, 2009.

[7] "Guideline on the investigation of subgroups in confirmatory clinical trials," Tech. Rep. January, European Medicines Agency, 2014.

[8] A. Dmitrienko, C. Muysers, A. Fritsch, and I. Lipkovich, "General guidance on exploratory and confirmatory subgroup analysis in late-stage clinical trials," *Journal of Biopharmaceutical Statistics*, vol. 26, no. 1, pp. 71–98, 2016.

[9] J. C. Foster, J. M. G. Taylor, and S. J. Ruberg, "Subgroup identification from randomized clinical trial data," *Statistics in Medicine*, vol. 30, no. 24, pp. 2867–2880, 2011.

[10] I. Lipkovich and A. Dmitrienko, "Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using SIDES," *Journal of Biopharmaceutical Statistics*, vol. 24, no. 1, pp. 130–153, 2014.

[11] W.-Y. Loh, X. He, and M. Man, "A regression tree approach to identifying subgroups with differential treatmenteffects.," *Statistics in medicine*, vol. 34, pp. 1818–33, may 2015.

[12] A. Negassa, A. Ciampi, M. Abrahamowicz, S. Shapiro, and J.-F. Boivin, "Tree-structured subgroup analysis for censored survival data: Validation of computationally inexpensive model selection criteria," *Statistics and Computing*, vol. 15, no. 3, pp. 231–239, 2005.

[13] X. Su, T. Zhou, X. Yan, J. Fan, and S. Yang, "Interaction Trees with Censored Survival Data," *The International Journal of Biostatistics*, vol. 4, no. 1, p. Article 2, 2008.

[14] W. Y. Loh, H. Fu, M. Man, V. Champion, and M. Yu, "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables," *Statistics in Medicine*, vol. 35, pp. 4837–4855, 2016.

[15] S. J. Pocock, S. E. Assmann, L. E. Enos, and L. E. Kasten, "Subgroup analysis, covariate adjustment and baseline comparisons in clinical trial reporting: current practiceand problems," *Statistics in Medicine*, vol. 21, pp. 2917–2930, oct 2002.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.

[17] T. Zou, Hui, Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society*, vol. 67, no. 2, pp. 301–320, 2005.

[18] J. O. Berger, X. Wang, and L. Shen, "A Bayesian approach to subgroup identification.," *Journal of biopharmaceutical statistics*, vol. 24, no. 1, pp. 110–29, 2014.

[19] X. Wang, J. O. Berger, D. Banks, M. A. Clyde, and D. S. Burdick, *Bayesian Modeling Using Latent Structures*. PhD thesis, Duke University, 2012.

[20] A. Zeileis, T. Hothorn, and K. Hornik, "Model-based recursive partitioning," *Journal of Computational and Graphical Statistics*, vol. 17, no. 2, pp. 492–514, 2008.

[21] E. Dusseldorp and I. Van Mechelen, "Qualitative interaction trees: A tool to identify qualitative treatment-subgroup interactions," *Statistics in Medicine*, vol. 33, no. 2, pp. 219–237, 2014.

[22] I. Lipkovich, A. Dmitrienko, J. Denne, and G. Enas, "Subgroup identification based on differential effect search–a recursive partitioning method for establishing response to treatment in patient subpopulations.," *Statistics in medicine*, vol. 30, pp. 2601–21, sep 2011.

[23] N. I. Friedman, Jerome H, Fisher, "Bump hunting in high-dimensional data," *Statistics and Comupting*, vol. 9, pp. 123–143, 1999.

[24] V. Kehl and K. Ulm, "Responder identification in clinical trials with censored data," *Computational Statistics & Data Analysis*, vol. 50, no. 5, pp. 1338–1355, 2006.

[25] E. B. Wilson and M. M. Hilferty, "The Distribution of Chi-Square," *Proceedings of the National Academy of Science of the USA*, vol. 17, pp. 684–688, 1931.

[26] W.-y. Loh, "Variable Selection for Classification and Regression in Large p , Small n Problems," *Probability Approximations and Beyond*, vol. 205, pp. 133–157, 2012.

[27] G. Hooker, "Discovering Additive Structure in Black Box Functions," in *Knowledge Discovery and Data Mining*, pp. 575–580, 2004.

[28] J. C. Foster, *Subgroup Identification and Variable Selection from Randomized Clinical Trial Data by*. PhD thesis, University of Michigan, 2013.

[29] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145–1159, jul 1997.

[30] L. Shen, Y. Ding, and C. Battioui, "A Framework of Statistical Methods for Identificaton of Subgroups with Differential Treatment Effects in Randomized Trials," in *Applied Statistics in Biomedicine and Clinical Trials Design*, ch. 25, pp. 411–425, Springer International Publishing Switzerland, 2015.

[31] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[32] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[33] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 30, pp. 927–961, aug 2007.

[34] T. Hothorn, K. Hornik, W. Wien, and A. Zeileis, "Unbiased Recursive Partitioning: A Conditional Inference Framework," *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, pp. 651–674, 2006.

[35] "Statistical principles for clinical trials," Tech. Rep. ICH E9, International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use, Geneva, 1998.