THE UNIVERSITY OF CHICAGO

NON-PARAMETRIC BAYESIAN INFERENCE WITH APPLICATION TO SYSTEM BIOLOGY

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF STATISTICS

BY
YI LIU

CHICAGO, ILLINOIS
JUNE 2022

To Shujun and Yutang:

It is with your encouragement and support that I manage to finish this Thesis. I dedicate this to you.

In the last five years, you learn more about others, but more importantly, you learn more about yourself.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to dedicate my thesis to my wife Shujun (Candy) Tang and my little baby (Yutang) who is still to be born. Your encouragement helps me finish this thesis. My sincere thanks also goes to my parents who supported me throughout these years. You are always there for me.

Next, I would like thank my advisors Professor Veronika Rockova and Professor John Reinitz whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. Here, I would also like to give thanks to Professor Sanz-Alonso and Professor Chen who gives me guidance in terms of what to pursue.

In addition, I would like to thank my collaborators Yuexi Wang and Kenneth Barr who worked with me throughout the projects. Finally I would also like to acknowledge all the interesting discussion from my colleges You-lin Chen, Xialiang Dou, Nathan Gill, Andrew Goldstein, Zhan Lin, Lizhen Nie, Abby Stevens, Jiacheng Wang, Daniel Xiang, and Qing Yan.

ABSTRACT

The issue of interpretability for machine learning algorithms is vital to the natural science community. If the output of machine learning procedure is fully interpretable, natural scientists can directly brought conclusions from machine learning algorithms applied to their experimental data. In this thesis, I use Bayesian machine learning techniques to develop two new variable selection methods and present two new applications to the control of biological transcription. With respect to these applications, I introduce a class of fully interpretable Deep Neural Networks (DNN) and a hierarchical Bayesian method for parameter inference.

The two variable selection algorithms make use of Approximate Bayesian Computation (ABC) for variable selection in the context of Bayesian forests. I proved that the spike-and-slab forest prior is consistent for model-free variable selection and proposed an ABC algorithm that efficiently samples from the posterior. I further improve the speed of this algorithm using Thompson sampling.

In the next part of the thesis, I consider interpretability in the context of biological control of transcription. I first introduce a fully interpretable DNN that implements a model for the control of transcription in *Drosophila*. This DNN has the unusual property that every layer is full interpretable in a sense that specifically implements a prior. I next consider a Bayesian method for parameter inference in a model of stochastic burst of transcription.

CHAPTER 1

INTRODUCTION

1.1 Interpretable learning in natural science

Machine learning is a set of structured methods to obtain information from data. Most machine learning methods are encoded in the form of an algorithm and the output is a model that can be used for prediction, inference, simulation, or visualization. Unlike scientific models, machine learning models make fewer assumptions and derive most of their knowledge from the data they train from. It is generally believed in machine learning community that with few restrictions and plentiful data, the model is going to be increasingly predictive and hence useful.

To the natural science community, machine learning is a good tool. It improves current experimental methods by replacing manual interpretation with automated methods [105, 252] or building better simulation algorithms that direct future experiments [131, 215]. However, one would expect that machine learning, with its astonishing successes in image recognition, game strategies, and natural language processing, would play a much bigger role in calculations in the natural sciences. In fact, the use of machine learning to directly extract scientific insight from experimental data is limited.

This situation is the consequence of a number of obstacles. One intrinsic weakness surrounding machine learning algorithms is that they attempt to harness all the predictive power from the training data rather than inferring a model from existing scientific principles. In contrast, models coming from scientific principles often starts off with well known scientific laws or phenomenological analogues and derive the functions using these fundamental principles. In contrast, with very few restrictions on how the model should be formulated, many machine learning models instead rely on complex mathematical structure so that they can fit any training data given to them. How the dependent variables are associated with independent variable is hidden within a mass of complexity. In fact, the internal structure

of machine learning model frequently does not follow existing scientific laws.

For example, Deep Neural Networks (DNNs) are built from layers of functions stacked together [65]. The output of one layer is fed to the next layer as input with no justification from independent scientific experiments. In fact, because of the complexity, a human cannot deduce how the eventual prediction is made from the data. It is difficult, if not impossible to map the structure of a deep neural network to the structure of domain knowledge except under very special conditions[147]. The model is therefore not interpretable and the reasons for its good empirical performance remain an enigma.

Another set of examples are Random Forests and Bayesian Additive Regression Trees (BART: Bayesian Trees). Both provide some level of interpretability through a measure of variable importance. This measure gives the relative contribution of each independent variable to the dependent variable, an obviously interpretable idea. However, variable importance is an ad-hoc measure. In fact, there is no formal mathematical definition of variable importance and there was a lack of theory on how variable importance will behave with noise in the data [148, 150] . This is one issue we will address in this thesis.

This thesis addresses the interpretability issue in four chapters tackling this problem from different perspectives. One common theme is Bayesian Machine Learning, i.e. machine learning algorithms based on Bayesian principles. Using Bayesian machine learning is advantageous in the sense that the errors associated with the data are taken into account very naturally in the form of the posterior distribution. In fact, the uncertainty of the associated with the parameter estimation is represented directly without the need to invoke the apparatus of the Delta method and Central Limit Theorem (CLT).

The outcome of Bayesian inference is an distribution of the parameters where one can make correlation studies, estimation of mean, covariance and even the presence of dependent variables in the final models. One can argue that the posterior distribution is infinite dimensional and we can ask harder questions of the experimental data. However, such convenience comes at a cost. With the exception of very ideal cases where one can denote the posterior

distribution in a closed form, one has to sample from the posterior distribution using costly Monte Carlo techniques. This computational cost will be discussed in detail in chapters 2, 3, and 5.

Below, we will consider these issue one by one: Concept of interpretability (section 1.2), Bayesian Inference (section 1.3), and gene regulation and transcriptional control (section 1.4)

1.2 Interpretability

Although a precise mathematical definition of interpretability is elusive, most would agree that an interpretable model can be mapped to a domain where humans can understand [178]. In fact, one can infer causality relationships between the dependent and independent variables through the model. For example, Hooke's law encodes an enormous family of physical phenomena connected with oscillators.

According to a survey by Lipton [146], interpretability in machine learning can be further broken down to Simulatability, Decomposability, and Algorithmic Transparency. Simulatability is defined as understanding over the entire model, i.e. that is there is a unified theoretical framework from which we can understand the model [65]. A model is Decomposible when it can be understood in terms of its components [65]. For instance, in machine learning, a tree model is Decomposible and can be seen as a modularized method in which each node has an explicit utility to judge if a discriminative condition is satisfied or not, each branch delivers an output of a judgement, and each leaf node represents the final decision after computing all attributes. Algorithmic Transparency is to understand the training process even under noisy data [65]. A generalized linear model is algorithmically transparent given that it is solving a convex optimization problem with a unique solution. In fact, the solution is known to converge to the underlying truth when the number of training data reaches infinity.

This thesis deals with interpretability in two ways: variable selection and parameter

estimation. Variable selection consists of determining if a independent variable has any relationship with the dependent variable without making any assumption on how to represent this relationship mathematically. The goal of the variable selection algorithm is to select a subset of independent variables for future studies. One can see variable selection as a coarse solution to the problem of Simulatability. Parameter estimation happens when the model is Simulatable and Decomposable. i.e. it is where one can formulate the relationship using equations and existing scientific principles but some of the key parameters are missing has to be inferred from experimental data. Here, we will focus on Algorithmic Transparency by coming up with different algorithms for the same problem. In this thesis, chapter 2 and 3 address variable selection and chapter 4 and 5 address parameter estimation.

As a practical application, in Chapter 4 we focus on the use case of transcriptional control. i.e. how DNA sequence interacts with transcription factors to activate or repress a gene. In the study of transcriptional control, one can construct a model using laws of natural science. One first build different modules using thermodynamic principles and phenomenological observations. Then these different modules are combined together to form a complex model which has good predictive power[119, 147]. Simulatability and Decomposability are present in this model but Algorithmic Transparency is absent. While some of the parameters can be determined via independent experiments, there are parameters which cannot be isolated and measured [118, 119]. Multiple methods can to be used for parameter estimation and I showed that stochastic gradient descent makes use of recent advances in DNN to give good solutions. In Chapter 5, I considered a parameter estimation example with Simulatability, Decompsability and Algorithmic Transparency by developing a non-parametric Bayesian method to estimate key parameters in transcriptional bursting .

1.3 Bayesian methods and computation

Bayesian inference has a long history and was the predominant statistical method prior to Fisher's introduction of the maximum likelihood estimator. In Bayesian inference,

one first considers a 'prior' on the estimators. This prior can be a well-known probability distribution such as Beta or Gaussian or a stochastic process such as a Dirichlet process or a Gaussian process. The goal of Bayesian inference is to determine the posterior distribution of parameters after observing the data and make use of the posterior distribution to make scientific and statistic statements about the results. Unlike the outcome of a optimization procedure, outcome from a Bayesian algorithm is a posterior distribution rather a simple point estimate.

One key advantage of Bayesian methods is that it can turn questions of Simulatability to questions of Algorithmic Transparency because it converts the problem of hypothesis testing into a problem of probability or false discovery rate estimation [200, 148, 150, 198, 59]. Using the estimators, one then make decisions based on theoretical optimality of each estimator [11, 59]. In Chapter 2 and 3, the problem of variable selection is reduced to an estimation of posterior inclusion probability. Using the median probability model cut-off [12], one then pick variables with inclusion probability higher than 0.5.

If the goal is estimation, one wants to first prove the consistency of the posterior distribution. Without using mathematical language, one can define consistency as the posterior distribution concentrating to the true underlying parameter as the number of samples approaches infinity. If the posterior distribution is consistent, then one has the assurance knowing that one will arrive at the right set of solutions. Therefore, in Chapter 2 and 5, we have started off with proving the consistency of our model prior to demonstrating its uses in variable selection and transcriptional bursting.

When it comes to Algorithmic Transparency, Bayesian Methods relies on algorithms that can sample from the posterior distribution. In special cases, one can directly induce the posterior distribution through symbolic manipulation. However, in general, posterior distributions are sampled through Monte Carlo simulations. A Monte Carlo simulator is a sample generator of the posterior distribution that generates posterior samples of parameters. To translate from the posterior samples to estimation of parameters, one derive the key

statistics from the posterior samples such as the mean, standard deviations and correlations. The posterior mean is generally used as the estimates of the parameter and the posterior standard deviation is used a measure of uncertainty. Since one have an infinite sampler of the posterior distribution, these statistics can be made infinitely close to the true value underpinning posterior distribution.

1.3.1 Priors

For users of Bayesian methods in natural science, one uncomfortable fact is the existence of prior distributions. One key argument against the prior distribution is that it feeds potentially biased information into the estimation process. In fact, the mean of posterior distribution with a proper prior—one with a well defined probability distribution—is known to be biased. Under mild conditions, as long as the model is consistent, the prior has very weak impact on the posterior distribution when the quantity of data is sufficiently large. In fact, the effect of bias is order of $1/n$ (where n is number of data points). Not only is this bias argument against the prior distribution weak, I would further argue that there are distinct advantages of having the prior distribution.

The first argument is philosophical. Subjective Bayesian is a school of thought addressing the meaning of priors [89]. It argues that the prior distributions encompasses all the information that one gathers about the model parameters prior to the statistical experiments. In this framework, the posterior distribution is an update on the information about the parameters. If posterior distribution is consistent, the update will get us closer to the truth as more data are collected in experiments. Such arguments stands well in natural science since it closely approximate the increment in knowledge from scientific experiments and at the same time also highlights the importance of consistency when making use of Bayesian methods.

The next argument comes from theoretical machine learning. Here, the prior is seen as a penalty factor on the parameters that prevents over-fitting, i.e. the model is so flexible

that it fits the noise in the training process. In Chapter 5, the number of parameters in the model can go to infinity. In this case, a maximum likelihood estimator will result in clear over-fitting of any experimental data. Without any constraint on the number of parameters, one can a designated sets of parameters for each data point (optimized to fit the value of the data point) since more parameters always increases the likelihood. The non-parametric Bayesian model that we propose in Chapter 5 is a natural way to restrict the number of parameters because it placed exponentially decreasing weights on that number.

1.3.2 Monte Carlo

In most cases, there is no closed form solution for the posterior distribution. As such, Monte Carlo is the usual way for Bayesian statisticians to compute the posterior distribution. Monte Carlo simulation provides a random deviate generator that generates samples from the posterior distribution. With the invention of Markov Chain Monte Carlo (MCMC), it is then possible to simulate from any distribution as long as we able to get a quick computation of the likelihood [5]. This sampling procedure allows posterior computation of a wide range of models, ranging from the hierarchical to the non-parametric. In Chapter 5, we proposed a variation of MCMC known as Block Gibbs sampling for sampling from the posterior distribution of a non-parametric process [82]. We leverage the fact that all the conditional distribution of each parameter that we try to estimate can be done in closed form even though the posterior distribution itself is stochastic process that is very difficult to describe.

However, MCMC is restricted only to the case where the likelihood is known and easy to compute. In Chapter 2, we showed that in non-parametric variable selection using Bayesian Forest, the likelihood is unknown. Here, a new approach known as Approximate Bayesian Computation (ABC) is introduced to give a useful approximation to the posterior distribution. [158] Another weakness of MCMC is the computation time linear in number of samples . This slows the computation time when we are dealing with a huge data set [5]. In Chapter

3, we introduce a machine learning technique known as Thompson Sampling to approximate the posterior. One needs to only look at a small portion of the data at each iteration. This new technique improves the speed of sampling for large data sets. We prove the theoretical efficiency of our method and demonstrate its empirical performance is simulations and real data study.

1.4 Gene regulation and transcriptional control

One interesting use case is in the study of transcriptional control and gene regulation. This problem asks for the use of machine learning methods because it is complex, noisy and difficult to describe. An existing model of this phenomenon [109, 119, 14] was organised in layers suggesting a DNN, but expressed in mathematics bearing little resembles to that used in neural networks. We show a mathematical mapping in Chapter 4 of this model to DNN and demonstrate that the training can be done through a modern stochastic gradient descent method known as Adaptive Moment Estimation (Adam). This stochastic gradient descent method gives a great algorithmic speed up over the previously method of simulated annealling. This is an interesting example where a difficult problem not only shed lights into the interpretability of DNN, but can also make use of current technologies in machine learning to improve its performance.

Extending this use case, in Chapter 5 we develop a method that infers the number of states and associated parameters in the bursting transcription from inter-burst times derived from live imaging data. Here, the waiting times between successive binding events of RNA polymerase II (PolII) are known to come from a mixed exponential distribution with unknown number of states. Since the number of states is unknown, we use a non-parametric Bayesian method for parameter inference to avoid overfitting. Our method takes full advantage of Bayesian inference because we output estimators with standard errors and associated probabilities. Furthermore, we prove that this method is consistent in a probabilistic sense and show that it can perform well in simulations and real data experiments.

1.5 Outline of the thesis

In summary, in Chapter 2, we introduce a new Approximate Bayesian Computation (ABC) for variable selection. In Chapter 3, we improve the ABC variable selection algorithm through the use of Thompson sampling. Chapter 4 considers the transcriptional problem with DNN. Finally, Chapter 5 introduce a non-parametric Bayesian method for inference. We discuss the impact of these results in interpretability in Chapter 6.

CHAPTER 2

APPROXIMATE BAYESIAN COMPUTATION FOR NON-PARAMETRIC INFERENCE

2.1 Abstract

Few problems in statistics are as perplexing as variable selection in the presence of very many redundant covariates. The variable selection problem is most familiar in parametric environments such as the linear model or additive variants thereof. In this work, we abandon the linear model framework, which can be quite detrimental when the covariates impact the outcome in a non-linear way, and turn to tree-based methods for variable selection. Such variable screening is traditionally done by pruning down large trees or by ranking variables based on some importance measure. Despite heavily used in practice, these ad-hoc selection rules are not yet well understood from a theoretical point of view. In this work, we devise a Bayesian tree-based probabilistic method and show that it is consistent for variable selection when the regression surface is a smooth mix of $p > n$ covariates. These results are the first model selection consistency results for Bayesian forest priors. Probabilistic assessment of variable importance is made feasible by a spike-and-slab wrapper around sum-of-trees priors. Sampling from posterior distributions over trees is inherently very difficult. As an alternative to MCMC, we propose ABC Bayesian Forests, a new ABC sampling method based on data-splitting that achieves higher ABC acceptance rate. We show that the method is robust and successful at finding variables with high marginal inclusion probabilities. Our ABC algorithm provides a new avenue towards approximating the median probability model in non-parametric setups where the marginal likelihood is intractable.

This chapter is published as a paper in Journal of Royal Statistical Society [149] with joint authorship with Veronika Rockova and Yuexi Wang. I designed the ABC algorithm and conducted part of the simulation and data analysis. Yuexi Wang designed the MCMC algorithms for spike-and-slab forest and conducted some of the simulations.

2.2 Introduction

In its simplest form, variable selection is most often carried out in the context of linear regression [233, 83, 66]. However, confinement to linear parametric forms can be quite detrimental for variable importance screening, when the covariates impact the outcome in a non-linear way [235]. Rather than first selecting a parametric model to filter out variables, another strategy is to first select variables and then build a model. Adopting this reversed point of view, we focus on developing methodology for the so called "model-free" variable selection [45].

There is a long strand of literature on the fundamental problem of non-parametric variable selection. One line of research focuses on capturing non-linearities and interactions with basis expansions and performing grouped shrinkage/selection on sets of coefficients [212, 190, 142, 188]. [128] propose the RODEO method for sparse non-parametric function estimation through regularization of the derivative expectation operator and provide a consistency result for the selection of the optimal bandwidth. [33] propose a model-free knock-off procedure, controlling FDR in settings when the conditional distribution of the response is arbitrary. In the Bayesian literature, [210] deploy spike-and-slab priors on covariance parameters of Gaussian processes to erase variables. In this work, we focus on other non-parametric regression techniques, namely trees/forests which have been ubiquitous throughout machine learning and statistics [24, 48]. The question we wish to address is whether one can leverage the flexibility of regression trees for effective (consistent) variable importance screening.

While trees are routinely deployed for data exploration, prediction and causal inference [100, 226, 91], they have also been used for dimension reduction and variable selection. This is traditionally done by pruning out variables or by ranking them based on some importance measure. The notion of variable importance was originally proposed for CART using overall improvement in node impurity involving surrogate predictors [26]. In random forests, for example, the importance measure consists of a difference between prediction errors before and after noising the covariate through a permutation in the out-of-bag sample. However,

this continuous variable importance measure is on an arbitrary scale, rendering variable selection ultimately ad-hoc. Principled selection of the importance threshold (with theoretical guarantees such as FDR control or model selection consistency) is still an open problem. Simplified variants of importance measures have begun to be understood theoretically for variable selection only very recently [106, 117].

Bayesian trees and forests select variables based on probabilistic considerations. The BART procedure [48] can be adapted for variable selection by forcing the number of available splits (trees) to be small, thereby introducing competition between predictors. BART then keeps track of predictor inclusion frequencies and outputs a probabilistic importance measure: an average proportion of all splitting rules inside a tree ensemble that split on a given variable, where the average is taken over the MCMC samples. This measure cannot be directly interpreted as the posterior variable inclusion probability in anisotropic regression surfaces, where wigglier directions require more splits. [21] consider a permutation framework for obtaining the null distribution of the importance weights. [254] implement reinforcement learning for selection of splitting variables during tree construction to encourage splits on fewer more important variables. All these developments point to the fact that regularization is key to enhancing performance of trees/forests in high dimensions. Our approach differs in that we impose regularization from *outside* the tree/forest through a spike-and-slab wrapper.

Spike-and-slab variable selection consistency results have relied on analytical tractability (approximation availability) of the marginal likelihood [169, 115, 40]. Nicely tractable marginal likelihoods are ultimately unavailable in our framework, rendering the majority of the existing theoretical tools inapplicable. For these contexts, [245] characterized general conditions for model selection consistency, extending the work of [134] to non *iid* setting. Exploiting these developments, we show variable selection consistency of our non-parametric spike-and-slab approach when the regression function is a smooth mix of covariates. Building on [202], our paper continues the investigation of missing theoretical properties of Bayesian CART and BART. We show model selection consistency when the smoothness is known as

well as joint consistency for both the regularity level *and* active variable set when the smoothness is not known and when $p > n$. These results are the first model selection consistency results for Bayesian forest priors.

The absence of a tractable marginal likelihood complicates not only theoretical analysis, but also computation. We turn to Approximate Bayesian Computation (ABC) [184, 158, 53] and propose a procedure for model-free variable selection. Our ABC method *does not* require the use of low-dimensional summary statistics and, as such, it *does not* suffer from the known difficulty of ABC model choice [196]. Our method is based on sample splitting where at each iteration (a) a random subset of data is used to come up with a proposal draw and (b) the rest of the data is used for ABC acceptance. This new data-splitting approach increases ABC effectiveness by increasing its acceptance rate. ABC Bayesian forests relate to the recent line of work on combining machine learning with ABC [187, 113]. We propose dynamic plots that describe the evolution of marginal inclusion probabilities as a function of the ABC selection threshold.

The paper is structured as follows. Section 2 introduces the spike-and-slab wrapper around tree priors. Section 3 develops the ABC variable selection algorithm. Section 4 presents model selection consistency results. Section 5 demonstrates the usefulness of the ABC method on simulated data and Section 6 wraps up with a discussion.

2.2.1 Notation

With $\|\cdot\|_n$ we denote the empirical L^2 norm. The class of functions $f(\boldsymbol{x}) : [0,1]^p \to \mathbb{R}$ such that $f(\cdot)$ is constant in all directions excluding $\mathcal{S}_0 \subseteq \{1,\ldots,p\}$ is denoted with $\mathcal{C}(\mathcal{S}_0)$. With $\mathcal{H}_p^\alpha$, we denote α-Hölder continuous functions with a smoothness coefficient α. $a \lesssim b$ denotes a is less or equal to b, up to a multiplicative positive constant, and $a \asymp b$ denotes $a \lesssim b$ and $b \lesssim a$. The ε-covering number of a set Ω for a semimetric d, denoted by $N(\varepsilon;\ \Omega;\ d)$, is the minimal number of d-balls of radius ε needed to cover set Ω.

2.3 Variable selection with Bayesian forest

We will work within the purview of non-parametric regression, where a vector of continuous responses $\mathbf{Y}^{(n)} = (Y_1, \ldots, Y_n)'$ is linked to fixed (rescaled) predictors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})' \in [0,1]^p$ for $1 \leq i \leq n$ through

$$Y_i = f_0(\mathbf{x}_i) + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \quad \text{for} \quad 1 \leq i \leq n, \tag{2.1}$$

where $f_0(\cdot)$ is the regression mixing function and $\sigma^2 > 0$ is a scalar. It is often reasonable to expect that only a small subset $\mathcal{S}_0$ of $q_0 =\mid \mathcal{S}_0 \mid$ predictors actually exert influence on $\mathbf{Y}^{(n)}$ and contribute to the mix. The subset $\mathcal{S}_0$ is seldom known with certainty and we are faced with the problem of variable selection. Throughout this paper, we assume that the regression surface is smoothly varying (α-Hölder continuous) along the active directions $\mathcal{S}_0$ and constant otherwise, i.e. we write $f_0 \in \mathcal{H}_p^{\alpha} \cap \mathcal{C}(\mathcal{S}_0)$.

Unlike linear models that capture the effect of a single covariate with a single coefficient, we permit non-linearities/interactions and capture variable importance with (additive) regression trees. By doing so, we hope to recover non-linear signals that could be otherwise missed by linear variable selection techniques.

As with any other non-parametric regression method, regression trees are vulnerable to the curse of dimensionality, where prediction performance deteriorates dramatically as the number of variables p increases. If an oracle were to isolate the active covariates $\mathcal{S}_0$, the fastest achievable estimation rate would be $n^{-\alpha/(2\alpha+|\mathcal{S}_0|)}$. This rate depends only on the intrinsic dimensionality $q_0 =\mid \mathcal{S}_0 \mid$, not the actual dimensionality p which can be much larger than n. Recently, [202] showed that with *suitable regularization*, the posterior distribution for Bayesian CART and BART actually concentrates at this fast rate (up to a log factor), adapting to the intrinsic dimensionality and smoothness.

Later in Section 2.5, we continue their theoretical investigation and focus on consistent *variable selection*, i.e. estimation of $\mathcal{S}_0$ rather than $f_0(\cdot)$. Spike-and-slab regularization plays

a key role in obtaining these theoretical guarantees. Many applications offer a plethora of predictors and some form of redundancy penalization has to be incurred to cope with the curse of dimensionality. Bayesian regression trees were originally conceived for prediction rather than variable selection. Indeed, original tree implementations of Bayesian CART [56, 47] do not seem to penalize inclusion of redundant variables aggressively enough. As noted by [143], the prior expected number of active variables under the Bayesian CART prior of [47] satisfies $\lim_{p\to\infty} \mathbb{E}[q] = K-1$ as $p \to \infty$ where K is the fixed number of bottom leaves. This behavior suggests that (in the limit) the prior forces inclusion of the maximal number of variables while splitting on them only once. This is far from ideal. To alleviate this issue, we deploy the so-called *spike-and-forest priors*, i.e. spike-and-slab wrappers around sum-of-trees priors [202]. As with the traditional spike-and-slab priors, the specification starts with a prior distribution over the 2^p active variable sets:

$$\mathcal{S} \sim \pi(\mathcal{S}) \quad \text{for each} \quad \mathcal{S} \subseteq \{1, \ldots, p\}. \tag{2.2}$$

We elaborate on the specific choices of $\pi(\mathcal{S})$ later in Section 2.4.2 and Section 2.5.

Given the pool of variables $\mathcal{S}$, a regression tree/forest is grown using *only* variables inside $\mathcal{S}$. This prevents the trees from using too many variables and thereby from overfitting. Recall that each individual regression tree is characterized by two components: (1) a tree-shaped K-partition of $[0,1]^p$, denoted with $\mathcal{T}$, and (2) bottom node parameters (step heights), denoted with $\boldsymbol{\beta} \in \mathbb{R}^K$. Starting with a parent node $[0,1]^p$, each K-partition is grown by recursively dissecting rectangular cells at chosen internal nodes along one of the active coordinate axes, all the way down to K terminal nodes. Each tree-shaped K-partition $\mathcal{T} = \{\Omega_k\}_{k=1}^K$ consists of K partitioning rectangles $\Omega_k \subset [0,1]^p$.

While Bayesian CART approximates $f_0(\boldsymbol{x})$ with a single tree mappings $f_{\mathcal{T},\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{k=1}^K \mathbb{I}(\boldsymbol{x} \in \Omega_k)\beta_k$, Bayesian Additive Regression Trees (BART) use an aggregate of T

mappings

$$f_{\mathcal{E},\boldsymbol{B}}(\boldsymbol{x}) = \sum_{t=1}^{T} f_{\mathcal{T}^t,\boldsymbol{\beta}^t}(\boldsymbol{x})$$

where $\mathcal{E} = \{\mathcal{T}^1, \ldots, \mathcal{T}^T\}$ is an ensemble of tree partitions and $\boldsymbol{B} = [\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^T]$ is an ensemble of step coefficients. In a fully Bayesian approach, prior distributions have to be specified over the set of tree structures $\mathcal{E}$ and over terminal node heights $\boldsymbol{B}$. The spike-and-forest construction can accommodate various tree prior options.

To assign a prior over $\mathcal{E}$ for a given T, one possibility is to first pick the number of bottom nodes, independently for each tree, from a prior

$$K^t \sim \pi(K) \quad \text{for} \quad K = 1, \ldots, n, \tag{2.3}$$

such as the Poisson distribution [56]. Given the vector of tree sizes $\boldsymbol{K} = (K^1, \ldots, K^T)'$ and a set of covariates $\mathcal{S}$, we assign a prior over so-called valid ensembles/forests $\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}$. We say that a tree ensemble $\mathcal{E}$ is valid if it consists of trees that have non-empty bottom leaves. One can pick a tree partition ensemble from a uniform prior over *valid* forests $\mathcal{E} \in \mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}$, i.e.

$$\pi(\mathcal{E} \mid \mathcal{S}, \boldsymbol{K}) = \frac{1}{\Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}})} \mathbb{I}\left(\mathcal{E} \in \mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}\right), \tag{2.4}$$

where $\Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}})$ is the number of valid tree ensembles characterized by $\boldsymbol{K}$ bottom leaves and split directions $\mathcal{S}$. The prior (2.3) and (2.4) was deployed in the Bayesian CART implementation of [56] (with $T = 1$) and it was studied theoretically by [202]. Another related Bayesian forest prior (implemented in the BART procedure and studied theoretically by [206] consists of an independent product of branching process priors (one for each tree) with decaying split probabilities [47]. The implementation is very similar to the one of [56].

Finally, given the partitions $\mathcal{T}^t$ of size K^t for $1 \leq t \leq T$, one assigns (independently for

each tree) a Gaussian product prior on the step heights

$$\pi(\boldsymbol{\beta}^t \mid K^t) = \prod_{k=1}^{K^t} \phi(\beta_k^t;\ \sigma^2_{\boldsymbol{\beta}}), \tag{2.5}$$

where $\phi(x;\ \sigma^2_{\boldsymbol{\beta}})$ denotes a Gaussian density with mean zero and variance $\sigma^2_{\boldsymbol{\beta}} = 1/T$ (as suggested by [48]). The prior for σ^2 can be chosen as inverse chi-squared with hyperparameters chosen based on an estimate of the residual standard deviation of the data [48].

The most crucial component in the spike-and-forest construction, which sets it apart from existing BART implementations, is the active set $\mathcal{S}$ which serves to mute variables by restricting the pool of predictors available for splits. The goal is to learn which set $\mathcal{S}$ is most likely (a posteriori) and/or how likely each variables is to have contributed to f_0. Unlike related tree-based variable selection criteria, the spike-and-slab envelope makes it possible to perform variable selection directly by evaluating posterior model probabilities $\Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$ or marginal inclusion probabilities $\Pi(j \in \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)})$ for $1 \leq j \leq p$. Random forests [24] also mute variables, but they do so from within the tree by randomly choosing a small subset of variables for each split. The spike-and-slab approach mutes variables externally rather than internally. [21] note that when the number of trees is small, the Gibbs sampler for BART can get trapped in local modes which can destabilize the estimation procedure. On the other hand, when the number of trees is large, there are ample opportunities for the noise variables to enter the model without necessarily impacting the model fit, making variable selection very challenging. Our spike-and-slab wrapper is devised to get around this problem.

The problem of variable selection is fundamentally challenged by the sheer size of possible variable subsets. For linear regression, (a) MCMC implementations exist that capitalize on the availability of marginal likelihood [169, 94], (b) optimization strategies exist for both continuous [201, 205] and point-mass spike-and slab priors [35]. These techniques do not directly translate to tree models, for which tractable marginal likelihoods $\pi(\boldsymbol{Y}^{(n)} \mid \mathcal{S})$ are unavailable. To address this computational challenge, we explore ABC techniques as a new

promising avenue for non-parametric spike-and-slab methods.

2.4 ABC-sampling algorithm

Performing (approximate) posterior inference in complex models is often complicated by the analytical intractability of the marginal likelihood. Approximate Bayesian Computation (ABC) is a simulation-based inference framework that obviates the need to compute the likelihood directly by evaluating the proximity of (sufficient statistics of) observed data and pseudo-data simulated from the likelihood. Simon Tavaré first proposed the ABC algorithm for posterior inference [229] in the 1990's and since then it has widely been used in population genetics, systems biology, epidemiology and phylogeography.

Combined with a probabilistic structure over models, marginal likelihoods give rise to posterior model probabilities, a standard tool for Bayesian model choice. When the marginal likelihood is unavailable (our case here), ABC offers a unique computational solution. However, as pointed out by [196], ABC cannot be trusted for model comparisons when model-wise sufficient summary statistics are not sufficient across models. The ABC approximation to Bayes factors then does not converge to exact Bayes factors, rendering ABC model choice fundamentally untrustworthy. A fresh new perspective to ABC model choice was offered in [187], who rephrase model selection as a classification problem that can be tackled with machine learning tools. Their idea is to treat the ABC reference table (consisting of samples from a prior model distribution and high-dimensional vectors of summary statistics of pseudo-data obtained from the prior predictive distribution) as an actual data set, and to train a random forest classifier that predicts a model label using the summary statistics as predictors.

Their goal is to produce a stable model decision based on a classifier rather than on an estimate of posterior model probabilities. Our approach has a similar flavor in the sense that it combines machine learning with ABC, but the concept is fundamentally very different. Here, the fusion of Bayesian forests and ABC is tailored to non-parametric variable selection

towards obtaining posterior variable inclusion probabilities. Our model selection approach does not suffer from the difficulty of ABC model choice as we *do not* commit to any summary statistics and use random subsets of observations to generate the ABC reference table.

2.4.1 Naive ABC Implementation

For its practical implementation, our Bayesian variable selection method requires sampling from the analytically intractable posterior distribution over subsets $\Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$ under the *spike-and-forest* prior (2.4), (2.3) and (2.2). Given a single tree partition $\mathcal{T}$, the (conditional) marginal likelihood $\pi(\boldsymbol{Y}^{(n)} \mid \mathcal{T}, \mathcal{S})$ is available in closed form, facilitating implementations of Metropolis-Hastings algorithms [47, 56] (see Section A.3). However, such MCMC schemes can suffer from poor mixing. Taking advantage of the fact that, despite being intractable, one can *simulate from* the marginal likelihood $\pi(\boldsymbol{Y}^{(n)} \mid \mathcal{S})$, we will explore the potential of ABC as a complementary development to MCMC implementations.

The principle at the core of ABC is to perform approximate posterior inference from a given dataset by simulating from a prior distribution and by comparisons with numerous synthetic datasets. In its standard form, an ABC implementation of model choice creates a reference table, recording a large number of datasets simulated from the model prior and the prior predictive distribution under each model. Here, the table consists of M pairs $(\mathcal{S}_m, \mathbf{Y}^\star_m)$ of model indices $\mathcal{S}_m$, simulated from the prior $\pi(\mathcal{S})$, and pseudo-data $\boldsymbol{Y}^\star_m \in \mathbb{R}^n$, simulated from the marginal likelihood $\pi(\boldsymbol{Y}^{(n)} \mid \mathcal{S}_m)$. To generate $\boldsymbol{Y}^\star_m$ in our setup, one can hierarchically decompose the marginal likelihood

$$\pi(\boldsymbol{Y}^{(n)} \mid \mathcal{S}) = \int_{(f_{\mathcal{E},\boldsymbol{B}},\sigma^2)} \pi(\boldsymbol{Y}^{(n)} \mid f_{\mathcal{E},\boldsymbol{B}}, \sigma^2) \mathrm{d}\, \pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}) \tag{2.6}$$

and first draw $(f^m_{\mathcal{E},\boldsymbol{B}}, \sigma^2_m)$ from the prior $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$ and obtain $\boldsymbol{Y}^\star_m$ from (2.1), given $(f^m_{\mathcal{E},\boldsymbol{B}}, \sigma^2_m)$. ABC sampling is then followed by an ABC rejection step, which extracts pairs $(\mathcal{S}_m, \boldsymbol{Y}^\star_m)$ such that $\boldsymbol{Y}^\star_m$ is close enough to the actual observed data. In other words, one

trims the reference table by keeping only model indices $\mathcal{S}_m$ paired with pseudo-observations that are at most ϵ-away from the observed data, i.e. $\|\boldsymbol{Y}^{obs} - \boldsymbol{Y}^{\star}_m\|_2 \leq \epsilon$ for some tolerance level ϵ. These extracted values comprise an approximate ABC sample from the posterior $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$, which should be informative for the relative ordering of the competing models, and thus variable selection. Note that this particular ABC implementation does not require any use of low-dimensional summary statistics, where rejection is based solely on $\boldsymbol{Y}^{obs}$. While theoretically justified, this ABC variant has two main drawbacks.

First, with very many predictors, it will be virtually impossible to sample from all 2^p model combinations at least once, unless the reference table is huge. Consequently, relative frequencies of occurrence of a model $\mathcal{S}_m$ in the trimmed ABC reference table *may not* be a good estimate of the posterior model probability $\pi(\mathcal{S}_m \mid \boldsymbol{Y}^{(n)})$.

While the model with the highest posterior probability $\pi(\mathcal{S}_m \mid \boldsymbol{Y}^{(n)})$ is commonly conceived as the right model choice, it may not be the optimal model for prediction. Indeed, in nested correlated designs and orthogonal designs, it is the median probability model that is predictive optimal [10]. The median probability model (MPM) consists of those variables whose *marginal* inclusion probabilities $\mathbb{P}(j \in \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)})$ are at least 0.5. While simulation-based estimates of posterior model probabilities $\mathbb{P}(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$ can be imprecise, we argue (and show) that ABC estimates of marginal inclusion probabilities $\mathbb{P}(j \in \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)})$ are far more robust and stable.

The second difficulty is purely computational and relates to the issue of coming up with good proposals $f^m_{\mathcal{E},\boldsymbol{B}}$ such that the pseudo-data are sufficiently close to $\boldsymbol{Y}^{obs}$. Due to the vastness of the tree ensemble space, it would be naive to think that one can obtain solid guesses of f_0 purely by sampling from non-informative priors. This is why we call this ABC implementation naive. These considerations lead us to a new data-splitting ABC modification that uses a random portion of the data to train the prior and to generate pseudo-data with more affinity to the left-out observations.

2.4.2 ABC Bayesian Forests

By sampling directly from noninformative priors over tree ensembles $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$, the acceptance rate of the naive ABC can be prohibitively small where huge reference tables would be required to obtain only a few approximate samples from the posterior. . To address this problem, we suggest a sample-splitting approach to come up with draws that are less likely to be rejected by the ABC method. At each ABC iteration, we first draw a random subsample $\mathcal{I} \subset \{1, \ldots, n\}$ of size $\mid \mathcal{I} \mid = s$ with no replacement. Then we split the observed data $\boldsymbol{Y}^{(n)}$ into two groups, denoted with $\boldsymbol{Y}^{(n)}_{\mathcal{I}}$ and $\boldsymbol{Y}^{(n)}_{\mathcal{I}^c}$, and instead of (2.6) we consider the marginal likelihood conditionally on $\boldsymbol{Y}^{(n)}_{\mathcal{I}}$

$$\pi(\boldsymbol{Y}^{(n)} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S}) = \int_{(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2)} \pi(\boldsymbol{Y}^{(n)}_{\mathcal{I}^c} \mid f_{\mathcal{E},\boldsymbol{B}}, \sigma^2) \mathrm{d}\, \pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}) \tag{2.7}$$

where

$$\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}) = \pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S}). \tag{2.8}$$

This simple decomposition unfolds new directions for ABC sampling based on data splitting. Instead of using all observations $\boldsymbol{Y}^{obs}$ to accept/reject each draw, we set aside a random subset of data $\boldsymbol{Y}^{obs}_{\mathcal{I}^c}$ for ABC rejection and use $\boldsymbol{Y}^{obs}_{\mathcal{I}}$ to "train the prior". The key observation is that the samples from the prior $\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$, i.e. the *posterior* $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S})$, will have seen a part of the data and will produce more realistic guesses of f_0. Such guesses are more likely to yield pseudo-data that match $\boldsymbol{Y}^{obs}_{\mathcal{I}^c}$ more closely, thereby increasing the acceptance rate of ABC sampling. Note that the acceptance step is based solely on the left-out sample $\boldsymbol{Y}^{obs}_{\mathcal{I}^c_m}$, not the entire data. Similarly as the naive ABC outlined in the previous section, we first sample the subset $\mathcal{S}$ from the prior $\pi(\mathcal{S})$ and then obtain draws from the conditional marginal likelihood under an updated prior $\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$. This corresponds to an ABC strategy for sampling from $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}^c})$ under the priors (2.2) and (2.8). As will be seen later, this posterior is effective for assessing variable importance. Moreover, if $\pi(\mathcal{S})$ is a good proxy for $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}})$ (when the training set is small relative to the ABC rejection

set), this ABC will produce approximate samples from the original target $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$.

The idea of using a portion of the data for training the prior and the rest for model selection goes back to at least [90]. The most common prescription for choosing training samples in Bayesian analysis is to convert improper priors into propers ones for meaningful model selection with Bayes factors [135, 173]. [16] advocated choosing the training set as small as possible subject to yielding proper posteriors (so called minimal training samples). [17] argue that data can vary widely in terms of their information content and the use of single minimal training samples can be inadequate/ suboptimal. Since there are many possible training samples, it is natural to average the resulting Bayes factors over the training samples in some fashion. While intrinsic Bayes factors [16] average Bayes factors over all possible minimal training samples, expected posterior priors [182] average the prior first. In particular, the empirical expected-posterior prior for model $\mathcal{S}$ [88, 182] writes as

$$\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}) = \frac{1}{L}\sum_{l=1}^{L} \pi_{\mathcal{I}_l}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}), \tag{2.9}$$

where $\pi_{\mathcal{I}_l}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$ was defined in (8) and where L is the number of all minimal training samples $\mathcal{I}_l$. The marginal likelihood under this prior can be then written as (equation (3.5) in [182]) $m(\boldsymbol{Y}^{(n)} \mid \mathcal{S}) = \frac{1}{L}\sum_{l=1}^{L} \pi(\boldsymbol{Y}^{(n)} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S})$, where $\pi(\boldsymbol{Y}^{(n)} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S})$ was defined in (7). Our ABC analysis with internal data splitting can be thus regarded as arising from the empirical expected posterior prior (2.9). While the motivation for using training samples in Bayesian analysis has been largely to make improper priors proper, here we use this idea in a different context to increase ABC acceptance rate.

The ABC Bayesian Forests algorithm is formally summarized in Table 1. It starts by splitting the dataset into two subsets at each (m^{th}) iteration: $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ for fitting and $\boldsymbol{Y}^{obs}_{\mathcal{I}^c_m}$ for ABC rejection. The algorithm then proceeds by sampling an active set $\mathcal{S}$ from $\pi(\mathcal{S})$. Using the spike-and-slab construction, one can draw Bernoulli indicators $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)'$ where $\mathbb{P}(\gamma_j = 1 \mid \theta) = \theta$ for some prior inclusion probability $\theta \in (0, 1)$ and set $\mathcal{S}_m = \{j : \gamma_j = 1\}$.

Algorithm 1 : ABC Bayesian Forests

Data: Data $(Y_i^{obs}, \boldsymbol{x}_i)$ for $1 \leq i \leq n$
Result: $\pi_j(\epsilon)$ for $1 \leq j \leq p$ where $\pi_j(\epsilon) = \widehat{\mathbb{P}}(j \in \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)})$
Set M: the number of ABC simulations; s: the subsample size; ϵ: the tolerance threshold; $m = 0$ the counter
while $m \leq M$ **do**
  (a) Split data $\boldsymbol{Y}^{obs}$ into $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ and $\boldsymbol{Y}^{obs}_{\mathcal{I}_m^c}$, where $\mathcal{I}_m \subset \{1, \ldots, n\}$ of size $\mid \mathcal{I}_m \mid = s$ is obtained by sampling with no replacement.
  (b) Pick a subset $\mathcal{S}_m$ from $\pi(\mathcal{S})$.
  (c) Sample $(f^m_{\mathcal{E},\boldsymbol{B}}, \sigma^2_m)$ from $\pi_{\mathcal{I}_m}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}_m) = \pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S}_m)$.
  (d) Generate pseudo-data $\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c}$ by sampling white noise $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_m)$ and setting $Y_i^{\star} = f^m_{\mathcal{E},\boldsymbol{B}}(\boldsymbol{x}_i) + \varepsilon_i$ for each $i \notin I_m$.
  (e) Compute discrepancy $\epsilon_m = \|\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c} - \boldsymbol{Y}^{obs}_{\mathcal{I}_m^c}\|_2$.
  if $\epsilon_m < \epsilon$ **then**
    Accept $(\mathcal{S}_m, f^m_{\mathcal{E},\boldsymbol{B}})$ and set $m = m + 1$
  else
    Reject $(\mathcal{S}_m, f^m_{\mathcal{E},\boldsymbol{B}})$ and set $m = m + 1$
  end
end
Compute $\pi_j(\epsilon)$ as the proportion of times j^{th} variable is used in the accepted $f^m_{\mathcal{E},\boldsymbol{B}}$'s.

When sparsity is anticipated, one can choose θ to be small or to arise from a beta prior $\mathcal{B}(a, b)$ for some $a > 0$ and $b > 0$ (yielding the beta-binomial prior). We discuss other suitable prior model choices in Section 2.5.

In the *(c) step* of ABC Bayesian Forests, one obtains a sample from the posterior of $(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2)$, given $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$. For this step, one can leverage existing implementations of Bayesian CART and BART (e.g. the `BART` R package of [164]). A single draw from the posterior is obtained after a sufficient burn-in. In this vein, one can view ABC Bayesian Forests as a computational envelope around BART to restrict the pool of available variables. The *(d) step* then consists of predicting the outcome $\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c}$ for left-out observations $\boldsymbol{x}_i$ using (2.1) for each $i \in \mathcal{I}_m^c$. The last step is ABC rejection based on the discrepancy between $\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c}$ and $\boldsymbol{Y}^{obs}_{\mathcal{I}_m^c}$.

For the computation of marginal inclusion probabilities $\pi_j(\epsilon)$, one could conceivably report the proportion of ABC accepted samples $\mathcal{S}_m$ that contain the j^{th} variable. However, $\mathcal{S}_m$ is a pool of *available* predictors and not all of them are necessarily used in $f^m_{\mathcal{E},\boldsymbol{B}}$. Thereby,

we report the proportion of ABC accepted samples $f^m_{\mathcal{E},\boldsymbol{B}}$ that use the j^{th} variable at least once, i.e.

$$\pi_j(\epsilon) = \frac{1}{M(\epsilon)} \sum_{m:\epsilon_m<\epsilon} \mathbb{I}(j \text{ used in } f^m_{\mathcal{E},\boldsymbol{B}}), \tag{2.10}$$

where $M(\epsilon)$ is the number of accepted ABC samples at ϵ. Each tree ensemble $f^m_{\mathcal{E},\boldsymbol{B}}$ thus performs its own variable selection by picking variables from $\mathcal{S}_m$ rather than from $\{1,\ldots,p\}$. Limiting the pool of predictors prevents from too many false positives. In addition, the inclusion probabilities (2.10) do use the training data $\boldsymbol{Y}^{(n)}_{\mathcal{I}}$ to shrink and update the subset $\mathcal{S}$ by leaving out covariates not picked by $f^m_{\mathcal{E},\boldsymbol{B}}$. In this way, the mechanism for selecting the subsets $\mathcal{S}$ is not strictly sampling from the prior $\pi(\mathcal{S})$ but it seizes the information in the training set $\mathcal{I}$. In this way, $\mathcal{S}_m$'s can be regarded as approximate samples from $\pi(\mathcal{S} \mid \boldsymbol{Y}^{obs})$. When $\mathcal{I} = \emptyset$, we recover the naive ABC as a special case.

Dynamic ABC

The estimates of marginal inclusion probabilities $\pi_j(\epsilon)$ obtained with ABC Bayesian Forests unavoidably depend on the level of approximation accuracy ϵ. The acceptance threshold ϵ can be difficult to determine in practice, because it has to accommodate random variation of data around f_0 as well as the error when approximating smooth surfaces f_0 with trees. As $\epsilon \to 0$, the approximations $\pi_j(\epsilon)$ will be more accurate, but the acceptance rate will be smaller. It is customary to pick ϵ as an empirical quantile of ϵ_m, keeping only the top few closest samples. Rather than choosing one value ϵ, we suggest a dynamic strategy by considering a sequence of decreasing values $\epsilon_N > \epsilon_{N-1} > \cdots > \epsilon_1 > 0$. By filtering out the ABC samples with stricter thresholds, we track the evolution of each $\pi_j(\epsilon)$ as ϵ gets smaller and smaller. This gives us a dynamic plot that is similar in spirit to the Spike-and-Slab LASSO [201] or EMVS [199] coefficient evolution plots. However, our plots depict approximations to posterior inclusion probabilities rather than coefficient magnitudes. Other strategies for selecting the threshold ϵ are discussed in [224, 158, 53].

2.4.3 ABC Bayesian Forests in Action

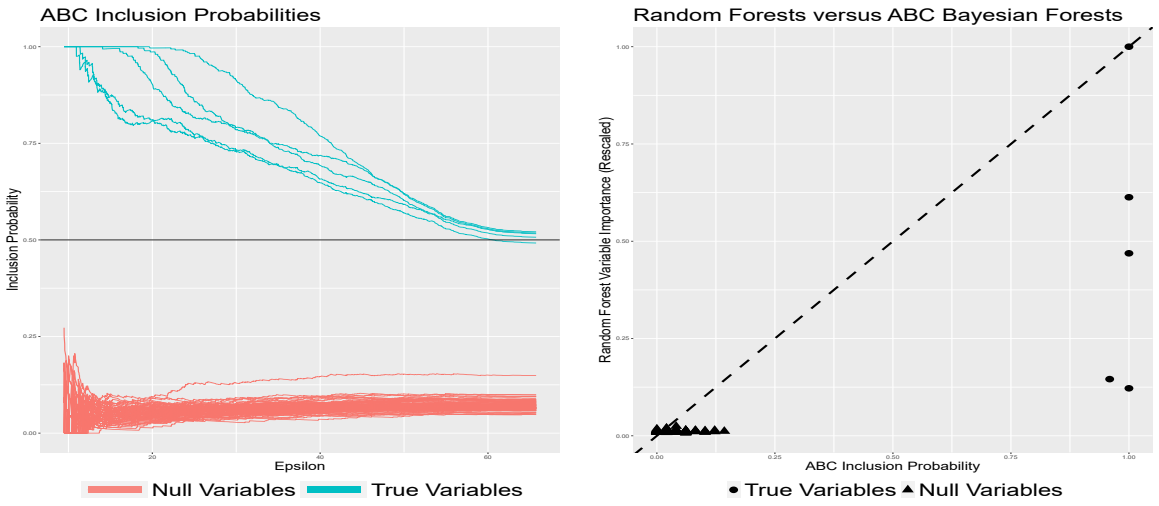
We demonstrate the usefulness of ABC Bayesian Forests on the benchmark Friedman dataset [76], where the observations are generated from (2.1) with $\sigma = 1$ and

$$f_0(\boldsymbol{x}_i) = 10\,\sin(\pi\, x_{i1}\, x_{i2}) + 20\,(x_{i3} - 0.5)^2 + 10\, x_{i4} + 5\, x_{i5}, \tag{2.11}$$

where $x_i \in [0,1]^p$ are *iid* from a uniform distribution on a unit cube. Because the outcome depends on $x_1, \ldots, x_p$, the predictors $x_6, \ldots, x_p$ are irrelevant, making it more challenging to find $f_0(\boldsymbol{x})$. We begin by illustrating the basic features of ABC Bayesian Forests with $p = 100$ and $n = 500$, assuming the beta-binomial prior $\pi(\mathcal{S} \mid \theta)$ with $\theta \sim \mathcal{B}(1,1)$ (see Section 2.4.2). At the m^{th} ABC iteration, we draw one posterior sample $f^m_{\mathcal{E},\boldsymbol{B}}$ after 100 burnin iterations using the BART MCMC algorithm [45] with $T = 10$ trees. We generate $M = 1\,000$ ABC samples (with $s = n/2$) and we keep track of variables used in $f^m_{\mathcal{E},\boldsymbol{B}}$'s to estimate the marginal posterior inclusion probabilities $\pi_j(\epsilon)$. It is worth pointing out that unlike MCMC, ABC Bayesian Forests are embarrassingly parallel, making distributed implementations readily available.

Following the dynamic ABC strategy, we plot the estimates of posterior inclusion indicators $\pi_j(\epsilon)$ as a function of ϵ (Figure 2.1). The true signals are depicted in blue, while the noise covariates are in red. The estimated inclusion probabilities clearly segregate the active and non-active variables, even for large ϵ values. This is because BART itself performs variable selection to some degree, where not all variables in $\mathcal{S}_m$ end up contributing to $f^m_{\mathcal{E},\boldsymbol{B}}$. For small enough ϵ, the inclusion probabilities of true signals eventually cross the 0.5 threshold. Based on the median probability model rule [10], one thereby selects the true model when ϵ is sufficiently small. Because the inclusion probabilities get a bit unstable as ϵ gets smaller (they are obtained from smaller reference tables), we excluded the 10 smallest ϵ values from the plot.

We repeated the experiment with more trees ($T = 50$) and a single tree ($T = 1$). Using



(a) $T = 10$ (b) Random Forests versus Bayesian Forests

Figure 2.1: (Left) Dynamic ABC plots for evolving inclusion probabilities as ϵ gets smaller. (Right) Plot of $\pi_j(\epsilon)$ obtained with ABC Bayesian Forests (ϵ is the 5% quantile of ϵ_m's) and the variable importance measure from Random Forests (rescaled to have a maximum at 1).

more trees, one still gets the separation between signal and noise. However, many more noisy covariates would be included by the MPM rule. This is in accordance with [45] who state that BART can over-select with many trees. With a single tree, on the other hand, one may miss some of the low-signal predictors, where deeper trees and more ABC iterations would be needed to obtain a clearer separation.

In this simulation, we observe a curious empirical connection between $\pi_j(\epsilon)$, obtained with ABC Bayesian Forests (taking top 5% ABC samples), and rescaled variable importances obtained with Random Forests (RF). From Figure 2.1(b), we see that the two measures largely agree, separating the signal coefficients (triangles) from the noise coefficients (dots). However, the RF measure is a bit more conservative, yielding smaller normalized importance scores for true signals. While variable importance for RF is yet not understood theoretically, in the next section we provide conditions under which the posterior distribution is consistent for variable selection.

2.5 Consistency of the posterior

In this section, we develop large sample model selection theory for spike-and-forest priors. As a jumping-off point, we first assume that α (the regularity of f_0) is known, where model selection essentially boils down to finding the active set $\mathcal{S}_0$. Later in this section, we investigate *joint* model selection consistency, acknowledging uncertainty about $\mathcal{S}_0$ and, at the same time, the regularity α.

Several consistency results for non-parametric regression already exist [254, 245]. [52] characterized tight conditions on (n, p, q_0), under which it is possible to consistently estimate the sparsity pattern in two regimes. For fixed q_0, consistency is attainable when $(\log p)/n \leq c$ for some $c > 0$. When q_0 tends to infinity as $n \to \infty$, consistency is achievable when $c_1 q_0 + \log\log(p/q_0) - \log n \leq c_2$ for some $c_1, c_2 > 0$. Throughout this section, we will treat q_0 as fixed and show variable selection consistency when $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$. As an overture to our main result, we start with a simpler case when $T = 1$ (a single tree) and when α is known. The full-fledged result for Bayesian forests and unknown α is presented in Section 2.5.3. Throughout this section, we will assume $\sigma^2 = 1$.

2.5.1 The Case of Known α

Spike-and-forest mixture priors are constructed in two steps by (1) first specifying a conditional prior $\Pi_{\mathcal{S}}(f)$ on tree (ensemble) functions expressing a qualitative guess on f_0, and then (2) attaching a prior weight $\pi(\mathcal{S})$ to each "model" (i.e. subset) $\mathcal{S}$. The posterior distribution $\Pi(f \mid \boldsymbol{Y}^{(n)})$ can be viewed as a mixture of individual posteriors for various models $\mathcal{S}$ with weights given by posterior model probabilities $\Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$, i.e.

$$\Pi(f \mid \boldsymbol{Y}^{(n)}) = \sum_{\mathcal{S}} \Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)}) \Pi_{\mathcal{S}}(f \mid \boldsymbol{Y}^{(n)}).$$

Our aim is to establish "model-free" variable selection consistency in the sense that

$$\Pi(\mathcal{S} = \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)}) \to 1 \quad \text{in } \mathbb{P}_{f_0}^{(n)}\text{-probability} \quad \text{as} \quad n \to \infty,$$

where $\mathbb{P}_{f_0}^{(n)}$ is the distribution of $\boldsymbol{Y}^{(n)}$ under (2.1). The adjective "model-free" merely refers to the fact that we are selecting subsets in a non-parametric regression environment without necessarily committing to a linear model. We start by defining the model index set $\boldsymbol{\Gamma} = \big\{\mathcal{S} : \mathcal{S} \subseteq \{1, \ldots, p\}\big\}$, consisting of all 2^p variable subsets, and we partition it into (a) the true model $\mathcal{S}_0$, (b) models that *overfit* $\boldsymbol{\Gamma}_{\mathcal{S} \supset \mathcal{S}_0}$ (i.e. supersets of the true subset $\mathcal{S}_0$) and (c) models that *underfit* $\boldsymbol{\Gamma}_{\mathcal{S} \not\supset \mathcal{S}_0}$ (i.e. models that miss at least one active covariate). Each model $\mathcal{S} \in \boldsymbol{\Gamma}$ is accompanied by a convergence rate $\varepsilon_{n,\mathcal{S}}$ that reflects the inherent difficulty of the estimation problem. For each model $\mathcal{S}$ of size $\mid \mathcal{S} \mid$, we define

$$\varepsilon_{n,\mathcal{S}} = C_\varepsilon \, n^{-\alpha/(2\alpha + |\mathcal{S}|)} \sqrt{\log n} \quad \text{for some} \quad C_\varepsilon > 0, \tag{2.12}$$

the $\|\cdot\|_n$-near-minimax rate of estimation of a $\mid \mathcal{S} \mid$-dimensional α-smooth function.

Prior Specification

Prior distribution on the model index $\Pi(\mathcal{S})$ has to be chosen carefully for model selection consistency to hold when $p > n$ [166]. Traditional spike-and-slab priors introduce $\Pi(\mathcal{S})$ through a prior inclusion probability $\theta = \Pi(i \in \mathcal{S}_0 \mid \theta)$, independently for each $i = 1, \ldots, p$. This prior mixing weight is often endowed with a prior, such as the uniform prior $\pi(\theta) = \mathcal{B}(1, 1)$ [213], yielding a uniform prior on the model size, or the "complexity prior" $\pi(\theta) = \mathcal{B}(1, p^c)$ for $c > 2$ [41], yielding an exponentially decaying prior on the model size. We propose a different approach, directly assigning a prior on model weights through

$$\pi(\mathcal{S}) \propto \mathbf{e}^{-C \left(n^{|\mathcal{S}|/(2\alpha + |\mathcal{S}|)} \log n \vee |\mathcal{S}| \log p \right)} \tag{2.13}$$

where $C > 0$ is a suitably large constant. When $\mid \mathcal{S} \mid \log p \leq n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)}$, this prior is proportional to $\mathbf{e}^{-C/C_\varepsilon^2\, n\, \varepsilon_{n,\mathcal{S}}^2}$ and, as such, it puts more mass on models that yield faster rates convergence (similarly as in Lember and van der Vaart (2007)). When $\mid \mathcal{S} \mid \log p > n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)} \log n$, the implied prior on the effective dimensionality $\pi(\mid \mathcal{S} \mid) = \binom{p}{|\mathcal{S}|}\pi(\mathcal{S})$ will be exponentially decaying in the sense that $\pi(\mid \mathcal{S} \mid) \lesssim \mathbf{e}^{-(C-1)|\mathcal{S}|\log p}$ for $C > 1$. It was recently noted by [39] that the complexity prior "penalizes slightly more than necessary". With our prior specification (2.13), however, the exponential decay kicks in *only* when $\mid \mathcal{S} \mid$ is sufficiently large.

Assuming that the level of smoothness α is known, the optimal number of steps (i.e. tree bottom leaves K) needed to achieve the rate-optimal performance for estimating f_0 should be of the order $n^{q_0/(2\alpha+q_0)} = 1/C_\varepsilon^2\, n\, \varepsilon_{n,\mathcal{S}_0}^2 / \log n$ [202]. For our toy setup with a known α, we thus assume a point-mass prior on K with an atom near the optimal number of steps for each given $\mathcal{S}$, i.e.

$$\pi(K \mid \mathcal{S}) = \mathbb{I}[K = K_{\mathcal{S}}], \quad \text{where} \quad K_{\mathcal{S}} = \lfloor C_K / C_\varepsilon^2\, n\, \varepsilon_{n,\mathcal{S}}^2 / \log n \rfloor \tag{2.14}$$

for some $C_K > 0$ such that $K_{\mathcal{S}_0} = 2^{q_0 s}$ for some $s \in \mathbb{N}$. In Section 2.5.2, we allow for more flexible trees with variable sizes.

Identifiability

The active variables ought to be sufficiently relevant in order to make their identification possible. To this end, we introduce a non-parametric signal strength assumption, making sure that f_0 is not too flat in active directions [245, 52].

We first introduce the notion of an approximation gap. For any given model $\mathcal{S}$, we denote with $\mathcal{F}_{\mathcal{S}}$ a set of approximating functions (only single trees $f_{\mathcal{T},\boldsymbol{\beta}}$ with $K_{\mathcal{S}}$ leaves for now)

and define the approximation gap as follows:

$$\delta_n^{\mathcal{S}} \equiv \inf_{f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}} \|f_0 - f_{\mathcal{T},\boldsymbol{\beta}}\|_n = \|f_0 - f^{\mathcal{S}}_{\mathcal{T},\widehat{\boldsymbol{\beta}}}\|_n, \tag{2.15}$$

where $f^{\mathcal{S}}_{\mathcal{T},\widehat{\boldsymbol{\beta}}}$ is the $\|\cdot\|_n$-projection of f_0 onto $\mathcal{F}_{\mathcal{S}}$. For identifiability of $\mathcal{S}_0$, we require that those models that miss one of the active covariates have a large separation gap.

Definition 2.5.1. *(Identifiability) We say that $\mathcal{S}_0$ is (f_0, ε)-identifiable if, for some $M > 0$,*

$$\inf_{i \in \mathcal{S}_0} \delta_n^{\mathcal{S}_0 \setminus i} > 2M\varepsilon. \tag{2.16}$$

We provide a more intuitive explanation of (2.16) in terms of directional variability of f_0. The best approximating tree $f^{\mathcal{S}}_{\mathcal{T},\widehat{\boldsymbol{\beta}}}$ can be written as

$$f^{\mathcal{S}}_{\mathcal{T},\widehat{\boldsymbol{\beta}}}(\boldsymbol{x}) = \sum_{k=1}^{K_{\mathcal{S}}} \mathbb{I}(\boldsymbol{x} \in \widehat{\Omega}_k^{\mathcal{S}}) \widehat{\beta}_k \text{ with } \widehat{\beta}_k = \bar{f}_0(\widehat{\Omega}_k^{\mathcal{S}}) \equiv \frac{1}{n(\widehat{\Omega}_k^{\mathcal{S}})} \sum_{\boldsymbol{x}_i \in \widehat{\Omega}_k^{\mathcal{S}}} f_0(\boldsymbol{x}_i),$$

where $\widehat{\mathcal{T}} = \{\widehat{\Omega}_k^{\mathcal{S}}\}_{k=1}^{K_{\mathcal{S}}}$ is the tree-shaped partition of the $\|\cdot\|_n$-projection of f_0 defined in (2.15) with $K_{\mathcal{S}}$ leaves and where $n(\widehat{\Omega}_k^{\mathcal{S}}) = \sum_{i=1}^n \mathbb{I}(\boldsymbol{x}_i \in \widehat{\Omega}_k^{\mathcal{S}}) \equiv n\, \mu(\widehat{\Omega}_k^{\mathcal{S}})$. The separation gap in (2.15) can be then re-written as

$$\delta_n^{\mathcal{S}} = \sqrt{\sum_{k=1}^{K_{\mathcal{S}}} \mu(\widehat{\Omega}_k^{\mathcal{S}}) V[f_0 \mid \widehat{\Omega}_k^{\mathcal{S}}]},$$

where

$$V[f_0 \mid \widehat{\Omega}_k^{\mathcal{S}}] \equiv \frac{1}{n(\widehat{\Omega}_k^{\mathcal{S}})} \sum_{\boldsymbol{x}_i \in \widehat{\Omega}_k^{\mathcal{S}}} \Big(f_0(\boldsymbol{x}_i) - \bar{f}_0(\widehat{\Omega}_k^{\mathcal{S}})\Big)^2$$

is the local variability of f_0 inside $\widehat{\Omega}_k^{\mathcal{S}}$. Given this characterization, (2.16) will be satisfied, for instance, when variability of f_0 inside best approximating cells that miss an active direction is too large, i.e. $\inf_{i \in \mathcal{S}_0} \inf_k V[f_0 \mid \widehat{\Omega}_k^{\mathcal{S}_0 \setminus i}] > 4M^2\, \varepsilon^2$.

Our identifiability condition is a theoretical assumption on f_0 which indicates how large signal in each direction should be in order to be capturable. It generalizes the more traditional sufficient "beta-min conditions" [40, 251] for variable selection consistency (see Remark 2.5.1). Here, we gauge the amount of signal in terms of local variation in cells that *do not split* on an active covariate. Intuitively, if we do not split on $i \in \mathcal{S}_0$, the "variation" of f_0 inside the cells of the best tree we can get without i will be too large. The following example links our identifiability assumption with beta-min conditions.

Example 2.5.1. *Assume for now that $p = 2$ and that f_0 is linear, i.e.*

$$f_0(\boldsymbol{x}_i) = a + bx_{i1} + cx_{i2}.$$

Moreover, assume that $n = 16$ predictor observations are located on a regular grid $\mathcal{X} = \{k/4 : 1 \leq k \leq 4\} \times \{j/4 : 1 \leq j \leq 4\}$, where $\times$ denotes the Cartesian product. Suppose $\mathcal{S}_0 = \{1, 2\}$ and set $\mathcal{S} = \mathcal{S}_0 \backslash \{2\} = \{1\}$ and $K_{\mathcal{S}} = 2$. It can be verified that the partition $\widehat{\mathcal{T}}$ of the best approximating tree that does not *split on the covariate x_2 consists of two rectangles $\widehat{\Omega}_1^{\mathcal{S}} = [0, 1/2) \times [0, 1]$ and $\widehat{\Omega}_2^{\mathcal{S}} = [1/2, 1] \times [0, 1]$. Then we have*

$$\bar{f}_0(\widehat{\Omega}_1^{\mathcal{S}}) = a + \frac{3}{2}\left(\frac{b}{4}\right) + \frac{5}{2}\left(\frac{c}{4}\right) \quad \textit{and} \quad \bar{f}_0(\widehat{\Omega}_2^{\mathcal{S}}) = a + \frac{7}{2}\left(\frac{b}{4}\right) + \frac{5}{2}\left(\frac{c}{4}\right)$$

and thereby

$$(\delta_m^{\mathcal{S}})^2 = V(f_0 \mid \widehat{\Omega}_1^{\mathcal{S}}) = V(f_0 \mid \widehat{\Omega}_2^{\mathcal{S}}) = \frac{1}{4}\frac{b^2}{16} + \frac{5}{4}\frac{c^2}{16}. \tag{2.17}$$

From the expression (2.17) we can immediately see the connection to the beta-min conditions. When the signal in the direction of x_2 is large enough, i.e. $c > 16/\sqrt{5}M\varepsilon$, our identifiability condition will be satisfied.

The second sufficient condition needed for methods such as the LASSO to fully recover $\mathcal{S}_0$ is "irrepresentability" [251, 238]. This condition restricts the amount of correlation between (active and non-active) covariates by imposing a regularization constraint on the magnitudes

of regression coefficients of the inactive predictors onto the active ones. Here, we generalize the notion of irrepresentability to the non-parametric setup. Consider an underfitting model $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \not\supset \mathcal{S}_0$, where $\mathcal{S}_1 \subset \mathcal{S}_0$ are true positives and $\mathcal{S}_2$ is a possibly empty set of false positives, i.e. $\mathcal{S}_2 \cap \mathcal{S}_0 = \emptyset$. We define

$$\rho_n^{\mathcal{S}} \equiv \frac{1}{n}\sum_{i=1}^{n}[f_0(\boldsymbol{x}_i) - f_{\mathcal{T},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}_1}(\boldsymbol{x}_i)][f_{\mathcal{T},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}(\boldsymbol{x}_i) - f_{\mathcal{T},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}_1}(\boldsymbol{x}_i)], \tag{2.18}$$

the sample covariance between the surplus signals in f_0 and $f_{\mathcal{T},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}$ obtained by removing the effect of $f_{\mathcal{T},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}_1}$. This quantity will be large if noise covariates inside $\mathcal{S}_2$ can compensate for the missed true covariates in $\mathcal{S}_0 \backslash \mathcal{S}_1$, i.e. when the true and fake covariates are strongly correlated. To obviate this substitution effect, we introduce the following nonparametric "irrepresentability"condition. Similarly as in [251], we require that "the total amount of an irrelevant covariate represented by the covariates in the true model" is small.

Definition 2.5.2. *(Irrepresentability) We say that ε-irrepresentability holds for f_0 and $\mathcal{S}_0$ if, for some $M > 0$, we have $\sup_{\mathcal{S} \not\supset \mathcal{S}_0} \mid \rho_n^{\mathcal{S}} \mid < \frac{M}{2}\varepsilon$, where $\rho_n^{\mathcal{S}}$ was defined in* (2.18).

It follows from Lemma A.1.2 (Appendix) that under the irrepresentability and identifiability conditions (Definition 2.5.1 and 2.5.2), we obtain

$$\inf_{\mathcal{S} \not\supset \mathcal{S}_0} \inf_{f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}} \|f_{\mathcal{T},\boldsymbol{\beta}} - f_0\|_n > M\,\varepsilon. \tag{2.19}$$

This condition essentially states that *all* models that miss *at least one* active covariate (i.e. not only subsets of the true model) have a large separation gap.

The following theorem characterizes variable selection consistency of spike-and-tree posterior distributions. Namely, the posterior distribution over the model index is shown to concentrate on the true model $\mathcal{S}_0$. One additional assumption is needed to make sure that the (fixed) design $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is sufficiently regular. [202] define the notion of a fixed $\mathcal{S}_0$-regular design in terms of cell diameters of a k-d tree partition (Definition 3.3). This

assumption essentially excludes outliers, making sure that the data cloud is spread evenly in active directions (while permitting correlation between covariates).

Theorem 2.5.1. *Assume* $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$ *for some* $\alpha \in (0,1]$ *and* $\mathcal{S}_0 \subset \{1,\dots,p\}$ *with* $q_0 =\mid \mathcal{S}_0 \mid$ *and* $\|f_0\|_\infty \lesssim B$. *Denote with* $\widetilde{\varepsilon}_n = C_\varepsilon\, n^{-\alpha/(2\alpha+q_n)}\sqrt{\log n}$, *where* $q_n = C_q\lceil n\,\varepsilon^2_{n,\mathcal{S}_0}/\log p\rceil$ *for some* $C_q > 0$, *and assume* $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$ *with* $2 \leq q_0 = \mathcal{O}(1)$ *as* $n \to \infty$. *Assume that (a)* $\mathcal{S}_0$ *is* $(f_0, \widetilde{\varepsilon}_n)$*-identifiable, (b)* $\widetilde{\varepsilon}_n$*-irrepresentability holds and that (c) the design* $\mathcal{X}$ *is* $\mathcal{S}_0$*-regular. Under the* spike-and-tree *prior comprising (with* $T = 1$*)* (2.4),(2.5),(2.13) *with* $C > 2$ *and* (2.14)*, we have*

$$\Pi[\mathcal{S} = \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)}] \to 1 \quad \textit{in } \mathbb{P}^{(n)}_{f_0}\textit{-probability as } n \to \infty.$$

Proof: Section A.1.1

Remark 2.5.1. *The assumption of* $(f_0, \widetilde{\varepsilon}_n)$*-identifiability pertains to the more traditional sufficient beta-min conditions for variable selection consistency in sparse high-dimensional models. For example, [40] in their Corollary 1 require that* $\min_{i\in\mathcal{S}_0} \mid \beta_i^0 \mid \geq M\sqrt{\frac{q_0 \log p}{n}}$, *for some "large enough constant"* $M > 0$ *that depends on the compatibility number (see e.g. Definition 2.1 in [40] of the design matrix* X *(rescaled to have an* $\|\cdot\|_2$ *norm* $\sqrt{n}$*). Our identifiability threshold also depends on the rate of convergence* ε_n *(similarly as in [40]). However, unlike in the linear models we measure the signal strength in a non-parametric way. Lastly, note that the identifiability gap* $\widetilde{\varepsilon}_n$ *in Theorem 2.5.1 is a bit larger than the near-minimax rate* $\varepsilon_{n,\mathcal{S}_0}$. *This requirement will be relaxed in the next section, where* α *will be treated as unknown.*

For *iid* models, [85] considered the problem of nonparametric Bayesian model selection and averaging and characterized conditions under which the posterior achieves adaptive rates of convergence. The authors also study the posterior distribution of the model index, showing that it puts a negligible weight on models that are bigger than the optimal one. [245] characterized similar conditions for the non-*iid* case, see Section A.1.1 for more details.

Remark 2.5.2. *(Theory for ABC) It is worth pointing out that Theorem 2.5.1 is obtained for the* actual *posterior* $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$*, not the ABC posterior. Theory for ABC recently started emerging with the first results focussing on ABC bias [9], consistency and asymptotic normality [160, 72, 73] and on convergence of the posterior mean [136]. For our non-parametric regression scenario, we can conclude (variable selection) consistency for ABC Bayesian forests under the assumption that the residual variance* σ^2 *decreases with the sample size (as is typical in the Gaussian sequence model). In particular, Theorem A.1.1 in Supplemental Materials (Section A.1.4) shows that the ABC posterior concentrates at the rate* $\lambda_n = 4\epsilon_n^T/3 + 1/\sqrt{n}$*, where* $\epsilon_n^T = \sqrt{2\log n/n}$ *is the ABC tolerance level. This result implies that the ABC posterior will* not *reward underfitting model as long as our identifiability and irrepresentability conditions are satisfied with* $\varepsilon = \lambda_n$*. Regarding over-fitting models, an ABC analogue of Lemma 1.1 (Section 1.1.2 in Supplemental Materials) implies that the ABC posterior probability of over-fitting models goes to zero, which concludes variable selection consistency of a (naive) ABC method. These considerations can be extended to ABC Bayesian Forests with data splitting using the empirical expected posterior prior justification in* (2.9)*. More details are in Supplemental Materials (Section A.1.4).*

Remark 2.5.3. *(Consistency of the Median Probability Model) In Section 2.4.3, we used the median probability model rule which may not the same as the highest-posterior model whose consistency we have shown in Theorem 2.5.1. However, even when* $p \to \infty$ *it can be verified (as in Corollary 4.1 in [169]) that the median probability model is* also *consistent under the same assumptions as Theorem 2.5.1. In particular,* $\mathbb{P}_{f_0}^{(n)}[\cap_{i=1}^{p} E_i] \to 1$ *as* $n \to \infty$ *where* $E_i = \{\Pi(\gamma_i = \gamma_i^0 \mid \boldsymbol{Y}^{(n)}) > 0.5\}$ *and where* $\gamma_i = \mathbb{I}(i \in \mathcal{S})$ *are binary inclusion indicators and* $\gamma_i^0 = \mathbb{I}(i \in \mathcal{S}_0)$*.*

2.5.2 The Case of Unknown α

The fact that the level α has to be known for the consistency to hold makes the result in Theorem 2.5.1 somewhat theoretical. In this section, we provide a joint consistency result

for the unknown regularity level K and, at the same time, the unknown subset $\mathcal{S}_0$. Finding the optimal regularity level K, given $\mathcal{S}_0$, is a model selection problem of independent interest [127]. Here, we acknowledge uncertainty about *both* K and $\mathcal{S}_0$ by assigning a joint prior distribution on $(K, \mathcal{S})$. Namely, we consider an analogue of (2.13), where $n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)}$ is now replaced with $K \log n$ (according to (2.14)), i.e.

$$\pi(K, \mathcal{S}) \propto \mathbf{e}^{-C(K \log n \vee |\mathcal{S}| \log p)} \quad \text{for} \quad 1 \leq K \leq n \quad \text{and} \quad \mathcal{S} \subseteq \{1, \ldots, p\}. \tag{2.20}$$

This prior penalizes models with too many splits or too many covariates. We now regard each model as a *pair of indices* $(K, \mathcal{S})$, where the "true" model is characterized by $\mathbf{\Gamma}_0 = (K_{\mathcal{S}_0}, \mathcal{S}_0)$ with $K_{\mathcal{S}_0}$ defined in (2.14). Again, we partition the model index set $\mathbf{\Gamma} = \{(K, \mathcal{S}) : \mathcal{S} \subseteq \{1, \ldots, p\}, 1 \leq K \leq n\}$ into (a) the true model $\mathbf{\Gamma}_0$, (b) models that underfit $\mathbf{\Gamma}_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{K < K_{\mathcal{S}_0}\}}$ (i.e. miss at least one covariate *or* use less than the optimal number of splits), and (c) models that overfit $\mathbf{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{K \geq K_{\mathcal{S}_0}\}}$ (i.e. use too many variables and splits).

We combine the identifiability and irrepresentability conditions into one as follows:

$$\inf_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{K < K_{\mathcal{S}_0}\}} \inf_{f_{\mathcal{T}, \boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}(K)} \|f_{\mathcal{T}, \boldsymbol{\beta}} - f_0\|_n > M \, \varepsilon_{n, \mathcal{S}_0} \tag{2.21}$$

for some $M > 1$, where $\mathcal{F}_{\mathcal{S}}(K)$ consists of all trees with K bottom leaves and splitting variables $\mathcal{S}$. This condition is an analogue of (2.19), essentially stating that one cannot approximate f_0 with an error smaller than a multiple of the near-minimax rate using underfitting models.

Theorem 2.5.2. *Assume $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$ for some $\alpha \in (0, 1]$ and $\mathcal{S}_0 \subset \{1, \ldots, p\}$ such that $|\ \mathcal{S}_0\ | = q_0$ and $\|f_0\|_\infty \lesssim B$. Assume $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$ and $2 \leq q_0 = \mathcal{O}(1)$ as $n \to \infty$. Furthermore, assume that the design $\mathcal{X}$ is $\mathcal{S}_0$-regular and that* (2.21) *holds. Under*

the spike-and-tree prior comprising (with $T = 1$) (2.4), (2.5) *and* (2.20) *for $C > 3$, we have*

$$\Pi\left[\{\mathcal{S} = \mathcal{S}_0\} \cap \{K_{\mathcal{S}_0} \leq K \leq K_n\} \mid \boldsymbol{Y}^{(n)}\right] \to 1 \quad \textit{in } \mathbb{P}_{f_0}^{(n)}\textit{-probability as } n \to \infty,$$

where $K_{\mathcal{S}_0}$ was defined in (2.14) *and $K_n = \lceil \bar{C}\, n\, \varepsilon^2_{n,\mathcal{S}_0} / \log n \rceil$ for some $\bar{C} > C_K / C^2_\varepsilon$.*

Proof: Section A.1.2

Note that both $K_{\mathcal{S}_0}$ and K_n are of the same (optimal) order, where the marginal posterior distribution $\Pi(K \mid \boldsymbol{Y}^{(n)})$ squeezes inside these two quantities as $n \to \infty$. [127] provide a similar result for their RODEO method, without the variable selection consistency part. [245] also provide a similar result for Gaussian processes, without the regularity selection consistency part. Here, we characterize *joint* consistency for both subset and regularity model selection.

2.5.3 Variable Selection Consistency with Bayesian Forests

Finally, we provide a variant of Theorem 2.5.2 for tree ensembles. Each Bayesian forest (i.e. additive regression tree) model is characterized by a triplet $(\mathcal{S}, T, \boldsymbol{K})$, where $\mathcal{S}$ is the active variable subset, $T \in \mathbb{N}$ is the number of trees and $\boldsymbol{K} = (K^1, \ldots, K^T)' \in \mathbb{N}^T$ is a vector of the bottom leave counts for the T trees. Rate-optimality of Bayesian forests can be achieved for a wide variety of priors, ranging from many weak learners (large T and small K^t's) to a few strong learners (small T and large K^t's) [202]. The optimality requirement is that the *total* number of leaves in the ensemble $\sum_{t=1}^T K^t$ behaves like $K_{\mathcal{S}_0}$, defined earlier in (2.14).

We thereby define models in terms of equivalence classes rather than individual triplets $(\mathcal{S}, T, \boldsymbol{K})$. We construct each equivalence class $E(Z)$ by combining ensembles with the same number Z of total leaves, i.e.

$$E(Z) = \bigcup_{T=1}^{\min\{Z,n\}} \left\{ \boldsymbol{K} \in \mathbb{N}^T : \sum_{t=1}^T K^t = Z \right\}. \tag{2.22}$$

The cardinality of $E(Z)$, denoted with $\Delta(E(Z))$, satisfies $\Delta(E(Z)) \leq Z!\, p(Z)$, where $p(Z)$ is the partitioning number (i.e. the number of ways one can write Z as a sum of positive integers). The "true" model $\boldsymbol{\Gamma}_0 = (\mathcal{S}_0, E(K_{\mathcal{S}_0}))$ consists of an equivalence class of forests that split on variables inside $\mathcal{S}_0$ with a total number of $K_{\mathcal{S}_0}$ leaves. Similarly as before, we define underfitting model classes $\boldsymbol{\Gamma}_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{E(Z): Z < K_{\mathcal{S}_0}\}}$ and overfitting model classes $\boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{E(Z): Z \geq K_{\mathcal{S}_0}\}}$. Regarding the prior on T, similarly as [202], we consider

$$\pi(T) \propto \mathbf{e}^{-C_T\, T}, \quad T = 1, \ldots, n, \quad \text{for} \quad C_T > 0. \tag{2.23}$$

Given T, we assign a joint prior over $\mathcal{S}_0$ and $\boldsymbol{K} \in \mathbb{N}^T$ as follows:

$$\pi(\mathcal{S}, \boldsymbol{K} \mid T) \propto \mathbf{e}^{-C\, \max\left\{|\mathcal{S}| \log p;\ \sum_{t=1}^{T} K^t \log n\right\}} \quad \text{for} \quad C > 1. \tag{2.24}$$

We conclude this section with a model selection consistency result for Bayesian forests under the following identifiability condition

$$\inf_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{E(Z): Z < K_{\mathcal{S}_0}\}} \inf_{f_{\mathcal{E},\boldsymbol{B}} \in \mathcal{F}_{\mathcal{S}}(\boldsymbol{K})} \|f_{\mathcal{E},\boldsymbol{B}} - f_0\|_n > M\, \varepsilon_{n,\mathcal{S}_0}, \tag{2.25}$$

where $\mathcal{F}_{\mathcal{S}}(\boldsymbol{K})$ denotes all forests $f_{\mathcal{E},\boldsymbol{B}}$ that split on variables $\mathcal{S}$ and consist of T trees with $\boldsymbol{K} = (K^1, \ldots, K^T)'$ bottom leaves.

Theorem 2.5.3. *Assume $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$ for some $\alpha \in (0, 1]$ and $\mathcal{S}_0 \subset \{1, \ldots, p\}$ such that $|\, \mathcal{S}_0 \,| = q_0$ and $\|f_0\|_\infty \lesssim B$. Assume $q_0 \log p \leq n^{q_0/(2\alpha + q_0)}$, where $2 \leq q_0 = \mathcal{O}(1)$ as $n \to \infty$. Furthermore, assume that the design is $\mathcal{S}_0$-regular and that (2.25) holds. Under the spike-and-forest prior comprising (2.4), (2.5), (2.23) and (2.24), we have*

$$\Pi\left[\{\mathcal{S} = \mathcal{S}_0\} \cap \left\{K_{\mathcal{S}_0} \leq \sum_{t=1}^{T} K^t \leq K_n\right\} \mid \boldsymbol{Y}^{(n)}\right] \to 1 \quad \textit{in } \mathbb{P}_{f_0}^{(n)}\textit{-probability as } n \to \infty,$$

where $K_{\mathcal{S}_0}$ was defined in (2.14) and $K_n = \lceil \bar{C}\, n\, \varepsilon_{n,\mathcal{S}}^2 / \log n \rceil$ for some $\bar{C} > C_K / C_\varepsilon^2$.

2.6 Simulation Study

We evaluate the performance of ABC Bayesian Forests on simulated data. We consider the following performance criteria: Precision $= 1 - \text{FDP} = \frac{\text{TP}}{\text{TP+FP}}$, Power $= \frac{\text{TP}}{\text{TP+FN}}$(defined as the proportion of true signals discovered as such), Hamming Distance (HD)= FP+FN (where FP and FN denotes the number of false positives and false negatives, respectively) and the area under the ROC curve (AUC). Traditionally, AUC assesses how well a classification method can differentiate between two classes in the absence of a clear decision boundary. We use this criterion to assess variable importance since many of the considered selection methods are based on an importance measure and, as such, do not have a clear decision boundary.

The synthetic data are generated from the model (2.1), where $\boldsymbol{x}_i$'s for $i = 1, \ldots, n$ are drawn independently from $N_p(0, \Sigma)$ with $\Sigma = (\rho_{ij})_{i,j=1}^{p,p}$. We make our comparisons under different combinations of f_0, σ and Σ. In particular, we consider a relatively large noise level with $\sigma = 5$ ($\sigma = \sqrt{5}$ for the linear setup) and

1. medium equi-correlation $\rho_{ij} = 0.5$ for $i \neq j$ with $\rho_{ii} = 1$,

2. high auto-correlation $\rho_{ij} = 0.9^{|i-j|}$.

Regarding the mean function f_0, we consider four choices: (1) a linear setup with $f_0(\boldsymbol{x}_i) = x_{i1} + 2x_{i2} + 3x_{i3} - 2x_{i4} - x_{i5}$; (2) the Friedman setup as described in (2.11); (3) a CART (tree-based) function $f_0(\boldsymbol{x}_i)$ generated from the first 5 covariates using the `rpart` function in R; (4) a simulated example from [138] (denoted with LLS hereafter) with $f_0(\boldsymbol{x}_i) = \frac{10x_{i2}}{1+x_{i1}^2} + 5\sin(x_{i3}x_{i4} + 2x_{i5})$. For the auto-correlation case, we permuted the covariates so that signals are not next to each other.

For each combination of settings, we repeat our simulation over 20 different datasets assuming $n = 500$ and $p \in \{100, 1\,000\}$. We compare ABC Bayesian Forests with Random Forests (RF), Dynamic Trees (DT) of [227], BART [48], DART of [143], LASSO and Spike-and-Forests (the MCMC counterpart of ABC Bayesian Forests outlined in Section A.3 of the

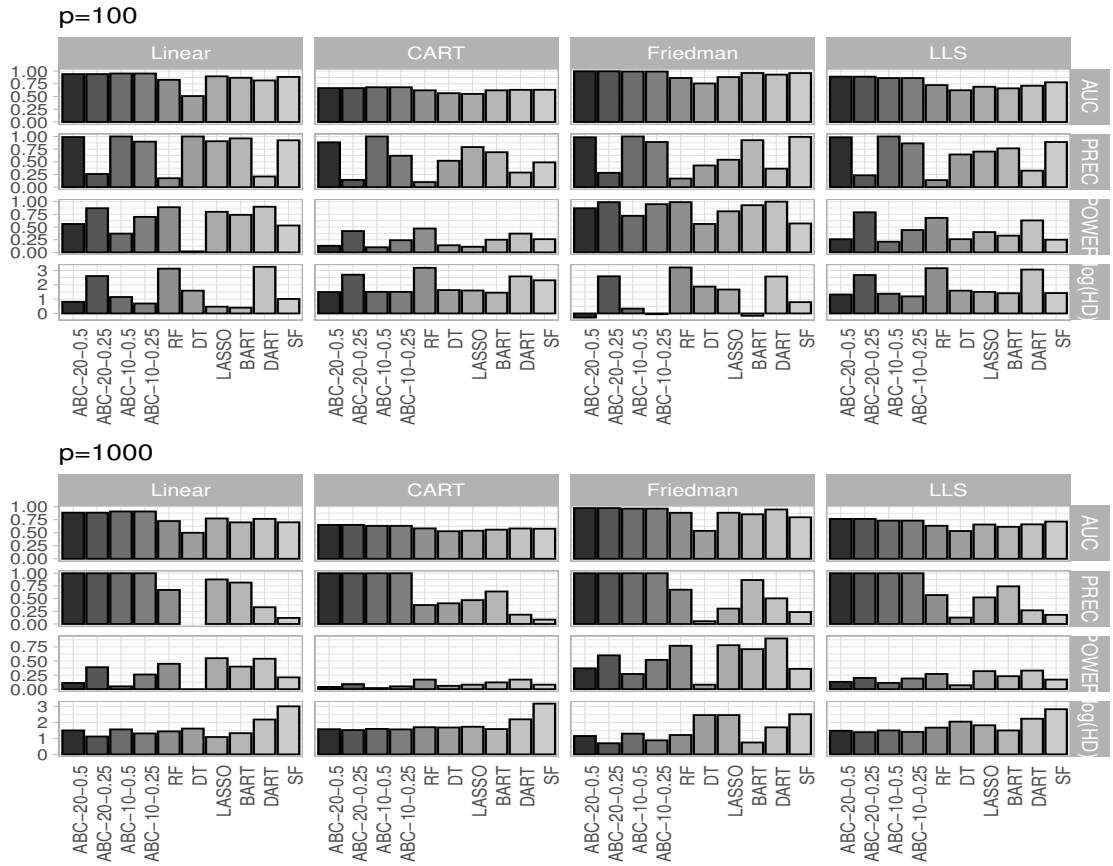


Figure 2.2: Average variable selection performance under equicorrelation $\rho_{ij} = 0.5$ over 20 simulations. Each panel corresponds to a different dimension $p \in \{100, 1000\}$. Each row reports a different statistic: AUC is the area under the ROC curve, PREC $= 1 - \text{FDP} = \frac{\text{TP}}{\text{TP+FP}}$, POWER $= \frac{\text{TP}}{\text{TP+FN}}$, $\log(\text{HD}) = \log(\text{FP} + \text{FN})$. ABC is run for $T \in \{10, 20\}$ and cutoff $\in \{0.5, 0.25\}$.Each column indicates a different data generating process.

Supplemental Materials). ABC Bayesian Forests are trained with $M = 1\,000$ ABC samples, where only a fraction of ABC samples (top 10%) are kept in the reference table. The prior $\pi(\mathcal{S})$ is the usual beta-binomial prior with $\theta \sim \mathcal{B}(1, 1)$. Inside each ABC step, we sample a subset of size $s = n/2$ and draw a tree ensemble using the default Bayesian CART prior [47] and $T \in \{10, 20\}$ trees. For each ABC sample, we draw the last BART sample after $B = 200$ burnin MCMC iterations. A sensitivity analysis to the choice s, T, B and M is reported in the Supplemental Materials (Section 4). Two versions of BART (without ABC) were deployed using the `R` package `BART`: (1) the standard BART from [48] with $T = 20$ (as recommended in [21]), and (2) the sparse version DART of [143] with a Dirichlet prior

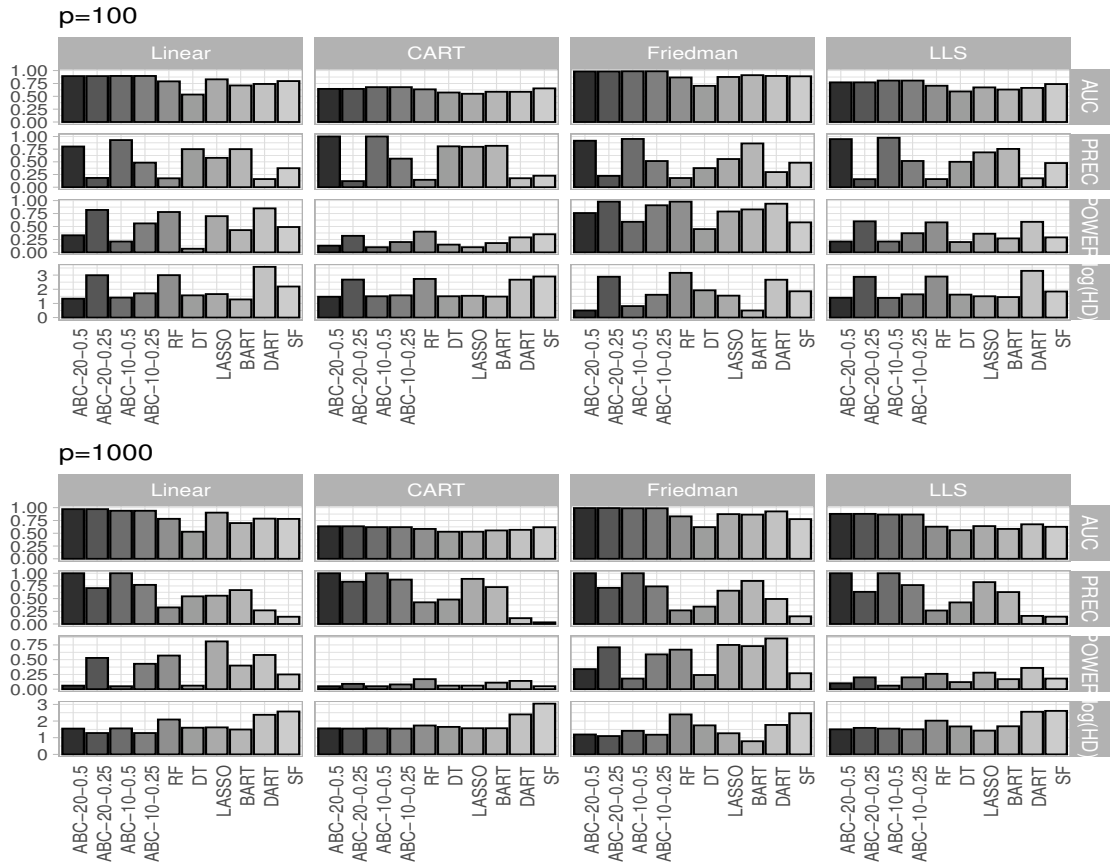


Figure 2.3: Average variable selection performance under autocorrelation $\rho_{ij} = 0.9^{|i-j|}$ over 10 simulations. Each panel corresponds to a different dimension $p \in \{100, 1000\}$. Each row reports a different statistics: AUC is the area under the ROC curve, PREC $= 1 - \text{FDP} = \frac{\text{TP}}{\text{TP+FP}}$, POWER $= \frac{\text{TP}}{\text{TP+FN}}$, $\log(\text{HD}) = \log(\text{FP} + \text{FN})$. ABC is run for $T \in \{10, 20\}$ and cutoff $\in \{0.5, 0.25\}$. Each column indicates a different data generating process.

(`sparse=TRUE, a=0.5, b=1`) with $T = 200$. Both versions are run with 10 000 MCMC samples after 10 000 burn-in. For LASSO, we use the `glmnet` package in `R` [75] using the 1-se rule to select the penalty λ. For Random Forests, we deploy the `randomForest` package in `R` [139] using the default number of 500 trees where variable importance is based on the difference in predictions (with and without each covariate) in out-of-bag samples.

To select variables with random forests, there are at least three commonly used strategies: (1) Recursive Feature Elimination (RFE) implemented in the `caret` package with 5-fold cross-validation (as suggested in [143]); (2) truncating importance at the $1 - \alpha$ quantile of a standard normal distribution (as suggested by [25]); (3) truncating importance at the

Bonferroni-corrected $(1 - \alpha/p)$ quantile of a standard normal distribution [21]. We report the third method, which was seen to perform the best. For BART and DART, we select those variables which have been split on inside a forest at least once on average. Alternative strategies based on truncating inclusion probabilities [143] using data-adaptive thresholds [21] did not perform better, in general. For ABC, we report results for two selection thresholds 0.5 and 0.25. For Spike-and-Forest (SF), we report the median probability model.

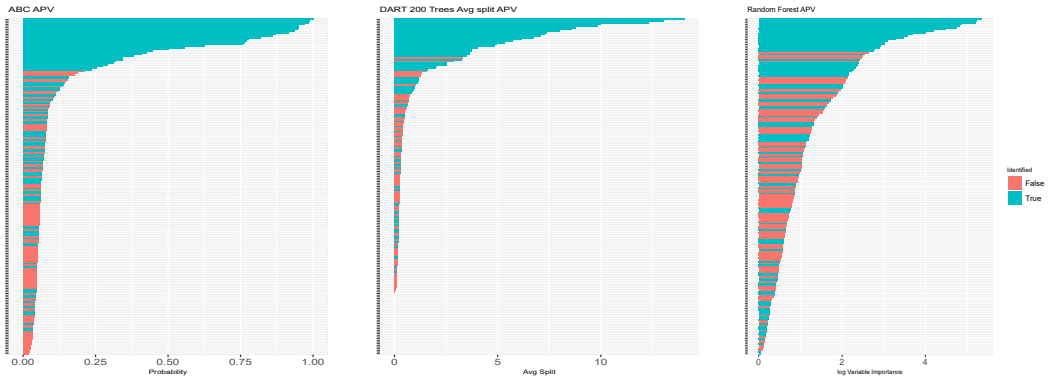
The performance comparisons for variable selection are summarized in Figure 2.2 (equicorrelation $\rho_{ij} = 0.5$) and Figure 2.3 (autocorrelation $\rho_{ij} = 0.9^{|i-j|}$). These figures show that ABC has an advantage in terms of AUC, suggesting that ABC can rank variables more efficiently. While RF tend to have a higher power, they are plagued with false discoveries (i.e. smaller precision). ABC Bayesian Forests, on the other hand, are seen to yield fewer false discoveries (i.e. higher precision) relative to the other procedures. The ABC threshold 0.5 yields higher precision whereas 0.25 yields higher power.

While ABC Bayesian Forests were designed to explore the posterior distribution over models, it is natural to ask whether they also yield reasonable prediction. There are various ways to perform prediction with our ABC method. One natural strategy is to save each draw $f^m_{\mathcal{S},\boldsymbol{B}}$ at the m^{th} ABC iteration when $\epsilon_m < \epsilon$ and average out individual predictions obtained from these single draws. Alternatively, one could first select variables based on ABC Bayesian Forests and then run a separate BART method (using the default number of $T = 200$ trees which is recommended for prediction) with the selected variables. Using both strategies, we report average out-of-sample mean squared prediction error, where the average is taken over 20 independent validation samples generated from the same data generating process (Table 2.1). We include both ABC predictions described above and denote them as ABC1 and ABC2, respectively, for the two different thresholds ($c \in \{0.5, 0.25\}$) and for the two choices of the number of trees ($T \in \{10, 20\}$).

The best method under each simulation setting is marked in bold. When the data becomes more non-linear (CART and LLS setups) and the correlation among variables gets

Table 2.1: Average out-of-sample mean squared prediction error over 20 independent validation datasets. ABC1 denotes predictions using ABC samples $f^m_{\mathcal{S},\boldsymbol{B}}$ and ABC2 uses ABC variable selection and runs BART ($T = 200$) on the selected subset. T designates the number of trees and c is the selection threshold. The best performing method for each row is denoted in bold.

	ABC2	**ABC1**	**ABC1**	**ABC2**	**ABC1**	**ABC1**	**RF**	**RLT**	**DT**	**BART**	**DART**
	$T = 20$	$T = 20, c = 0.5$	$T = 20, c = 0.25$	$T = 10$	$T = 10, c = 0.5$	$T = 10, c = 0.25$					
	Equi-correlation $\rho_{ij} = 0.5$ **for** $i \neq j$										
	Linear										
$p = 100$	5.56	5.58	5.84	5.60	5.84	5.55	5.63	5.45	5.92	5.49	**5.40**
$p = 1\,000$	5.79	6.15	5.73	5.86	6.28	5.95	5.83	5.70	6.04	5.82	**5.62**
	CART										
$p = 100$	34.21	34.63	37.19	**34.00**	36.10	35.81	34.21	34.64	34.61	35.48	35.57
$p = 1\,000$	32.00	34.27	35.72	**31.99**	33.93	33.17	32.30	32.40	33.08	33.77	34.04
	Friedman										
$p = 100$	30.32	29.28	31.59	30.52	30.30	**29.03**	31.84	30.17	41.41	31.31	**29.03**
$p = 1\,000$	33.14	35.97	31.54	33.54	38.42	32.71	34.35	32.22	45.69	32.99	**29.42**
	LLS										
$p = 100$	**26.23**	27.00	28.70	26.25	26.90	27.36	26.80	26.46	28.51	27.42	27.42
$p = 1\,000$	27.37	26.98	**26.94**	27.38	27.07	27.02	27.18	26.68	30.66	28.21	27.49
	Auto-correlation $\rho_{ij} = 0.9^{\lvert i-j \rvert}$										
	Linear										
$p = 100$	6.17	6.29	6.37	6.20	6.25	6.18	6.37	6.09	6.77	6.17	**5.91**
$p = 1\,000$	6.39	6.44	**6.00**	6.47	6.21	6.13	6.55	6.20	7.06	6.53	6.42
	CART										
$p = 100$	33.80	37.72	37.28	33.83	36.78	36.61	**33.57**	34.40	35.05	35.61	35.81
$p = 1\,000$	31.57	33.55	37.21	**31.52**	33.52	37.43	31.63	31.88	32.22	33.11	33.43
	Friedman										
$p = 100$	34.09	32.51	34.65	34.27	34.97	32.77	36.88	33.83	48.64	34.21	**30.36**
$p = 1\,000$	39.09	39.57	32.58	40.58	43.05	33.46	41.80	37.38	49.51	35.96	**30.81**
	LLS										
$p = 100$	28.57	**27.94**	30.71	28.45	28.03	29.12	28.88	27.87	30.69	28.83	28.81
$p = 1\,000$	29.98	**28.25**	28.96	30.14	28.40	28.38	30.19	28.56	32.29	31.76	29.28



(a) ABC (b) DART (c) RF

Figure 2.4: A barplot of ordered importance measures (inclusion probabilities for ABC, importance measures for DART and RF) for each of the $p = 201$ mutations for the drug APV, where blue represents mutations found in [194]. (a) Inclusion probabilities are computed using the top 1 000 out of $M = 10\,000$ ABC samples; (b) Average split of DART with 20 000 MCMC iterations; (c) log variable importance of Random Forest with 500 trees.

stronger, ABC tends to outperform the other methods. DART, on the other hand, works better for more linear datasets. Note that our default ABC implementation internally uses only a *small* number of $B = 200$ burn-in iterations and a small number of trees. For prediction, it has been recommended that BART is deployed with a larger number of trees [48]. In addition, the ABC computation produces forest samples $f^m_{\mathcal{S},B}$ which are from an *approximate* posterior. These two facts may affect resulting predictions which may not necessarily outperform BART (DART) across-the-board.

2.7 Performance on HIV-dataset

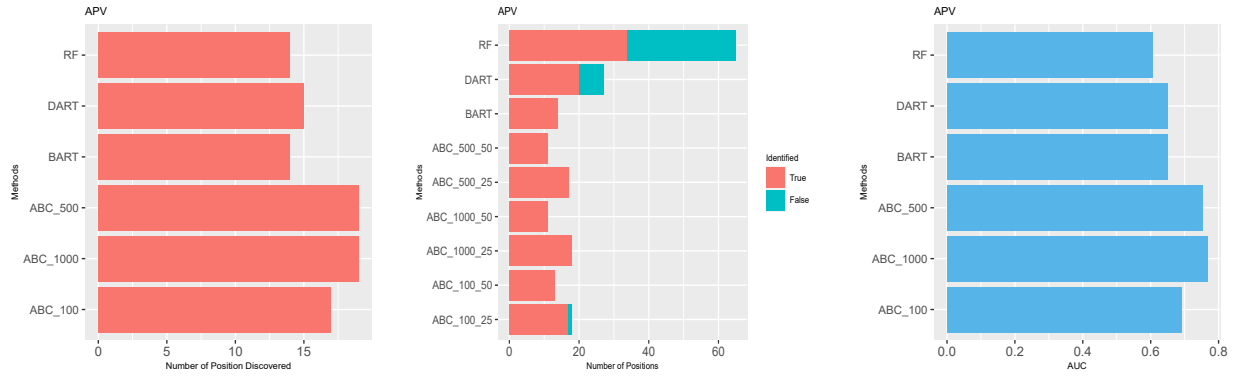
To further illustrate the usefulness of our approach, we consider a dataset described and analyzed in [195] and [8]. The data consists of genotype and resistance measurements (log-decrease in susceptibility) for three drug classes, i.e. protease inhibitors (PIs), nucleoside reverse transcriptase inhibitors (NRTIs) and non-nucleoside reverse transcriptase inhibitors (NNRTIs). The data is publicly available from the Stanford HIV Drug Resistance Database.

The goal of this analysis is to identify possible non-polymorphic mutation positions which

result in a log-fold increase of lab-tested drug resistance. The design matrix $X = (x_{ij})_{i,j=1}^{n,p}$ consists of binary indicators $x_{ij} \in \{0, 1\}$ for whether or not the j^{th} mutation occurred in the i^{th} sample. As in [8], only mutations that appear at least 3 times are taken into consideration. One appealing feature of this dataset is the availability of a proxy to the 'ground truth'. Indeed, in an independent experimental study, [194] identified mutations that are present at a significantly higher frequency in patients who have been treated with each drug. Similarly as [8], we treat this experimental data as an approximation to the truth for comparisons and for validation of our findings.

We run ABC with $M = 10\,000$ iterations, where each internal BART sample is obtained after 200 burnin iterations with 20 trees. The top 1 000 ABC samples with the smallest ϵ_m are kept and used to compute inclusion probabilities for each mutation. For illustration, we visualize results for one of the PI drugs (APV) and report the results for all the drugs in the Supplemental Material (Section A.5). The inclusion probabilities have been ordered and plotted in Figure 2.4, where the mutations experimentally validated by [194] (a proxy for true signals) are denoted in blue and the rest is in red. For comparisons, we also included the importance measure (the average number of splits on each variable) from DART run with 20 000 MCMC iterations and $T = 200$ trees as well as the importance measure (on a log scale) from Random Forests (RF) run with 500 trees.

Figure 2.4 reveals that ABC Bayesian Forests have a strong separation power, where experimentally validated mutations generally have a higher inclusion probability. Compared to DART and RF, ABC clearly stands out as being more effective in weeding out 'noise'. We gauge the strength of the signal/noise separation using several descriptive statistics. In these comparisons, we also consider plain BART method (using $T = 20$ trees and 20 000 MCMC iterations) and ABC using the top 100 and 500 samples with the smallest tolerance level ϵ_m. Since the selection of the cut-off point is not obvious for BART and RF, we first select variables based on an adaptive cut-off point so that there are no false discoveries (i.e. the cut-off is the largest importance weight of a *not* experimentally validated mutation). From



(a) Adaptive Cut-off (b) Automated Cut-off (c) AUC

Figure 2.5: (a) The number of true discoveries using an adaptive cut-off; (b) The number of true (red) and false (blue) discoveries using an automated cut-off; (c) The AUC of each method.

the plot of the number of 'True' locations selected (displayed in Figure 2.5(a)) we can see that all three ABC implementations find more signal variables. Next, we choose the cut-off point in an automated way, where ABC importance probabilities are truncated at 0.5 and 0.25, BART and DART measures are truncated at one (i.e. the variable has been used on average at least once), and RF select variables using recursive feature elimination as explain in the previous section. Similarly to [8], we report the number of 'True' locations and 'False' locations (Figure 2.5(b)). RF selection is plagued with false discoveries and DART is not free from false identifications either. The ABC selection cutoff 0.5 results in a more conservative selection, where lowering the cutoff point to 0.25 yields more discoveries. Finally, from the plot of the AUC values for all considered methods (Figure 2.5(c)), we conclude that ABC is better at separating the experimentally validated mutations from the rest even using a very few filtered ABC samples.

2.8 Discussion

This paper makes advancements at two fronts. One is the proposal of ABC Bayesian Forests for variable selection based on a new idea of data splitting, where a fraction of data is first

used for ABC proposal and the rest for ABC rejection. This new strategy increases ABC acceptance rate. We have shown that ABC Bayesian Forests are highly competitive with (and often better than) other tree-based variable selection procedures. The second development is theoretical and concerns consistency for variable and regularity selection. Continuing the theoretical investigation of BART by [202], we proposed new complexity priors which jointly penalize model dimensionality and tree size. We have shown joint consistency for variable *and* regularity selection when the level of smoothness is unknown and no greater than 1. Our results are the first model selection consistency results for BART priors.

Our ABC sampling routine has the potential to be extended in various ways. Sampling from $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S}_m)$ in ABC Bayesian Forests is one way of distilling $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ to propose a candidate ensemble $f^m_{\mathcal{E},\boldsymbol{B}}$. We noticed that the ABC acceptance rate can be further improved by replacing a randomly sampled tree with a fitted tree. Indeed, instead of drawing from $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S})$, one can *fit* a tree $\widehat{f}^m_{\mathcal{T},\boldsymbol{\beta}}$ to $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ using recursive partitioning algorithms (such as the `rpart` R package of [231] or with BART (by taking the posterior mean estimate $\widehat{f}^m_{\mathcal{E},\boldsymbol{B}} = \mathbb{E}[f_{\mathcal{E},\boldsymbol{B}} \mid \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S}]$). This variant, further referred to as ABC Forest Fit, is indirectly linked to other model-selection methods based on resampling.

[68] proposed a "first-order bootstrap" to assess confidence of an estimated tree phylogeny. The idea was to construct a tree from each bootstrap sample and record the proportion of bootstrap trees that have a feature of interest (for us, this would be variables used for splits). [60] embedded this approach within a parametric bootstrap framework, linking the bootstrap confidence level to both frequentist p-values and Bayesian a posteriori model probabilities. The authors proposed a second-order extension by reweighting the first-order resamples according to a simple importance sampling scheme. This second-order variant performs frequentist calibration of the a-posteriori probabilities and amounts to performing Bayesian analysis with Welch-Peers uninformative priors. [58] further develops the connection between parametric Bootstrap and posterior sampling through reweighting in exponential family models. Using non-parametric bootstrap ideas, [170] introduce the weighted

likelihood bootstrap (WLB) to sample from approximate posterior distributions. The WLB samples are obtained by maximum reweighted likelihood estimation with random weights. Such posterior sampling can be beneficial when, for instance, maximization is easier than Gibbs sampling from conditionals. In a similar spirit, our ABC Forest Fit variant would perform optimization (instead of sampling) on a random subset of the dataset to obtain a candidate tree/ensemble.

It is worth pointing out that $\widehat{f}^m_{\mathcal{E},\boldsymbol{B}}$ does not necessarily have to be a tree/forest. We suggest trees because they are are easily trainable and produce stable results using traditional software packages. In principle, however, this method could be deployed in tandem with other non-parametric methods, such as deep learning, to perform variable selection.

CHAPTER 3

COMPUTATIONAL SPEED-UPS USING BANDIT APPROACH

3.1 Abstract

Thompson sampling is a heuristic algorithm for the multi-armed bandit problem which has a long tradition in machine learning. The algorithm has a Bayesian spirit in the sense that it selects arms based on posterior samples of reward probabilities of each arm. By forging a connection between combinatorial binary bandits and spike-and-slab variable selection, we propose a stochastic optimization approach to subset selection called Thompson Variable Selection (TVS). TVS is a framework for interpretable machine learning which does not rely on the underlying model to be linear. TVS brings together Bayesian reinforcement and machine learning in order to extend the reach of Bayesian subset selection to non-parametric models and large datasets with very many predictors and/or very many observations. Depending on the choice of a reward, TVS can be deployed in offline as well as online setups with streaming data batches. Tailoring multiplay bandits to variable selection, we provide regret bounds without necessarily assuming that the arm mean rewards be unrelated. We show a very strong empirical performance on both simulated and real data. Unlike deterministic optimization methods for spike-and-slab variable selection, the stochastic nature makes TVS less prone to local convergence and thereby more robust.

This chapter has been published as a paper in the Journal of American Statistical Association in joint authorship with Veronika Rockova[150]. In this paper, I prove the key theorems and conducted the simulations and data analysis.

3.2 Interpretable Machine Learning

A fundamental challenge in statistics that goes beyond mere prediction is to glean interpretable insights into the nature of real-world processes by identifying important correlates of variation. Many today's most powerful prediction tools, however, lack an intuitive algebraic form which renders their interpretability (i.e. insight into the black box decision process) far from straightforward. Substantial effort has been recently devoted to enhancing the explainability of machine learning through the identification of key variables that drive predictions ([80, 174, 250, 153, 30, 103]). While these procedures may possess nice theoretical guarantees, they may not yet be feasible for large-scale applications. This work develops a new computational platform for understanding black-box predictions which is based on reinforcement learning and which can be applied to very large datasets.

A variable can be important because its change has a causal impact or because leaving it out reduces overall prediction capacity ([162]). Such leave-one-covariate-out type inference has a long tradition, going back to at least [24]. In random forests, for example, variable importance is assessed by the difference between prediction errors in the out-of-bag sample before and after noising the covariate through a permutation. [132] propose the LOCO method which gauges local effects of removing each covariate on the overall prediction capability and derives an asymptotic distribution for this measure to conduct proper statistical tests. There is a wealth of literature on variable importance measures, see [70] for a recent overview. In Bayesian forests, such as BART ([46]), one keeps track of predictor inclusion frequencies and outputs an average proportion of all splitting rules inside a tree ensemble that split on a given variable. In deep learning, one can construct variable importance measures using network weights ([80, 246]). [177] introduce a variable importance based on a Shapley value and [102] investigates diagnostics of black box functions using functional ANOVA decompositions with dependent covariates. While useful for ranking variables, importance measures are less intuitive for model selection and are often not well-understood theoretically (with a few exceptions including [106, 117]).

This work focuses on high-dimensional applications (either very many predictors or very many observations, or both), where computing importance measures and performing tests for predictor effects quickly becomes infeasible. We consider the non-parametric regression model which provides a natural statistical framework for supervised machine learning. The data setup consists of a continuous response vector $\mathbf{Y}^{(n)} = (Y_1, \cdots, Y_n)'$ that is linked stochastically to a fixed set of predictors $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})'$ for $1 \leq i \leq n$ through

$$Y_i = f_0(\mathbf{x}_i) + \epsilon_i \quad \text{where } \epsilon_i \overset{iid}{\sim} \mathrm{N}(0, \sigma^2), \tag{3.1}$$

and where f_0 is an unknown regression function. The variable selection problem occurs when there is a subset $\mathcal{S}_0 \subset \{1, \cdots, p\}$ of $q_0 =\mid \mathcal{S}_0 \mid$ predictors which exert influence on the mixing function f_0 and we do not know which subset it is. In other words, f_0 is constant in directions outside $\mathcal{S}_0$ and the goal is to identify active directions (regressors) in $\mathcal{S}_0$ while, at the same time, permitting nonlinearities and interactions. The traditional Bayesian approach to this problem starts with a prior distribution over the 2^p sets of active variables. This is typically done in a hierarchical fashion by first assigning a prior distribution $\pi(q)$ on the subset size $q =\mid \mathcal{S} \mid$ and then a conditionally uniform prior on $\mathcal{S}$, given q, i.e. $\pi(\mathcal{S} \mid q) = \frac{1}{\binom{p}{q}}$. This prior can be translated into the *spike-and-slab* prior where, for each coordinate $1 \leq i \leq p$, one assumes a binary indicator γ_i for whether or not the variable x_i is active and assigns a prior

$$\mathbb{P}(\gamma_i \mid \theta) = \theta, \quad \theta \sim \texttt{Beta}(a, b) \text{ for some } a, b > 0. \tag{3.2}$$

The active subset $\mathcal{S}$ is then constructed as $\mathcal{S} = \{j : \gamma_j = 1\}$. There is no shortage of literature on spike-and-slab variable selection in the linear model, addressing prior choices ([165, 200, 203, 241, 27]), computational aspects ([36, 198, 23], [83],[84]) and/or variable selection consistency results ([40, 115, 169]). Traditionally, spike-and-slab methodology relies on the underlying model to be linear, which may be woefully inaccurate, and can be computationally slow. In this work, we leave behind the linear model framework and fo-

cus on interpretable machine learning linking spike-and-slab methods with binary bandits. The two major methodological benefits are (a) ability to capture non-linear effects and (b) scalability to very large datasets. Existing non-linear variable selection approaches include grouped shrinkage/selection of basis-expansion coefficients [142, 188, 190, 212], regularization of the derivative expectation operator [128] or model-free knockoffs [34]. The main distinguishing feature of our approach is the development of a new computational platform via a spike-and-slab wrapper that extends the reach of machine learning to large-scale data.

This paper introduces Thompson Variable Selection (TVS), a stochastic optimization approach to subset selection based on reinforcement learning. The key idea behind TVS is that variable selection can be regarded as a combinatorial bandit problem where each variable is treated as an arm. TVS sequentially learns promising combinations of arms (variables) that are most likely to provide a reward. Depending on the learning tool for modeling f_0 (not necessarily a linear model), TVS accommodates a wide range of rewards for both offline and online (streaming batches) setups. The fundamental appeal of active learning for subset selection (as opposed to MCMC sampling) is that those variables which provided a small reward in the past are less likely to be pulled again in the future. This exploitation aspect steers model exploration towards more promising combinations and offers dramatic computational dividends. Indeed, similarly as with backward elimination TVS narrows down the inputs contributing to f_0 but does so in a stochastic way by learning from past mistakes. TVS aggregates evidence for variable inclusion and quickly separates signal from noise by minimizing regret motivated by the median probability model rule [10]. We provide regret bounds which do not necessarily assume that the arm outcomes be unrelated. In addition, we show strong empirical performance and demonstrate the potential of TVS to meet demands of very large datasets.

This paper is structured as follows. Section 3.3 revisits known facts about multi-armed bandits. Section 3.4 develops the bandits framework for variable selection and Section 3.5 proposes Thompson Variable Selection and presents a regret analysis. Section 3.6 presents

two implementations (offline and online) on two benchmark simulated data. Section 3.7 presents a thorough simulation study and Section 3.8 showcases TVS performance on real data. We conclude with a discussion in Section 3.9.

3.3 Multi-Armed Bandits Revisited

Before introducing Thompson Variable Selection, it might be useful to review several known facts about multi-armed bandits. The multi-armed bandit (MAB) problem can be motivated by the following gambling metaphor. A slot-machine player needs to decide between multiple arms. When pulled at time t, the i^{th} arm gives a random payout $\gamma_i(t)$. In the Bernoulli bandit problem, the rewards $\gamma_i(t) \in \{0, 1\}$ are binary and $\mathbb{P}(\gamma_i(t) = 1) = \theta_i$. The distributions of rewards are unknown and the player can only learn about them through playing. In doing so, the player faces a dilemma: *exploiting* arms that have provided high yields in the past and *exploring* alternatives that may give higher rewards in the future.

More formally, an algorithm for MAB must decide which of the p arms to play at time t, given the outcome of the previous $t-1$ plays. A natural goal in the MAB game is to minimize *regret*, i.e. the amount of money one loses by not playing the optimal arm at each step. Denote with $i(t)$ the arm played at time t, with $\theta^\star = \max_{1 \le i \le p} \theta_i$ the best average reward and with $\Delta_i = \theta^\star - \theta_i$ the gap between the rewards of an optimal action and a chosen action. The expected regret after T plays can be then written as $\mathbb{E}[\mathcal{R}(T)] = \sum_{i=1}^{p} \Delta_i \mathbb{E}[k_i(T)]$, where $k_j(T) = \sum_{t=1}^{T} \mathbb{I}[i(t) = j]$ is the number of times an arm j has been played up to step T. There have been two main types of algorithms designed to minimize regret in the MAB problem: Upper Confidence Bound (UCB) of [129] and Thompson Sampling (TS) of [232]. Thompson Sampling is a Bayesian-inspired heuristic algorithm that achieves a logarithmic expected regret [3] in the Bernoulli bandit problem. Starting with a non-informative prior $\theta_i \overset{iid}{\sim} \texttt{Beta}(1, 1)$ for $1 \le i \le p$, this algorithm: (a) updates the distribution of θ_i as $\texttt{Beta}(a_i(t)+1, b_i(t)+1)$, where $a_i(t)$ and $b_i(t)$ are the number of successes and failures of the arm i up to time t, (b) samples $\theta_i(t)$ from these posterior distributions, and (c) plays

the arm with the highest $\theta_i(t)$. [3] extended this algorithm to the general case where rewards are not necessarily Bernoulli but general random variables on the interval $[0, 1]$.

The MAB problem is most often formulated as a single-play problem, where only one arm can be selected at each round. [123] extended Thompson sampling to a *multi-play scenario*, where at each round t the player selects a subset $\mathcal{S}_t$ of $L < p$ arms and receives binary rewards of all selected arms. For each $1 \leq i \leq p$, these rewards $r_i(t)$ are iid Bernoulli with unknown success probabilities θ_i where $\gamma_i(t)$ and $\gamma_j(t)$ are independent for $i \neq j$ and where, without loss of generality, $\theta_1 > \theta_2 > \cdots > \theta_p$. The player is interested in maximizing the sum of expected rewards over drawn arms, where the optimal action is playing the top L arms $\mathcal{S}_0 = \{1, \ldots, L\}$. The regret depends on the combinatorial structure of arms drawn and, similarly as before, is defined as the gap between an expected cumulative reward and the optimal drawing policy, i.e. $\mathbb{E}[\mathcal{R}(T)] = \mathbb{E} \sum_{t=1}^{T} \left(\sum_{i \in \mathcal{S}_0} \theta_i - \sum_{i \in \mathcal{S}_t} \theta_i \right)$ Fixing L, the number of arms played, [123] propose a Thompson sampling algorithm for this problem and show that it has a logarithmic expected regret with respect to time and a linear regret with respect to the number of arms. Our metamorphosis of multi-armed bandits into a variable selection algorithm will ultimately require that the number L of arms played is random and that the rewards at each time t can be dependent.

Finally, we complete the review of MAB techniques with *combinatorial bandits* ([44, 79, 43]) which are the closest relative to our proposed method here. Combinatorial bandits can be seen as a generalization of multi-play bandits, where any arbitrary combination of arms $\mathcal{S}$ (called super-arms) is played at each round and where the reward $r(\mathcal{S})$ can be revealed for the entire collective $\mathcal{S}$ (a full-bandit feedback) or for each contributing arm $i \in \mathcal{S}$ (a semi-bandit feedback), see e.g. [243, 51, 126, 51, 126]. We will draw upon connections between combinatorial bandits and variable selection multiple times throughout Section 3 and 4.

3.4 Variable Selection as a Bandit Problem

The purpose of this section is to link spike-and-slab model selection with multi-armed bandits. Before formalizing the ideas, we discuss two possibilities inspired by the search for the MAP (maximum-a-posteriori) model and the MPM (median probability) model.

Bayesian model selection with spike-and-slab priors has often been synonymous to finding the MAP model $\widehat{\mathcal{S}} = \arg\max_{\mathcal{S}} \pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$. Even when the marginal likelihood is available, this model can computationally unattainable for p as small as 20. In order to accelerate Bayesian variable selection using multi-armed bandits techniques one idea immediately comes to mind. One could treat each of the 2^p *models* as a base arm. Assigning prior model probabilities according to $\theta_i \sim \texttt{Beta}(a_i, b_i)$ for $1 \leq i \leq 2^p$ for some[1] $a_i > 0$ and $b_i > 0$, one could play a game by sequentially trying out various arms (variable subsets) and collect rewards to prioritize subsets that were suitably "good". Identifying the arm with the highest mean reward could then serve as a proxy for the best model. This naive strategy, however, would not be operational due to the exponential number of arms to explore.

Instead of the MAP model, it has now been standard practice to report the *median probability model* (MPM) ([10]) consisting of those variables whose posterior inclusion probability $\pi_i \equiv \mathbb{P}(\gamma_i = 1 \mid \boldsymbol{Y}^{(n)})$ is at least 0.5. More formally, MPM is defined, for $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)'$, as

$$\widehat{\mathcal{S}}_{MPM} = \arg\max_{\mathcal{S}} r_{\boldsymbol{\pi}}(\mathcal{S}) = \{i : \pi_i \geq 0.5\} \quad \text{where} \quad r_{\boldsymbol{\pi}}(\mathcal{S}) = \left\{ \prod_{i \in \mathcal{S}} \pi_i \prod_{i \notin \mathcal{S}} (1 - \pi_i) \right\}. \quad (3.3)$$

This is now the default model selection rule with spike-and-slab priors (3.2). The MPM model is the optimal predictive model in linear regression under some assumptions [12]. Obtaining π_i's, albeit easier than finding the MAP model, requires posterior sampling over variable subsets. While this can be done using standard MCMC sampling techniques in linear regression ([84, 169, 19]), here we explore new curious connections to bandits in order

1. chosen to correspond to marginals of a Dirichlet distribution

to develop a much faster stochastic optimization routine for finding MPM-alike models when the true model is not necessarily linear.

Having reviewed the two traditional Bayesian model choice reporting methods (MAP and MPM), we can now forge connections to multi-armed bandits. While the MAP model suggests treating *each model* $\mathcal{S}$ as a bandit arm, the MPM model suggests treating *each variable* γ_i as a bandit arm. Under the MAP framework, the player would be required to play a single arm (i.e. a model) at each step. The MPM framework, on the other hand, requires playing a random subset of arms (i.e. a model) at each play opportunity. This is appealing for at least two reasons: (1) there are fewer arms to explore more efficiently, (2) the quantity $r_{\boldsymbol{\pi}}(\mathcal{S})$ in (3.3) can be regarded as a mean regret of a combinatorial arm (more below) which, given $\boldsymbol{\pi}$, has MPM as its computational oracle. The computational oracle is defined as the regret minimizer when an oracle furnishes yield probabilities θ_i (see forthcoming Lemma 3.4.1). Based on the discussion above, we regard the MPM framework as more intuitively appealing for bandit techniques. We thereby reframe spike-and-slab selection with priors (3.2) as a bandit problem treating each variable as an arm. This idea is formalized below.

We view Bayesian spike-and-slab selection through the lens of *combinatorial bandit problems* (reviewed earlier in Section 3.3) by treating variable selection indicators γ_i's in (3.2) as Bernoulli rewards. From now on, we will refer to each θ_i as an unknown mean reward, i.e. a probability that the i^{th} variable exerts influence on the outcome. In sharp contrast to (3.2) which deploys one θ for all arms, each arm $i \in \{1, \ldots, p\}$ now has *its own* prior inclusion probability θ_i, i.e.

$$\mathbb{P}(\gamma_i = 1 \mid \theta_i) = \theta_i,\ \theta_i \sim \texttt{Beta}(a_i, b_i) \text{ for some } a_i, b_i > 0. \tag{3.4}$$

In the original spike-and-slab setup (3.2), the mixing weight θ served as a global shrinkage parameter determining the level of sparsity and linking coordinates to borrow strength ([200]).

In our new bandit formulation (3.4), on the other hand, the reward probabilities θ_i serve as a proxy for posterior inclusion probabilities π_i whose distribution we want to learn by playing the bandits game. Recasting the spike-and-slab prior in this way allows one to approach Bayesian variable selection from a more algorithmic (machine learning) perspective.

3.4.1 The Global Reward

Before proceeding, we need to define the reward in the context of variable selection. One conceptually appealing strategy would be to collect a joint reward $R(\mathcal{S}_t)$ (e.g. goodness of model fit) reflecting the collective effort of all contributing arms and then redistribute it among arms inside the super-arm $\mathcal{S}_t$ played at time t. One example would be the Shapley value ([217, 177]), a construct from cooperative game theory for the attribution problem that distributes the value created by a team to its individual members.

We try a different route.Rather than distributing, we will aggregate. Namely, instead of collecting a global reward first and then redistributing it, we first collect individual rewards $\gamma_i^t \in \{0, 1\}$ for each played arm $i \in \mathcal{S}_t$ and then weave them into a global reward $R(\mathcal{S}_t)$. We assume that γ_i^t's are `iid` from (3.4) for each $i \in \{1, \ldots, p\}$. Unlike traditional combinatorial bandits that define the global reward $R(\mathcal{S}_t) = \sum_{i \in \mathcal{S}} \gamma_i^t$ as a sum of individual outcomes [79], we consider a global reward for variable selection motivated by the median probability model.

One natural choice would be a *binary* global reward $R(\mathcal{S}_t) = \prod_{i \in \mathcal{S}_t} \gamma_i^t \prod_{i \notin \mathcal{S}_t} (1 - \gamma_i^t) \in \{0, 1\}$ for whether or not *all* arms inside $\mathcal{S}_t$ yielded a reward and, at the same time, *none* of the arms outside $\mathcal{S}_t$ did. Assuming independent arms, the expected reward equals $\mathbb{E}[R(\mathcal{S}_t)] = \prod_{i \in \mathcal{S}_t} \theta_i \prod_{i \notin \mathcal{S}_t} (1 - \theta_i) = r_{\boldsymbol{\theta}}(\mathcal{S}_t)$ and has the "median probability model" as its computational oracle, as can be seen from (3.3). However, this expected reward is not monotone in θ_i's (a requirement needed for regret analysis) and, due to its dichotomous nature, it penalizes all mistakes (false positives and negatives) equally.

We consider an alternative reward function which also admits a computational oracle but

treats mistakes *differentially*. For some $0 < C < 1$, we define the *global reward* $R_C(\mathcal{S}_t)$ for a subset $\mathcal{S}_t$ at time t as

$$R_C(\mathcal{S}_t) = \sum_{i \in \mathcal{S}_t} \log\left(C + \gamma_i^t\right). \tag{3.5}$$

Similarly as $R(\mathcal{S}_t)$ (defined above) the reward is maximized for the model which includes all the positive arms and none of the negative arms, i.e. $\arg\max_{\mathcal{S}} R_C(\mathcal{S}) = \{i : \gamma_i^t = 1\}$. Unlike $R(\mathcal{S}_t)$, however, the reward will penalize subsets with false positives, a penalty $\log(C)$ for each, and there is an opportunity cost of $\log(1 + C)$ for each false negative. The expected global reward depends on the subset $\mathcal{S}_t$ and the vector of yield probabilities $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)'$, i.e.

$$r_{\boldsymbol{\theta}}^C(\mathcal{S}_t) = \mathbb{E}\left[R_C(\mathcal{S}_t)\right] = \sum_{i \in \mathcal{S}_t} \left[\theta_i \log\left(\frac{C+1}{C}\right) - \log\left(\frac{1}{C}\right)\right]. \tag{3.6}$$

Note that this expected reward *is monotone* in θ_i's and is Lipschitz continuous. Moreover, it *also has* the median probability model as its computational oracle.

Lemma 3.4.1. *Denote with* $\mathcal{S}_O = \arg\max_{\mathcal{S}} r_{\boldsymbol{\theta}}^C(\mathcal{S})$ *the computational oracle. Then we have*

$$\mathcal{S}_O = \left\{i : \theta_i \geq \frac{\log(1/C)}{\log[(C+1)/C]}\right\}. \tag{3.7}$$

With $C = (\sqrt{5} - 1)/2$*, the oracle is the median probability model* $\{i : \theta_i \geq 0.5\}$*.*

Proof: It follows immediately from the definition of $R_C(\mathcal{S}_t)$ and the fact that $\log(1/C) = 0.5 \log[(1 + C)/C]$ for $C = (\sqrt{5} - 1)/2$.

Note that the choice of $C = (\sqrt{5} - 1)/2$ incurs *the same* penalty/opportunity cost for false positives and negatives since $\log(1 + C) = -\log(C)$. In streaming feature selection, for example, [253] accommodate measurement cost and place cheaper variables earlier in the stream. In our framework, we can allow for different cost C_i (e.g. a measurement cost) for each variable $1 \leq i \leq p$. The existence of the computational oracle for the expected reward $r_{\boldsymbol{\theta}}^C(\mathcal{S})$ is very comforting and will be exploited in our Thompson sampling algorithm introduced in Section 3.5

3.4.2 The Local Rewards

The global reward (3.5) is a deterministic functional of the local rewards. We have opted for the reward functional (3.5) because the regret minimizer is the median probability model when the yield probabilities are provided (see Lemma 3.4.1). We now clarify the definition of local rewards γ_i^t. We regard $\mathcal{S}_t$ as a smaller pool of candidate variables, which can contain false positives and false negatives. The goal is to play a game by sequentially trying out different subsets and reward true signals so that they are selected in the next round and to discourage false positives from being included again in the future. Denote with $\mathbb{S}$ the set of all subsets of $\{1, \ldots, p\}$ and with $\mathcal{D}$ the "data" at hand consisting of $\mid \mathcal{D} \mid$ observations $(Y_i, \boldsymbol{x}_i)$ from (3.1). We introduce a *feedback rule*

$$r(\mathcal{S}_t, \mathcal{D}) : \mathbb{S} \times \mathbb{R}^{|\mathcal{D}|} \to \{0, 1\}^{|\mathcal{S}_t|}, \tag{3.8}$$

which, when presented with data $\mathcal{D}$ and a subset $\mathcal{S}_t$, outputs a vector of *binary rewards* $r(\mathcal{S}_t, \mathcal{D}) = (\gamma_i^t : i \in \mathcal{S}_t)'$ for whether or not a variable x_i for $i \in \mathcal{S}_t$ is relevant for predicting or explaining the outcome. This feedback is only revealed if $i \in \mathcal{S}_t$. We consider two sources of randomness that implicitly define the reward distribution $r(\mathcal{S}_t, \mathcal{D})$: (1) a *stochastic feedback rule* $r(\cdot)$ assuming that data $\mathcal{D}$ is given, and (2) a *deterministic feedback rule* $r(\cdot)$ assuming that data $\mathcal{D}$ is stochastic.

The first reward type has a Bayesian flavor in the sense that it is *conditional* on the observed data $\mathcal{D}_n = \{(Y_i, \boldsymbol{x}_i) : 1 \leq i \leq n\}$, where rewards can be sampled using Bayesian stochastic computation (i.e. MCMC sampling). Such rewards are natural in *offline settings* with Bayesian feedback rules, as we explore in Section 3.6.1. As a lead example of this strategy in this paper, we consider a stochastic reward based on BART (Chipman et al. (2001)). We refer to [101] and references therein for a nice recent overview of BART. In

particular, we use the following binary local reward $r(\mathcal{S}_t, \mathcal{D}_n) = (\gamma_i^t : i \in \mathcal{S}_t)'$ where

$$\gamma_i^t = \mathbb{I}(M^{th} \text{ sample from the BART posterior splits on the variable } x_i). \tag{3.9}$$

The mean reward $\theta_i = \mathbb{P}(\gamma_i^t = 1) = \mathbb{P}[i \in \mathcal{F} \mid \mathcal{D}_n]$ can be interpreted as the posterior probability that a variable x_i is split on in a Bayesian forest $\mathcal{F}$ given the entire data $\mathcal{D}_n$. The stochastic nature of the BART computation allows one to regard the reward (3.9) as an actual random variable, whose values can be sampled from using standard software. Since BART is run *only* with variables inside $\mathcal{S}_t$ (where $\mid \mathcal{S}_t \mid << p$) and *only* for M burn-in MCMC iterations, computational gains are dramatic (as we will see in Section 3.6.1).

The second reward type has a frequentist flavor in the sense that rewards are sampled by applying deterministic feedback rules on new streams (or bootstrap replicates) $\mathcal{D}_t$ of data. Such rewards are natural in *online settings*, as we explore in Section 3.6.2. As a lead example of this strategy in this paper, we assume that the dataset $\mathcal{D}_n$ consist of $n = sT$ observations and is partitioned into minibatches $\mathcal{D}_t = \{(Y_i, \boldsymbol{x}_i) : (t-1)s + 1 \leq i \leq ts\}$ for $t = 1, \ldots, T$. One could think of these batches as new independent observations arriving in an online fashion or as manageable snippets of big data. The 'deterministic' screening rule we consider here is running BART for a large number M of MCMC iterations and collecting an aggregated importance measure $IM(i;\ \mathcal{D}_t, \mathcal{S}_t)$ for each variable.[2] We define $IM(i;\ \mathcal{D}_t, \mathcal{S}_t)$ as the average number of times a variable x_i was used in a forest where the average is taken over the M iterations and we then reward those arms which were used at least once on average,

$$\gamma_i^t = \mathbb{I}[IM(i;\ \mathcal{D}_t, \mathcal{S}_t) \geq 1]. \tag{3.10}$$

The mean reward $\theta_i = \mathbb{P}(\gamma_i^t = 1)$ can be then interpreted as the (frequentist) probability that BART, when run on $s = n/T$ observations arising from (3.1), uses a variable x_i at least

2. This rule is deterministic in the sense that computing it again on the same data should in principle provide the same answer. One could, in fact, deploy any other machine learning method that outputs some measure of variable importance.

once on average over M iterations. We illustrate this online variant in Section 3.6.2.

3.4.3 Other Feedback Rules

TVS is not confined to a BART reward function. Deploying any variable selection method using only a subset $\mathcal{S}_t$ will yield a binary feedback rule. For example, the LASSO method yields $\hat{\beta}_i$ for $i \in \mathcal{S}_t$ which can be turned into the following feedback $\gamma_i^t = \mathbb{I}(\hat{\beta}_i \neq 0)$ for $i \in \mathcal{S}_t$. In offline setups, randomness of γ_i^t's can be induced by taking a bootstrap replicate of $\mathcal{D}_n$ at each play time $1 \leq t \leq T$. Another possibility for a binary reward is dichotomizing p-values, similarly as in alpha-investing for streaming variable selection by [253]. Instead of binary local rewards (3.8), one can also consider continuous rewards $\gamma_i^t \in [0,1]^{|\mathcal{S}_t|}$ by rescaling variable importance measures obtained by a machine learning method (e.g. random forests ([152]), deep learning ([103]), and BART ([49])). Our Thompson sampling algorithm can be then modified by dichotomizing these rewards through independent Bernoulli trials with probabilities equal to the continuous rewards [3].

3.5 Introducing Thompson Variable Selection (TVS)

This section introduces Thompson Variable Selection (TVS), a reinforcement learning algorithm for subset selection in non-parametric regression environments. The computation alternates between *Choose, Reward* and *Update* steps that we describe in more detail below.

The unknown mean rewards will be denoted with $\theta_i^\star$ and the ultimate goal of TVS is to learn their distribution once we have seen the 'data'[3]. To this end, we take the *combinatorial bandits* perspective ([44], Gai et al. (2010)) where, instead of playing one arm at each play opportunity t, we play a random subset $\mathcal{S}_t \subseteq \{1, \ldots, p\}$ of multiple arms. Each such *super-arm* $\mathcal{S}_t$ corresponds to a *model* configuration and the goal is to discover promising models by playing more often the more promising variables.

Similarly as with traditional Thompson Sampling, the t^{th} iteration of TVS starts off by sampling mean rewards $\theta_i(t) \sim \text{Beta}(a_i(t), b_i(t))$ from a posterior distribution that incorpo-

3. The 'data' here refers to the sequence of observed rewards.

Algorithm 2 Thompson Variable Selection with BART (* is an alternative with known $q^\star$)

Algorithm 1: *Thompson Variable Selection with BART*
INPUT
Define $\widetilde{C} = \frac{\log(1/C)}{\log[(1+C)/C}$ for some $0 < C < 1$ and pick $M, a, b > 0$
Initialize $a_i(0) := a$ and $b_i(0) := b$ for each arm $1 \leq i \leq p$.
LOOP
For $t = 1, \ldots, T$ repeat steps C(1)-(3), R and U.
Choose Step
C(1): Set $\mathcal{S}_t = \emptyset$ and for $i = 1 \cdots p$ do
C(2): Sample $\theta_i(t) \sim \texttt{Beta}(a_i(t), b_i(t))$
C(3): (Unknown $q^\star$) Compute $\mathcal{S}_t = \{i : \theta_i(t) \geq \widetilde{C}\}$ from (3.7)
C(3)*: (Known $q^\star$) Compute $\mathcal{S}_t$ from (3.13)
Reward Step
R: Collect local rewards γ_i^t for each $1 \leq i \leq p$ from (3.9) (offline) or (3.10) (online)
Update Step
U: If $\gamma_i^t = 1$ then set $a_i(t+1) = a_i(t) + 1$, else $b_i(t+1) = b_i(t) + 1$
OUTPUT
Evidence probabilities $\pi_i(t) = a_i(t)/[a_i(t) + b_i(t)]$ for $1 \leq i \leq p$ and $1 \leq t \leq T$.

rates past reward experiences up to time t (as we discussed in Section 3.3). The *Choose Step* then decides which arms will be played in the next round. While the single-play Thompson sampling policy dictates playing the arm with the highest sampled expected reward, the combinatorial Thompson sampling policy ([243]) dictates playing the *subset* that maximizes the expected global reward, given the vector of sampled probabilities $\boldsymbol{\theta}(t) = (\theta_1(t), \ldots, \theta_p(t))'$. The availability of the computational oracle (from Lemma 3.4.1) makes this step awkwardly simple as it boils down to computing $\mathcal{S}_O$ in (3.7). Unlike multi-play bandits where the number of played arms is predetermined [123], this strategy allows one to adapt to the size of the model. We do, however, consider a variant of the computational oracle (see (3.13) below) for when the size $q^\star =\mid \mathcal{S}^\star \mid$ of the 'true' model $\mathcal{S}^\star = \arg\max_{\mathcal{S}} r^C_{\boldsymbol{\theta}^\star}(\mathcal{S})$ is known. The *Choose Step* is then followed by the *Reward Step* (step R in Table 2) which assigns a prize to the chosen subset $\mathcal{S}_t$ by collecting individual rewards γ_i^t (for the offline setup in (3.9) or for the online setup (3.10)). Finally, each TVS iteration concludes with an *Update Step* which updates the beta posterior distribution (step U in Table 2).

The fundamental goal of Thompson Variable Selection is to learn the distribution of mean rewards θ_i's by playing a game, i.e. sequentially creating a dataset of rewards by

sampling from beta posterior[4] distributions that incorporate past rewards and the observed data $\mathcal{D}$. One natural way to distill evidence for variable selection is through the means $\boldsymbol{\pi}(t) = (\pi_1(t), \ldots, \pi_p(t))'$ of these beta distributions

$$\pi_i(t) = \frac{a_i(t)}{a_i(t) + b_i(t)}, \quad 1 \leq i \leq p, \tag{3.11}$$

which serve as a proxy for posterior inclusion probabilities. Similarly as with the classical median probability model [10], one can deem important those variables with $\pi_i(t)$ above 0.5 (this corresponds to one specific choice of C in Lemma 3.4.1). More generally, at each iteration t TVS outputs a model $\widehat{\mathcal{S}}_t$, which satisfies $\widehat{\mathcal{S}}_t = \arg\max_{\mathcal{S}} r^C_{\boldsymbol{\pi}(t)}$. From Lemma 3.4.1, this model can be simply computed by truncating individual $\pi_i(t)$'s. Upon convergence, i.e. when trajectories $\pi_i(t)$ stabilize over time, TVS will output the same model $\widehat{\mathcal{S}}_t$. We will see from our empirical demonstrations in Section 3.6 that the separation between signal and noise (based on $\pi_i(t)$'s) and the model stabilization occurs fast. Before our empirical results, however, we will dive into the regret analysis of TVS.

3.5.1 Regret Analysis

Thompson sampling (TS) is a policy that uses Bayesian ideas to solve a fundamentally frequentist problem of regret minimization. In this section, we explore regret properties of TVS and expand current theoretical understanding of combinatorial TS by allowing for correlation between arms. Theory for TS was essentially unavailable until the path-breaking paper by [3] where the first finite-time analysis was presented for single-play bandits. Later, [133] proved that TS converges to the optimal policy in probability and almost surely under some assumptions. Several theoretical and empirical studies for TS in *multi-play* bandits are also available. In particular, [123] extended TS to multi-play problems with a fixed number of played arms and showed that it achieves the optimal regret bound. Recently, [243]

4. This posterior treats the past rewards as the actual data.

introduced TS for combinatorial bandits and derived regret bounds for Lipschitz-continuous rewards under an offline oracle. We build on their development and extend their results to the case of related arms.

Recall that the goal of the player is to minimize the total (expected) regret under time horizon T defined below

$$Reg(T) = \mathbb{E}\left[\sum_{t=1}^{T}\left(r^{C}_{\boldsymbol{\theta}^\star}(\mathcal{S}^\star) - r^{C}_{\boldsymbol{\theta}^\star}(\mathcal{S}_t)\right)\right], \tag{3.12}$$

where $\mathcal{S}^\star = \arg\max_{\mathcal{S}} r^{C}_{\boldsymbol{\theta}^\star}(\mathcal{S}_t)$ with $q^\star =\mid \mathcal{S}^\star \mid$, $\theta^\star_i = \mathbb{E}[\gamma^t_i]$ and where the expectation is taken over the unknown drawing policy. Choosing C as in Lemma 3.4.1, one has $\log(1+C) = -\log(C) = D$ and thereby

$$Reg(T) = D\,\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{p}(2\theta^\star_i - 1)[\mathbb{I}(i \in \mathcal{S}^\star \backslash \mathcal{S}_t) - \mathbb{I}(i \in \mathcal{S}_t \backslash \mathcal{S}^\star)]\right].$$

Note that $(2\theta^\star_i - 1)$ is positive iff $i \in \mathcal{S}^\star$. Upper bounds for the regret (3.12) under the drawing policy of our TVS Algorithm 1 can be obtained under various assumptions. Below, we first review two available regret bounds following from [243] for when (a) $q^\star$ is known and arms are independent (Lemma 3.5.1) and (b) arms are related and $q^\star$ is unknown (Lemma 3.5.2). Later in our Theorem 3.5.1, we relax these assumptions and provide a regret bound assuming that $q^\star$ unknown and, at the same time, arms are related.

Assuming that the size $q^\star$ of the optimal model $\mathcal{S}^\star$ is *known*, one can modify Algorithm 1 to confine the search to models of size up to $q^\star$. Denoting $\mathcal{I} = \{\mathcal{S} \subset \{1,\ldots,p\} :\mid \mathcal{S} \mid \leq q^\star\}$, one plays the optimal set of arms *within the set* $\mathcal{I}$, i.e. replacing the computational oracle in (3.7) with $\mathcal{S}^{q^\star}_O = \arg\max_{\mathcal{S}\in\mathcal{I}} r_{\boldsymbol{\theta}}(\mathcal{S})$. We denote this modification with $C2^\star$ in Table 2. It turns out that this oracle can also be easily computed, where the solution consists of (up to)

the top $q^\star$ arms that pass the selection threshold, i.e.

$$\mathcal{S}_O^{q^\star} = \left\{ i : \theta_i \geq \frac{\log(1/C)}{\log[(1+C)/C]} \right\} \cap J(\boldsymbol{\theta}) = \{(i_1, \ldots, i_{q^\star})' \in \mathbb{N}^{q^\star} : \theta_{i_1} \geq \theta_{i_2} \geq \cdots \geq \theta_{i_q^\star}\}. \tag{3.13}$$

We have the following regret bound which, unlike the majority of existing results for Thompson sampling, *does not* require the arms to have independent outcomes γ_i^t. The regret bound depends on the amount of separation between signal and noise.

Lemma 3.5.1. *Define the identifiability gap* $\Delta_i = \min\left\{\theta_j^\star : \theta_j^\star > \theta_i^\star \text{ for } j \in \mathcal{S}^\star\right\}$ *for each arm* $i \notin \mathcal{S}^\star$. *The Algorithm 1 with a computational oracle* $\mathcal{S}_O^{q^\star}$ *in C2** *achieves the following regret bound*

$$Reg(T) \leq \sum_{i \notin \mathcal{S}^\star} \frac{(\Delta_i - \varepsilon)\log T}{(\Delta_i - 2\varepsilon)^2} + C\left(\frac{p}{\varepsilon^4}\right) + p^2$$

for any $\varepsilon > 0$ *such that* $\Delta_i > 2\varepsilon$ *for each* $i \notin \mathcal{S}^\star$ *and for some constant* $C > 0$.

Proof: Since $\mathcal{I}$ is a matroid ([125]) and our mean regret function is Lipschitz continuous and it depends only on expected rewards of revealed arms, one can apply Theorem 4 of [243].

Assuming that the size $q^\star$ of the optimal model is *unknown* and the rewards γ_i^t are *independent*, one can derive the following bound for the original Algorithm 1 (without restricting the solution to up to $q^\star$ variables).

Lemma 3.5.2. *Define the maximal reward gap* $\Delta_{max} = \max_{\mathcal{S}} \Delta_{\mathcal{S}}$ *where* $\Delta_{\mathcal{S}} = [r_{\boldsymbol{\theta}}(\mathcal{S}^\star) - r_{\boldsymbol{\theta}}(\mathcal{S})]$ *and for each arm* $i \in \{1, \ldots, p\}$ *define* $\eta_i \equiv \max_{\mathcal{S}: i \in \mathcal{S}} \frac{8B^2|\mathcal{S}|}{\Delta_{\mathcal{S}} - 2B(q^{\star 2}+2)\varepsilon}$ *for* $B = \log[(C+1)/C]$. *Assume that* γ_i^t*'s are independent for each* t. *Then the Algorithm 1 achieves the following regret bound*

$$Reg(T) \leq \log(T) \sum_{i=1}^{p} \eta_i + p\left(\frac{p^2}{\varepsilon^2} + 3\right)\Delta_{max} + C\frac{8\Delta_{max}}{\varepsilon^2}\left(\frac{4}{\varepsilon^2} + 1\right)^{q^\star} \log(q^\star/\varepsilon^2)$$

for some constant $C > 0$ *and for any* $\varepsilon > 0$ *such that* $\Delta_{\mathcal{S}} > 2B(q^{\star 2} + 2)\varepsilon$ *for each* $\mathcal{S}$.

Proof: Follows from Theorem 1 of [243].

The bandit literature has largely focused on studying the regret in terms of time T rather than the number of arms p. Note that the dependence on p in Lemma 3.5.2 is cubic, which (albeit relevant for large n setups) makes the bound less useful when p is very large. [129] showed a lower regret bound (in terms of p) that is $\mathcal{O}(p)$ for any bandit algorithm. [3] (Remark 3) further showed that Thompson Sampling does achieve this lower bound in a single-play bandit problem. Our Theorem 3.5.1 below shows a linear dependence on p for combinatorial bandits with correlated arms when $q^\star$ is unknown.

We now extend Lemma 3.5.2 to the case when the rewards obtained from pulling different arms are related to one another. [95] introduced a correlated single-play bandit version of Thompson sampling using pseudo-rewards (upper bounds on the conditional mean reward of each arm). Similarly as with structured bandits ([179]) we instead interweave the arms by allowing their mean rewards to depend on $\mathcal{S}_t$, i.e. instead of a single success probability θ_i we now have

$$\theta_i(\mathcal{S}) = \mathbb{P}(\gamma_i^t = 1 \mid \mathcal{S}_t = \mathcal{S}). \tag{3.14}$$

We are interested in obtaining a regret bound for the Algorithm 2 assuming (3.14) in which case the expected global regret (3.6) writes as

$$r_{\boldsymbol{\theta}}^C(\mathcal{S}_t) = \mathbb{E}\left[R_C(\mathcal{S}_t)\right] = \sum_{i \in \mathcal{S}_t} \left[\theta_i(\mathcal{S}_t) \log\left(\frac{C+1}{C}\right) - \log\left(\frac{1}{C}\right)\right]. \tag{3.15}$$

To this end we impose an identifiability assumption, which requires a separation gap between the reward probabilities of signal and noise arms.

Assumption 3.5.1. *Denote with* $\mathcal{S}^\star = \arg\max_{\mathcal{S}} r_{\boldsymbol{\theta}^\star}^C(\mathcal{S}_t)$ *the optimal set of arms. We say that* $\mathcal{S}^\star$ *is strongly identifiable if there exists* $0 < \alpha < 1/2$ *such that*

$$\begin{aligned} \forall i \in \mathcal{S}^* \quad \textit{we have} \quad \theta_i(\mathcal{S}^\star) \geq \theta_i(\mathcal{S}) > 0.5 + \alpha \quad \forall \mathcal{S} \quad \textit{such that} \quad i \in \mathcal{S}, \\ \forall i \notin \mathcal{S}^* \quad \textit{we have} \quad \theta_i(\mathcal{S}) < 0.5 - \alpha \quad \forall \mathcal{S} \quad \textit{such that} \quad i \in \mathcal{S}. \end{aligned}$$

Under this assumption we provide the following regret bound.

Theorem 3.5.1. *Suppose that $\mathcal{S}^\star$ is strongly identifiable with $\alpha > 0$. Choosing C as in Lemma 3.4.1, the regret of Algorithm 2 satisfies*

$$Reg(T) \leq \Delta_{\max}\left[\frac{8p\log(T)}{\alpha^2} + c(\alpha)q^* + \left(2 + \frac{4}{\alpha^2}\right)p\right], \tag{3.16}$$

where $c(\alpha) = \widetilde{C}\left[\frac{\mathbf{e}^{-4\alpha}}{1-\mathbf{e}^{-\alpha^2/2}}\right] + \frac{8}{\alpha^2}\frac{1}{\mathbf{e}^{2\alpha}-1} + \frac{\mathbf{e}^{-1}}{1-\mathbf{e}^{-\alpha/8}} + \left\lceil\frac{8}{\alpha}\right\rceil\left(3+\frac{1}{\alpha}\right)$ *and* $\widetilde{C} = \left(C_1 + C_2\frac{1-2\alpha}{32\,\alpha}\right)$ *for some $C_1, C_2 > 0$ not related to the Algorithm 2, and* $\Delta_{\max} = \max_{\mathcal{S}}[r^C_{\boldsymbol{\theta}}(\mathcal{S}^*) - r^C_{\boldsymbol{\theta}}(\mathcal{S})]$.

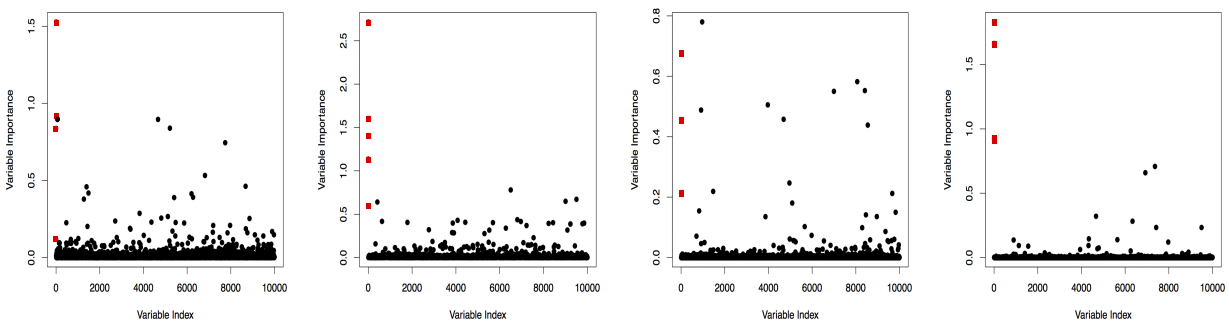
Proof: Appendix (Section B.1.1)

Two of the most common problems studied in reinforcement learning with bandits are (1) regret minimization, and (b) best arm identification [62, 28]. Variable selection could be loosely regarded as the 'top $q^\star$-arms' identification problem when $q^\star$ is unknown. While Thompson sampling is devised to minimize regret, not necessarily to select the best arms (see e.g. [208]), we nevertheless show that our sampling policy satisfies a version of variable selection consistency in the sense that the event $\{\mathcal{S}_t = \mathcal{S}^*\}$ occurs all but finitely many times as $t \to \infty$. This result is summarized in the following theorem.

Theorem 3.5.2. *Under the Assumption 1 with p fixed, the TVS sampling policy in Table 2 with* $C = (\sqrt{5}-1)/2$ *satisfies* $\mathbb{P}\left(\lim\inf_{t\to\infty}\{\mathcal{S}_t = \mathcal{S}^*\}\right) = 1$.

Remark 3.5.1. *Here is a sketch proof of the theorem: Because of Borel Cantelli Lemma, it is sufficient to prove that* $\sum_{t=1}^{\infty}\mathbb{P}\left(\{\mathcal{S}_t \neq \mathcal{S}^*\}\right) < \infty$. *The rest of the proof is an extension of the terms in theorem 3.5.1. Since the posterior distributions $\theta_i(t)$ concentrates quickly around the true mean of the arms ([3]),* $\mathbb{P}\left(\{\mathcal{S}_t \neq \mathcal{S}^*\}\right)$ *decays at an exponential rate with respect to each wrong pull. As such, one can show that* $\sum_{t=1}^{\infty}\mathbb{P}\left(\{\mathcal{S}_t \neq \mathcal{S}^*\}\right) < \infty$.

Proof: Appendix (Section B.1.1)



(a) $D = 50, M = 5\,000$ (b) $D = 50, M = 50\,000$ (c) $D = 10, M = 5\,000$ (d) $D = 10, M = 50\,000$

Figure 3.1: BART variable importance using `sparse=TRUE` and various number of trees D and MCMC iterations M. Red squares (the first five covariates) are signals and black dots are noise variables.

3.6 TVS in Action

This section serves to illustrate Thompson Variable Selection on benchmark simulated datasets and to document its performance. While various implementations are possible (by choosing different rewards $r(\mathcal{S}_t, \mathcal{D})$ in (3.8)), we will focus on two specific choices that we broadly categorize into *offline* variants for when $p >> n$ (Section 3.6.1) and *streaming/online* variants for when $n >> p$ (Section 3.6.2).

3.6.1 Offline TVS

As a lead example in this section, we consider the benchmark Friedman data set ([77]) with a vastly larger number of $p = 10\,000$ predictors $\boldsymbol{x}_i \in [0,1]^p$ obtained by iid sampling from `Uniform`$(0,1)$ and responses $\boldsymbol{Y} = (Y_1, \cdots Y_n)'$ obtained from (3.1) with $\sigma^2 = 1$ and

$$f_0(\boldsymbol{x}_i) = 10\sin(\pi x_{i1} x_{i2}) + 20(x_{i3} - 0.5)^2 + 10x_{i4} + 5x_{i5} \quad \text{for } i = 1, \ldots, 300.$$

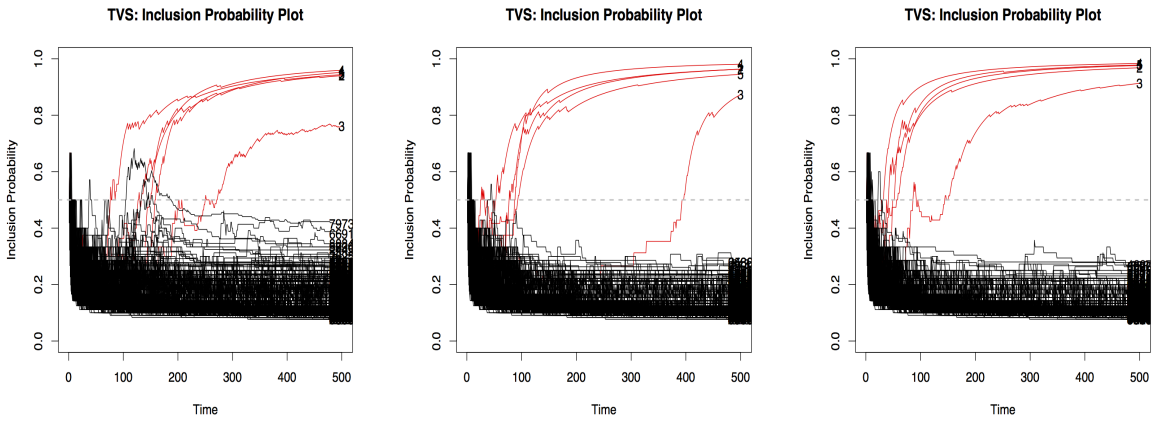
Due to the considerable number of covariates, feeding *all* 10 000 predictors into a black box to obtain variable importance may not be computationally feasible and/or reliable. However, TVS can overcome this limitation by deploying subsets of predictors. For instance, we

considered variable importance using the BART method (using the option `sparse=TRUE` for variable selection) with $D \in \{10, 50\}$ trees and $M \in \{5\,000, 50\,000\}$ MCMC iterations are plotted them in Figure 3.1. While increasing the number of iterations certainly helps in separating signal from noise, it is not necessarily obvious where to set the cutoff for selection. One natural rule would be selecting those variables which have been used at least once on average over the M iterations. With $D = 50$ and $M = 50\,000$, this rule would identify 4 true signals, leaving out the quadratic signal variable x_3. The computation took around 8.5 minutes.

The premise of TVS is that one can deploy a weaker learner (such as a forest with fewer trees) which generates a random reward that roughly captures signal and is allowed to make mistakes. With reinforcement learning, one hopes that each round will be wrong in a different way so that mistakes will not be propagated over time. The expectation is that (a) feeding *only a small subset* $\mathcal{S}_t$ in a black box and (b) reinforcing positive outcomes, one obtains a more principled way of selecting variables and speeds up the computation. We illustrate the effectiveness of this mechanism below.

We use the offline local binary reward defined in (3.9). We start with a non-informative prior $a_i(0) = b_i(0) = 1$ for $1 \leq i \leq p$ and choose $T = 10$ trees in BART so that variables are discouraged from entering the model too wildly. This is a *weak learner* which does not seem to do perfectly well for variable selection even after very many MCMC iterations (see Figure 3.1(d)). We use the TVS implementation in Table 2 with a dramatically smaller number $M \in \{100, 500, 1\,000\}$ of MCMC burn-in iterations for BART inside TVS. We will see below that large M is *not needed* for TVS to unravel signal even with as few as 10 trees.

TVS results are summarized in Figure 3.2, which depicts 'posterior inclusion probabilities' $\pi_i(t)$ defined in (3.11) over time t (the number of plays), one line for each of the $p = 10\,000$ variables. To better appreciate informativeness of $\pi_i(t)$'s, true variables $x_1, \ldots, x_5$ are depicted in red while the noise variables are black. Figure 3.2 shows a very successful demonstration for several reasons. The first panel (Figure 3.2(a)) shows a very weak learner



(a) $M = 100$ (b) $M = 500$ (c) $M = 1\,000$

Figure 3.2: The evolution of inclusion probabilities (3.11) over "time" t for the Friedman data set. The plot depicts posterior inclusion probabilities $\pi_i(t)$ in (3.11) over time (number of TVS iterations). Red lines indicate the 5 signal variables and black lines indicate the noise variables.

(as was seen from Figure 3.1) obtained by sampling rewards only after $M = 100$ burnin iterations. Despite the fact that learning at each step is weak, it took only around $T = 300$ iterations (obtained in less than 40 seconds!) for the $\pi_i(t)$ trajectories of the 5 signals to cross the 0.5 decision boundary. After $T = 300$ iterations, the noise covariates are safely suppressed below the decision boundary and the trajectories $\pi_i(t)$ stabilize towards the end of the plot. Using more MCMC iterations M, fewer TVS iterations are needed to obtain a cleaner separation between signal and noise (noise π_t's are closer to zero while signal π_t's are closer to one). With enough internal MCMC iterations ($M = 1\,000$ in the right panel), TVS is able to effectively separate signal from noise in around 200 iterations (obtained in less than 2 minutes). We have not seen such conclusive separation with plain BART (Figure 3.1) even after very many MCMC iterations which took considerably longer. Note that each iteration of TVS uses *only a small subset of predictors* and much fewer iterations M and TVS is thereby destined to be faster than BART on the entire dataset (compare 8.5 minutes for 20 000 iterations with 40 seconds in Figure 3.2(a)). Applying the more traditional variable selection techniques was also not as successful. For example, the Spike-and-Slab LASSO (SSL) method ($\lambda_1 = 0.1$ and $\lambda_0 \in \{0.1 + k \times 10 : k = 1, \ldots, 10\}$) which relies on a linear

model missed the quadratic predictor but identified all 4 remaining signals with no false positives.

3.6.2 Online TVS

As the second TVS example, we focus on the case with many more observations than covariates, i.e. $n >> p$. As we already pointed out in Section 3.4.2, we assume that the dataset $\mathcal{D}_n = \{(Y_i, \boldsymbol{x}_i) : 1 \leq i \leq n\}$ has been partitioned into mini-batches $\mathcal{D}_t$ of size $s = n/T$. We deploy our online TVS method (Table 2 with C2*) to sequentially screen each batch and transmit the posterior information onto the next mini-batch through a prior. This should be distinguished from streaming variable selection, where new features arrive at different time points ([71]). Using the notation $r_t \equiv r(\mathcal{S}_t, \mathcal{D}_t)$ in (3.8) with $\mathcal{D}_t = \{(Y_i, \boldsymbol{x}_i) : (t-1)s+1 \leq i \leq ts\}$ and having processed $t-1$ mini-batches, one can treat the beta posterior as a new prior for the incoming data points, where

$$\pi(\boldsymbol{\theta} \mid r_1, \ldots, r_t) \propto \pi(\boldsymbol{\theta} \mid r_1, \ldots, r_{t-1}) \prod_{i \in \mathcal{S}_t} \theta_i^{\gamma_i^t} (1-\theta_i)^{1-\gamma_i^t}.$$

Parsing the observations in batches will be particularly beneficial when processing the entire dataset (with overwhelmingly many rows) is not feasible for the learning algorithm. TVS leverages the fact that applying a machine learning method T times using only a subset of s observations and a subset $\mathcal{S}_t$ of variables is a lot faster than processing the entire data. While the posterior distribution[5] of θ_i's after one pass through the entire dataset will have seen all the data $\mathcal{D}_n$, θ_i's can be interpreted as the frequentist probability that the screening rule picks a variable x_i having access to only s measurements.

We illustrate this sequential learning method on a challenging simulated example from [138, Section 5.1] . We assume that the explanatory variables $\mathbf{x}_i \in [0,1]^p$ have been obtained from $x_{ij} = (e_i + z_{ij})/2$ for $1 \leq i \leq n$ and $1 \leq j \leq p$, where $e, z_{ij} \overset{iid}{\sim} \mathcal{N}(0,1)$. This creates a

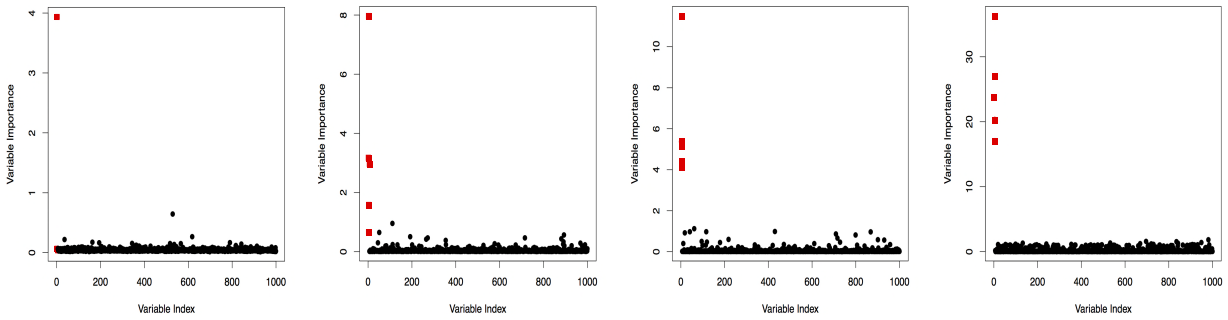
5. Treating the rewards as data.

sample correlation of about 0.5 between all variables. The responses $\mathbf{Y} = (Y_1, \cdots, Y_n)'$ are then obtained from (3.1), where

$$f_0(\mathbf{x}_i) = \frac{10x_{i2}}{1 + x_{i1}^2} + 5\sin(x_{i3}x_{i4}) + 2x_{i5}$$

with $\sigma^2 = 0.5$. This is a challenging scenario due to (a) the non-negligible correlation between signal and noise, and (b) the non-linear contributions of $x_1 - x_4$. Unlike Liang et al. (2008) who set $n = p = 500$, we make the problem considerably more difficult by choosing $n = 20\,000$ and $p = 1\,000$. We would expect a linear model selection method to miss these two nonlinear signals. Indeed, the Spike-and-Slab LASSO method (using $\lambda_1 = 0.1$ and $\lambda_0 \in \{0.1 + k \times 10 : k = 1, \ldots, 10\}$ only identifies variables x_1, x_2 and x_5. Next, we deploy the BART method with variable selection (Linero 2016) by setting `sparse=TRUE` ([145]) and 50 trees[6] in the BART software ([49]). The choice of 50 trees for variable selection was recommended in [20] and was seen to work very well. Due to the large size of the dataset, it might be reasonable to first inquire about variable selection from smaller fractions of data. We consider random subsets of different sizes $s \in \{100, 500, 1\,000\}$ as well as the entire dataset and we run BART for $M = 20\,000$ iterations. Figure 3.3 depicts BART importance measures (average number of times each variable was used in the forest). We have seen BART separating the signal from noise rather well on batches of size $s \geq 1\,000$ and with MCMC iterations $M \geq 10\,000$. The scale of the importance measure depends on s and it is not necessarily obvious where to make the cut for selection. A natural (but perhaps ad hoc) criterion would be to pick variables which were on average used at least once. This would produce false negatives for smaller s and many false positives (29 in this example) for $s = 20\,000$. The Hamming distance between the true and estimated model as well as the computing times are reported in Table 3.1. This illustrates how selection based on the importance measure is difficult to automate. While visually inspecting the

6. Results with 10 were not nearly as satisfactory.



(a) $s = 100$ (b) $s = 500$ (c) $s = 1\,000$ (d) Entire Data

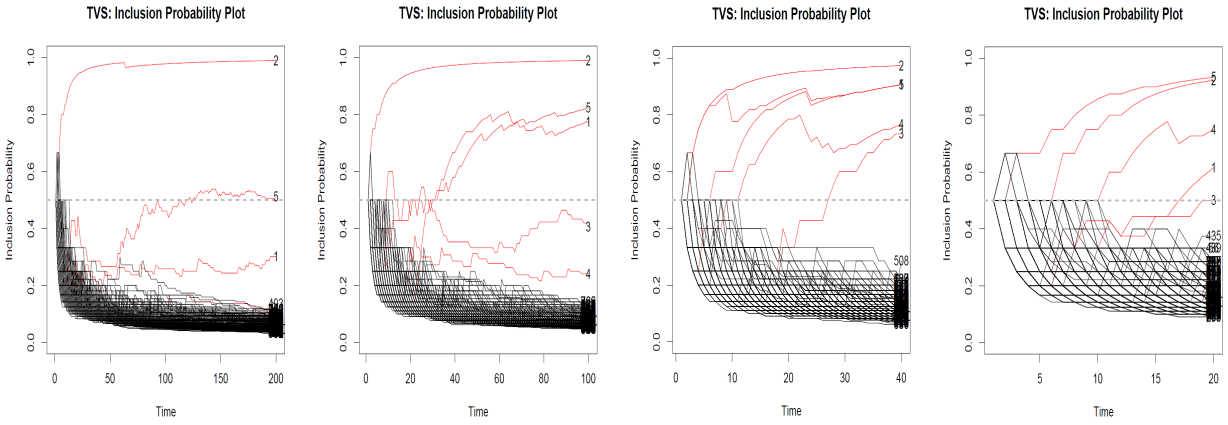
Figure 3.3: BART variable importance using $T = 20\,000$ (using 50 trees and `sparse=TRUE`) and various data subsets. Red squares (the first five covariates) are signals and black dots are noise variables.

	$s = 100$		$s = 200$		$s = 500$		$s = 1\,000$		$s = 5\,000$		$s = 10\,000$		$s = 20\,000$	
T	Time	HAM	Time	HAM	Time	HAM	Time	HAM	Time	HAM	Time	HAM	Time	HAM
5 000	6.7	4	7.7	5	11.4	3	21.5	2	103.1	3	264.2	8	794.1	16
10 000	16.2	4	16	4	23.6	2	37.8	3	213	8	549.9	18	2368.4	25
20 000	27.3	4	31.1	4	47.7	1	74.7	1	418.4	10	1090.6	21	4207.4	29

Table 3.1: Computing times (in seconds) and Hamming distance of BART on subsets of observations using all p covariates. Hamming distance compares the true model with a model obtained by truncating the BART importance measure at 1.

importance measure for $M = 20\,000$ and $s = 20\,000$ (the entire dataset) in Figure 3.3(d) is very instructive for selection, it took more than 70 minutes on this dataset. To enhance the scalability, we deploy our reinforcement learning TVS method for streaming batches of data.

Using the online local reward (3.10) with BART (with 10 trees and `sparse=TRUE` and M iterations) on batches of data $\mathcal{D}_n$ of size s. This is a weaker learning rule than the one considered in Figure 3.3 (50 trees). Choosing $s = 100$ and $M = 10\,000$, BART may not be able to obtain perfect signal isolation on a single data batch (see Figure 3.3(a) which identifies only one signal variable). However, by propagating information from one batch onto the next, TVS is able to tease out more signal (Figure 3.4(a)). Comparing Figure 3.3(b)) and Figure 3.4(c) is even more interesting, where TVS inclusion probabilities for all signals eventually cross the decision boundary after merely 40 TVS iterations. There is ultimately a tradeoff between the batch size s and the number of iterations needed for the TVS to

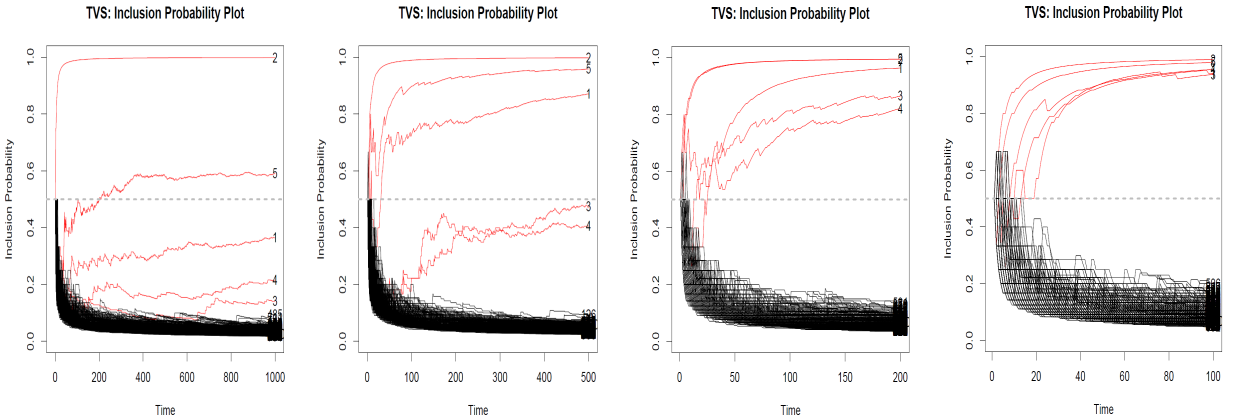


(a) $s = 100$ (b) $s = 200$ (c) $s = 500$ (d) $s = 1\,000$

Figure 3.4: TVS inclusion probabilities $T = 10\,000$ (using 10 trees and `sparse=TRUE`) and various batch sizes after single pass through the data.

stabilize. For example, with $s = 1\,000$ one obtains a far stronger learner (Figure 3.3(c)), but the separation may not be as clear after only $T = n/s = 20$ TVS iterations (Figure 3.4(d)). One can increase the number of TVS iterations by performing multiple passes through the data after bootstrapping the entire dataset and chopping it into new batches which are a proxy for future data streams. Plots of TVS inclusion probabilities after 5 such passes through the data are in Figure 3.5. Curiously, one obtains much better separation even for $s = 200$ and with larger batches ($s = 500$) the signal is perfectly separated. Note that TVS is a random algorithm and thereby the trajectories in Figure 3.5 at the beginning are slightly different from Figure 3.4. Despite the random nature, however, we have seen the separation apparent from Figure 3.5 occur consistently across multiple runs of the method.

Several observations can be made from the timing and performance comparisons presented in Table 3.2. When the batch size is not large enough, repeated runs will not help. The Hamming distance in all cases only consists of false negatives and can be decreased by increasing the batch size or increasing the number of iterations and rounds. Computationally, it seems beneficial to increase the batch size s and supply enough MCMC iterations. Variable selection accuracy can also be increased with multiple rounds.



(a) $s = 100$ (b) $s = 200$ (c) $s = 500$ (d) $s = 1\,000$

Figure 3.5: TVS inclusion probabilities $T = 10\,000$ (using 10 trees and `sparse=TRUE`) and various batch sizes after 5 passes through the data.

	$s = 100$		$s = 200$		$s = 500$		$s = 1\,000$	
	Time	HAM	Time	HAM	Time	HAM	Time	HAM
$T = 500$								
1 *round*	58.9	4	34.1	3	19.3	3	15.7	4
5 *rounds*	251.5	4	165	3	91.2	3	68.7	2
10 *rounds*	467.5	4	348.8	3	187.8	3	137.2	1
$T = 1\,000$								
1 *round*	84	4	49.8	3	29.5	3	23.6	2
5 *rounds*	411.8	4	241.5	3	140.8	1	111.4	0
10 *rounds*	870.6	3	507	2	288.2	1	224.2	0
$T = 1\,0000$								
1 *round*	541.8	3	330.9	2	220.4	0	182.8	0
5 *rounds*	2421.2	3	1501.9	2	1060.2	0	972.2	0
10 *rounds*	4841.9	3	3027.9	0	2248.3	0	2087.6	0

Table 3.2: Computing times (in seconds) and Hamming distance for TVS using different batch sizes s and BART iterations T and multiple passes through the data. The Hamming distance compares the true model with a model truncating the last TVS inclusion probability at 0.5.

3.7 Simulation Study

We further evaluate the performance of TVS in a more comprehensive simulation study. We compare TVS with several related non-parametric variable selection methods and with classical parametric ones. We assess these methods based on the following performance criteria: False Discovery Proportion (FDP) (i.e. the proportion of discoveries that are false), Power (i.e. the proportion of true signals discovered as such), Hamming Distance (between the true and estimated model) and Time.

3.7.1 Offline Cases

For a more comprehensive performance evaluation, we consider the following 4 mean functions $f_0(\cdot)$ to generate outcomes using (3.1). For each setup, we summarize results over 50 datasets of a dimensionality $p \in \{1\,000, 10\,000\}$ and a sample size $n = 300$.

- **Linear Setup**: The regressors $\boldsymbol{x}_i$ are drawn independently from $N(0, \Sigma)$, where $\Sigma = (\sigma_{jk})_{j,k=1,1}^{p,p}$ with $\sigma_{jj} = 1$ and $\sigma_{jk} = 0.9^{|j-k|}$ for $j \neq k$. Only the first 5 variables are related to the outcome (which is generated from (3.1) with $\sigma^2 = 5$) via the mean function $f_0(\boldsymbol{x}_i) = x_{i1} + 2x_{i2} + 3x_{i3} - 2x_{i4} - x_{i5}$.

- **Friedman Setup**: The Friedman setup was described earlier in Section 3.5. In addition, we now introduce correlations of roughly 0.3 between the explanatory variables.

- **Forest Setup**: We generate $\boldsymbol{x}_i$ from $N(0, \Sigma)$, where $\Sigma = (\sigma_{jk})_{j,k=(11)}^{p,p}$ with $\sigma_{jj} = 1$ and $\sigma_{jk} = 0.3^{|j-k|}$ for $j \neq k$. We then draw the mean function $f_0(\cdot)$ from a BART prior with 200 trees, using only first 5 covariates for splits. The outcome is generated from (3.1) with $\sigma^2 = 0.5$.

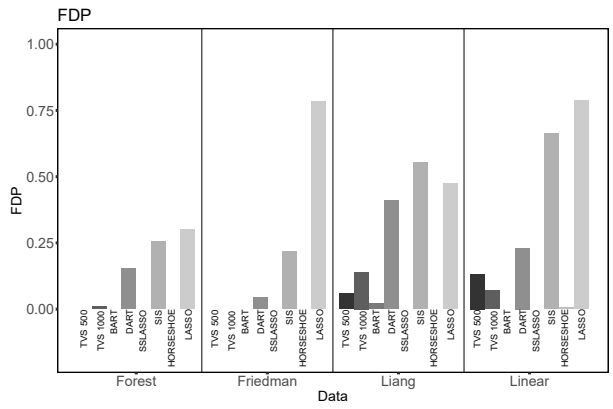
- **Liang et al (2016) Setup**: This setup was described earlier in Section 3.6.2. We now use $\sigma^2 = 0.5$.

We run TVS with $M = 500$ and $M = 1\,000$ internal BART MCMC iterations and with $T = 500$ TVS iterations. As two benchmarks for comparison, we consider the original BART method (in the R-package `BART`) and a newer variant called DART ([145]) which is tailored to high-dimensional data and which can be obtained in `BART` by setting `sparse=TRUE` (`a=1, b=1`). We ran BART and DART for $M = 50\,000$ MCMC iterations using the default prior settings with $D = 20$, $D = 50$ and $D = 200$ trees for BART and $D = 10$, $D = 50$ and $D = 200$ trees for DART. We considered two variable selection criteria: (1) posterior inclusion probability (calculated as the proportion of sampled forests that split on a given variable) at least 0.5 (see Linero (2018) and [20] for more discussion on variable selection using

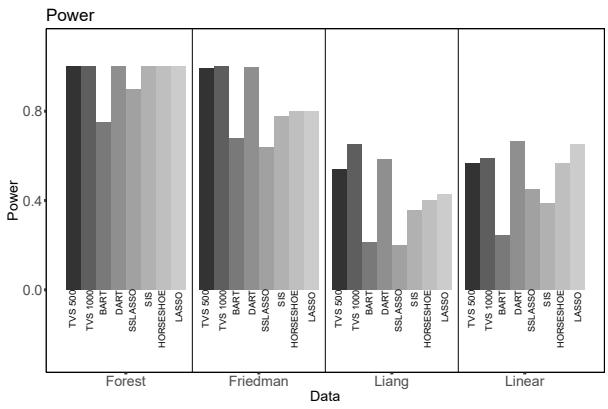
BART), (2) the average number of splits in the forest (where the average is taken over the M iterations) at least 1. We report the settings with the best performance, i.e. BART with $D = 20$ trees and DART with $D = 50$ trees using the second inclusion criterion. The third benchmark method we use for comparisons is the Spike-and-Slab LASSO ([200]) implemented in the R-package `SSLASSO` with `lambda1=0.1` and the spike penalty ranging from λ_1 to the number of variables p (i.e. `lambda0 = seq(1, p, length=p)`). We choose the same set of variable chosen by `SSLASSO` function after the regularization path has stabilized using the `model` output. We have also implemented Sure Independence Screen (SIS) ([67]) as a variable filter before applying BART variable selection[7]. Next, we applied one of the benchmark Bayesian variable selection methods, the horseshoe prior ([240, 37]) implementation from the `R-package horseshoe`. We run the Markov chain for 55 000 iterations, discarding the first 5 000 as a burnin period. Variable selection is performed by checking whether 0 is contained in the credible set (see [239]). Finally, we implement the LASSO ([234]) and report the model chosen according to the 1-s.e. rule ([74])).

We report the average performance (over 50 datasets) for $p = 10\,000$ in Figure 3.6 and the rest (for $p = 1\,000$) in the Appendix. Recall that the model estimated by TVS is obtained by truncating $\pi_i(500)$'s at 0.5. In terms of the Hamming distance, we notice that TVS performs best across-the-board. DART (with $D = 50$) performs consistently well in terms of variable selection, but the timing comparisons are less encouraging. BART (with $D = 20$) takes a relatively comparable amount of time as TVS with $M = 1\,000$, but suffers from less power. SS-LASSO's performance is strong, in particular for the less non-linear data setups. The performance of TVS is seen to improve with M. SIS is a screening method based on a linear model assumption. Albeit faster than TVS, we observe that SIS generally over-selects. In addition, SIS screens out key signals with non-linear effects (Liang and Friedman datasets). The Horseshoe prior performs very well but is much slower. LASSO seems to overfit, in spite of the 1-se rule.

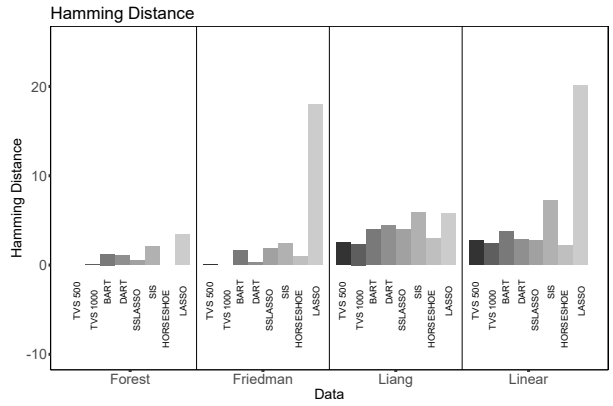
7. SIS on its own yielded too many false positives.



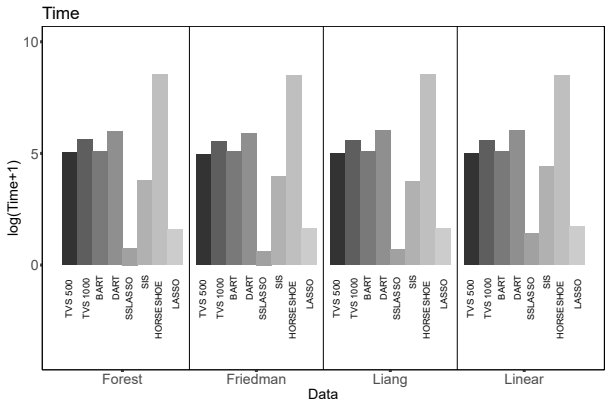
(a) FDP



(b) Power



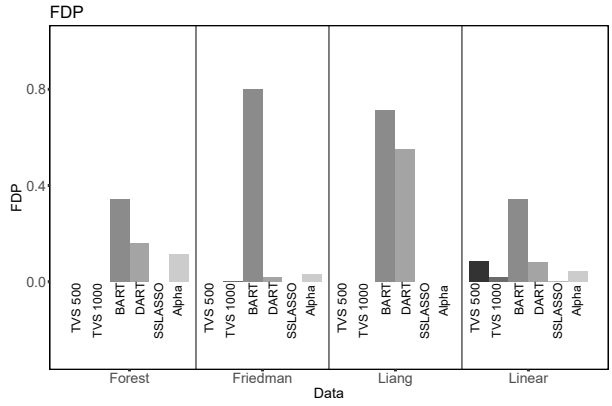
(c) Hamming Distance



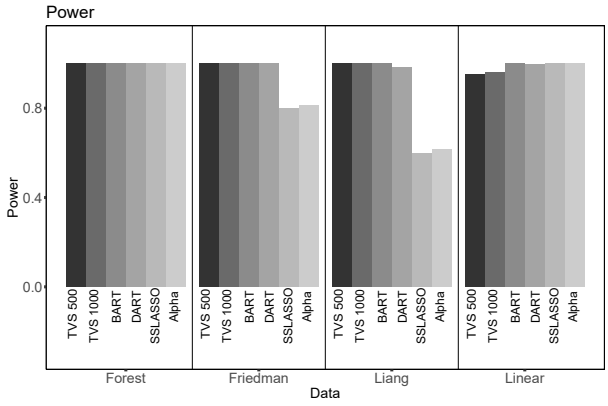
(d) Time

Figure 3.6: Graphs denoting FDP (3.6(a)), Power (3.6(b)), Hamming Distance (3.6(c)), and Time (3.6(d)) for the 4 choices of f_0 assuming $p = 10\,000$ and $n = 300$. The x-axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1\,000$.

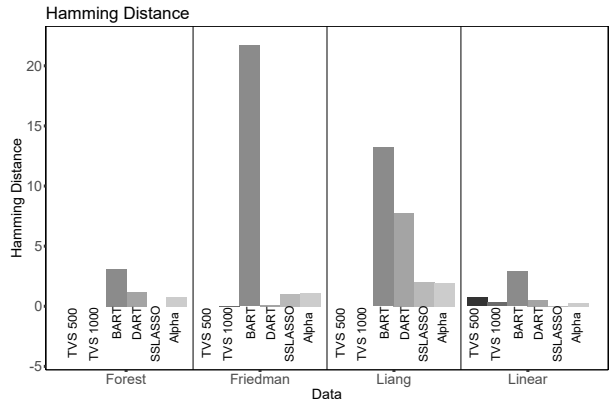
We also implement a stopping criterion for TVS based on the stabilization of the inclusion probabilities $\pi_i(t) = \frac{a_i(t)}{a_i(t)+b_i(t)}$. One possibility is to stop TVS when the estimated model $\widehat{\mathcal{S}}_t$ obtained by truncating $\pi_i(t)$'s at 0.5 has *not* changed for over, say, 100 consecutive TVS iterations. With this convergence criterion, the convergence times differs across the different data set-ups. Generally, TVS is able to converge in $\sim$ 200 iterations for $p = 1\,000$ and $\sim$ 300 iterations for $p = 10\,000$. While the computing times are faster, TVS may be more conservative (lower FDP but also lower Power). The Hamming distance is hence a bit larger, but comparable to TVS with 500 iterations (Appendix C.1)



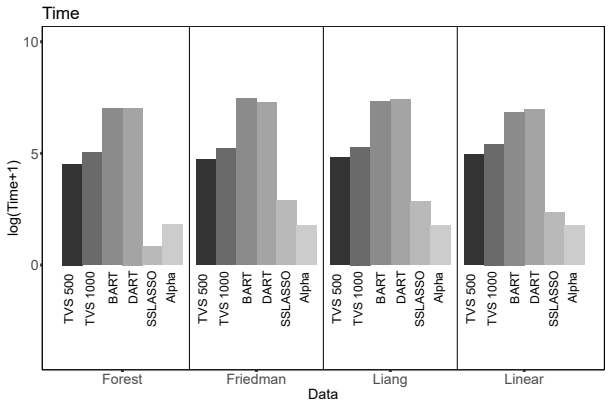
(a) FDP



(b) Power



(c) Hamming Distance



(d) Time

Figure 3.7: Graphs denoting FDP (3.7(a)), Power (3.7(b)), Hamming Distance (3.7(c)), and Time (3.7(d)) for the 4 choices of f_0 assuming $p = 1\,000$ and $n = 10\,000$. The x-axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1\,000$, both with $s = 1000$.

3.7.2 Online Cases

We now consider a simulation scenario where $n >> p$, i.e. $p = 1\,000$ and $n = 10\,000$. As described earlier in Section 3.6.2, we partition the data into minibatches $(\boldsymbol{Y}^{(b)}, \boldsymbol{X}^{(b)})$ of size s, where $\boldsymbol{Y}^{(b)} = (Y_i : (b-1)s + 1 \leq i \leq bs)$ and $\boldsymbol{X}^{(b)} = [\boldsymbol{x}_i : (b-1)s + 1 \leq i \leq bs]'$ for $b = 1, \ldots, n/s$ with $s \in \{500, 1000\}$ and $M \in \{500, 1000\}$ using $D = 10$ trees. In this study, we consider the same four set-ups as in Section 3.7.1. We implemented TVS with a fixed number of rounds $r \in \{1, 5, 10\}$ and a version with a stopping criterion based on the stabilization of the inclusion probabilities $\pi_i(t) = \frac{a_i(t)}{a_i(t)+b_i(t)}$. This means that TVS

will terminate when the estimated model $\widehat{\mathcal{S}}_t$ obtained by truncating $\pi(t)$'s at 0.5 has not changed for 100 consecutive iterations. The results using the stopping criterion are reported in Figure 3.7 and the rest is in the Appendix (Section B.2.2). As before, we report the best configuration for BART and DART, namely $D = 20$ for BART and $D = 50$ for DART (both with 50 000 MCMC iterations). For both methods, there are non-negligible false discoveries and the timing comparisons are not encouraging. In addition, we could not apply both BART and DART with $n \geq 50\,000$ observations due to insufficient memory. For TVS, we found the batch size $s = 1\,000$ to work better, as well as running the algorithm for enough rounds until the inclusion probabilities have stabilized (Figure 3.7 reports the results with a stopping criterion). The results are very encouraging.

Streaming feature selection methods are still being developed ([242, 63, 189, 111]). We have compared our online variant with the α-investing method for streaming variable selection of [253]. This method dynamically adjusts p-value thresholds for adding features to the model. We run this method using p-values from applying linear regression on batches of streaming data. The results are shown in Figure 3.7. This procedure performs well in Forest and Linear set-ups but misses the key non-linear variables in Friedman and Liang setups. While SSLASSO's performance is very strong, we notice that in the non-linear setup of [138] there are false non-discoveries.

3.8 Application on Real Data

3.8.1 HIV Data

We will apply (offline) TVS on a benchmark Human Immunodeficiency Virus Type I (HIV-I) data described and analyzed in [195] and [8]. This publicly available dataset consists of genotype and resistance measurements (decrease in susceptibility on a log scale) to three drug classes: (1) protease inhibitors (PIs), (2) nucleoside reverse transcriptase inhibitors (NRTIs), and (3) non-nucleoside reverse transcriptase inhibitors (NNRTIs).

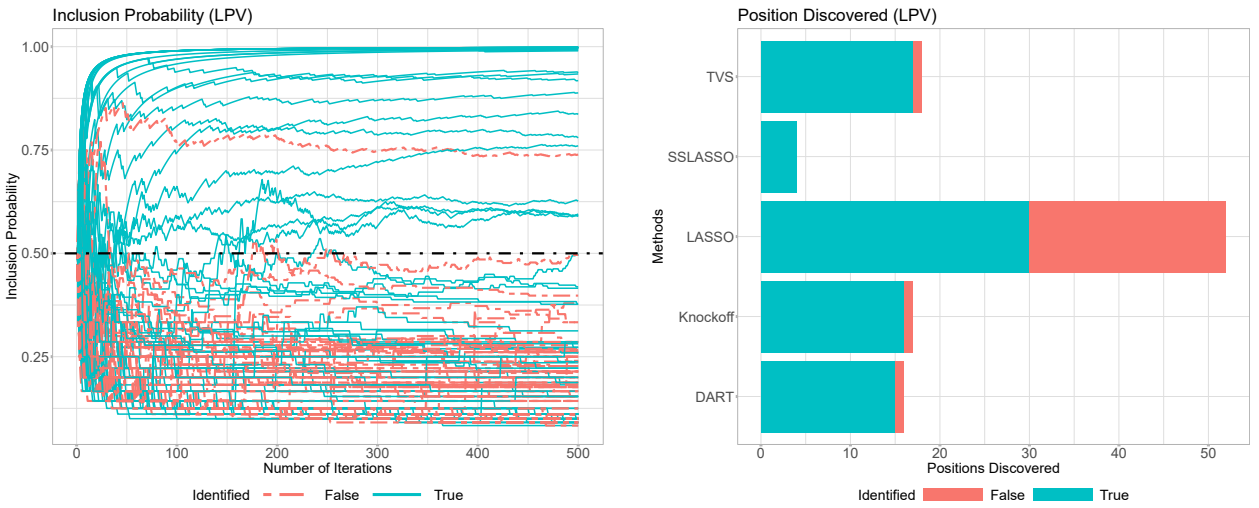
The goal in this analysis is to find mutations in the HIV-1 protease or reverse transcriptase that are associated with drug resistance. Similarly as in [8] we analyze each drug separately. The response Y_i is given by the log-fold increase of lab-tested drug resistance in the i^{th} sample with the design matrix X consisting of binary indicators $x_{ij} \in \{0, 1\}$ for whether or not the j^{th} mutation has occurred at the i^{th} sample.[8]

In an independent experimental study, [194] identified mutations that are present at a significantly higher frequency in virus samples from patients who have been treated with each class of drugs as compared to patients who never received such treatments. While, as with any other real data experiment, the ground truth is unknown, we treat this independent study are a good approximation to the ground truth. Similarly as [8], we only compare mutation positions since multiple mutations in the same position are often associated with the same drug resistance outcomes.

For illustration, we now focus on one particular drug called Lopinavir (LPV). There are $p = 206$ mutations and $n = 824$ independent samples available for this drug. TVS was applied to this data for $T = 500$ iterations with $M = 1\,000$ inner BART iterations. In Figure 3.8, we differentiate those mutations whose position were identified by [194] and mutations which were not identified with blue and red colors, respectively. From the plot of inclusion probabilities in Figure 3.8(a), it is comforting to see that only one unidentified mutation has a posterior probability $\pi_j(t)$ stabilized above the 0.5 threshold. Generally, we observe the experimentally identified mutations (blue curves) to have higher inclusion probabilities.

Comparisons are made with DART, Knockoffs [8], LASSO (10-fold cross-validation), and Spike-and-Slab LASSO ([200]), choosing $\lambda_1 = 0.1$ and $\lambda_0 \in \{\lambda_1 + 10 \times k;\ k = 0, 1, \ldots, p\}$). Knockoffs, LASSO, and the Spike-and-Slab LASSO assume a linear model with no interactions. DART was implemented using $T = 50$ trees and 50 000 MCMC iterations, where we select those variables whose average number splits was at least one. The numbers of dis-

8. As suggested in the analysis of [8], when analyzing each drug, only mutations that appear 3 or more times in the samples are taken into consideration.



(a) TVS Inclusion Probabilities

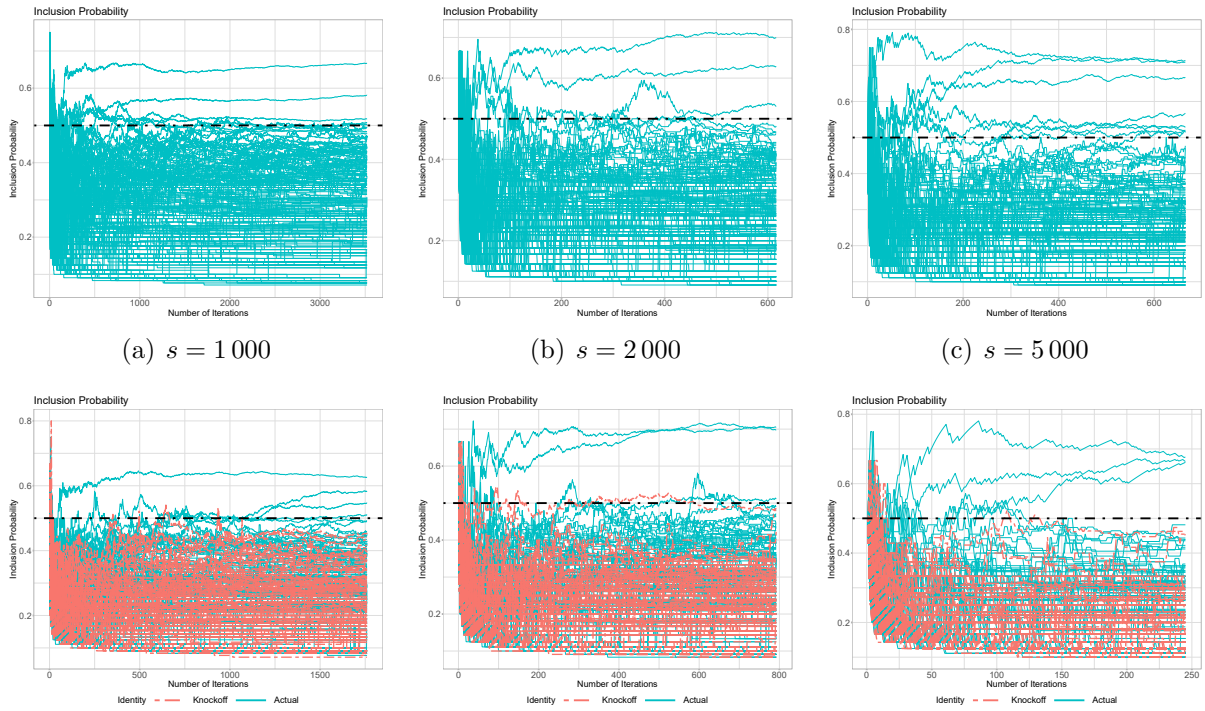
(b) Comparisons with other Methods

Figure 3.8: Thompson Variable Selection on the LPV dataset. (Left) Trajectories of the inclusion probabilities $\pi_j(t)$. (Right) Number of signals discovered, where blue denotes the experimentally validated signals and red are the unvalidated ones.

covered Positions for each method are plotted in Figure 3.8(b). While LASSO selects many more experimentally validated mutations, it also includes many unvalidated ones. TVS, on the other hand, has a very small number of "false discoveries" while maintaining good power. Additional results are included in the Appendix (Section B.3).

3.8.2 Durable Goods Marketing Data Set

Our second application examines a cross-sectional dataset described in Ni et al. (2012) consisting of durable goods sales data from a major anonymous U.S. consumer electronics retailer. The dataset features the results of a direct-mail promotion campaign in November 2003 where roughly half of the $n = 176\,961$ households received a promotional mailer with 10$ off their purchase during the promotion time period (December 4-15). If they did purchase, they would get 10% off on a subsequent purchase through December. The treatment assignment was random. The data contains $p = 146$ descriptors of all customers including prior purchase history, purchase of warranties etc. We will investigate the effect of the promotional campaign (as well as other covariates) on December sales. In addition,



(a) $s = 1\,000$ (b) $s = 2\,000$ (c) $s = 5\,000$

(d) $s = 1\,000$ (e) $s = 2\,000$ (f) $s = 5\,000$

Figure 3.9: Thompson Variable Selection on the marketing data. (Left) Trajectories of the inclusion probabilities $\pi_j(t)$ without knockoffs. (Right) Trajectories of the inclusion probabilities $\pi_j(t)$ with knockoffs (in red).

we will interact the promotion mail indicator with customer characteristics to identify the "mail-deal-prone" customers.

We dichotomized December purchase (in dollars) to create a binary outcome $Y_i = \mathbb{I}(\texttt{December-sales}_i > 0)$ for whether or not the i^{th} customer made any purchase in December. Regarding predictor variables, we removed any variables with missing values and any binary variables with less than 10 samples in one group. This pre-filtering leaves us with 114 variables whose names and descriptive statistics are reported in Section B.4 in the Appendix. We interact the promotion mail indicator with these variables to obtain $p = 227$ predictors. Due to the large volume of data ($n \approx 180\,000$), we were unable to run DART and BART (`BART` package implementation) due to memory problems. This highlights the need for TVS as a variable selector which can handle such voluminous data.

Unlike the HIV-I data in Section 3.8.1, there is no proxy for the ground truth. To under-

	$s = 1\,000$		$s = 2\,000$		$s = 5\,000$	
Knockoff	Yes	No	Yes	No	Yes	No
total number of medium ticket items in previous 60 months	0.30	**0.51**	**0.51**	0.46	0.44	**0.51**
total number of small ticket items in previous 60 months	0.49	**0.52**	0.40	0.41	0.25	**0.53**
number of months shopped once in previous 12 months	0.34	0.41	0.44	0.42	0.41	**0.57**
number of months shopped once in previous 24 months	**0.63**	**0.67**	**0.70**	**0.63**	**0.66**	**0.71**
count of unique purchase trips in previous 24 months	**0.55**	0.26	0.48	**0.53**	**0.68**	**0.67**
total number of items purchased in previous 12 months	**0.51**	0.30	0.24	0.21	0.24	0.11
promo_nov period: total sales	**0.58**	**0.58**	**0.71**	**0.70**	0.33	0.45
mailed in holiday 2001 mailer	0.25	0.43	0.08	0.41	0.42	**0.52**
percent audio category sales of total sales × mail indicator	0.41	**0.50**	0.15	0.28	0.13	0.10
promo_nov period: total sales × mail indicator	0.15	0.35	0.46	0.41	**0.66**	**0.71**
indicator of holiday mailer 2002 promotion response × mail indicator	0.19	0.38	0.44	0.21	0.44	**0.52**

Table 3.3: Variables Selected by TVS with different s and with/without knockoff. The numbers report conditional inclusion probabilities $\pi_i(t)$ after convergence, where values above 0.5 are in bold.

stand the performance quality of TVS, we added 227 normally distributed knockoffs. The knockoffs are generated using `create.second_order` function in the `knockoff` R package ([180]) using a Gaussian Distributions with the same mean and covariance structure ([34]). We run TVS with a batch size $s \in \{1\,000, 2\,000, 5\,000\}$ and $M = 1\,000$ inner iterations until the posterior probabilities have stabilized. The inclusion probabilities are plotted in Figure 3.9 for two cases (a) without knockoffs (the first row) and (b) with knockoffs (the second row). It is interesting to note that, apart from one setting with $s = 1\,000$, the knockoff trajectories are safely suppressed below 0.5 (dashed line). Both with and without knockoffs, TVS chooses 'the number of months with purchases in past 24 month' and 'the November Promotion Sales' as important variables. The selected variables for each combination of settings are summarized in Table 3.3.

Finally, we used the same set of variables (including knockoff variables) for different variable selection methods and recorded the number of knockoffs chosen by each one. We used BART ($D = 20$, MCMC iteration $= 50\,000$), and DART ($D = 50$, MCMC iteration $= 50\,000$) with the same selection criteria as before, i.e. a variable is selected if it was split on average at least once. We also consider LASSO where the sparsity penalty λ was chosen by cross-validation. BART and DART cannot be run on the entire data set so we only run it on a random subset of 10 000 data points. While TVS with large enough s does not include

any of the knockoffs, LASSO does include 14 and DART includes 4.

3.9 Discussion

Our work pursues an intriguing connection between spike-and-slab variable selection and binary bandit problems. This pursuit has lead to a proposal of Thompson Variable Selection, a reinforcement learning wrapper algorithm for fast variable selection in high dimensional non-parametric problems. In related work, [148] developed an ABC sampler for variable subsets through a split-sample approach by (a) first proposing a subset $\mathcal{S}_t$ from a prior, (b) keeping only those subsets that yield pseudo-data that are sufficiently close to the left-out sample. TVS can be broadly regarded as a reinforcement learning elaboration of this strategy where, instead of sampling from a (non-informative) prior $\pi(\mathcal{S}_t)$, one "updates the prior $\pi(\mathcal{S}_t)$" by learning from previous successes.

TVS can be regarded as a stochastic optimization approach to subset selection which balances exploration and exploitation. TVS is suitable in settings when very many predictors and/or very many observations can be too overwhelming for machine learning. By sequentially parsing subsets of data and reinforcing promising covariates, TVS can effectively separate signal from noise, providing a platform for interpretable machine learning. TVS minimizes regret by sequentially computing a median probability model rule obtained by truncating sampled mean rewards. We provide bounds for this regret without necessarily assuming that the mean arm rewards be unrelated. We observe strong empirical performance of TVS under various scenarios, both on real and simulated data. The TVS approach, coupled with BART, captures non-linearities and may thus be beneficial over linear techniques. In addition, TVS scales to very large datasets.

Thompson sampling is ultimately designed to minimize regret, not to select the best arms. [28] point out that algorithms satisfying regret bounds of order $\log(T)$ can be far from optimal for finding the best arm. [208] proposed a 'top-two' sampling version of Thompson sampling which is tailored for single best-arm identification. Alternative algorithms have

been proposed including the Successive Elimination algorithm of [62] for single top-arm identification and the SAR (Successive Accepts Rejects) algorithm of [29] for top m-arm identification. These works suggest possible refinements of our approach for directly targeting multiple best arms in correlated combinatorial bandits.

CHAPTER 4

A SYSTEM BIOLOGY PROBLEM WITH A DEEP LEARNING SOLUTION

4.1 Abstract

The universal expressibility assumption of Deep Neural Networks (DNNs) is the key motivation behind recent works in the system biology community to employ DNNs to solve important problems in functional genomics and molecular genetics. Typically, such investigations have taken a "black box" approach in which the internal structure of the model used is set purely by machine learning considerations with little consideration of representing the internal structure of the biological system by the mathematical structure of the DNN. DNNs have not yet been applied to the detailed modeling of transcriptional control in which mRNA production is controlled by the binding of specific transcription factors to DNA, in part because such models are in part formulated in terms of specific chemical equations that appear different in form from those used in neural networks. In this paper, we give an example of a DNN which can model the detailed control of transcription in a precise and predictive manner. Its internal structure is fully interpretable and is faithful to underlying chemistry of transcription factor binding to DNA. We derive our DNN from a systems biology model that was not previously recognized as having a DNN structure. Although we apply our DNN to data from the early embryo of the fruit fly *Drosophila*, this system serves as a testbed for analysis of much larger data sets obtained by systems biology studies on a genomic scale.

This chapter has been published as a paper in Bioinformatics [147]. It was written together with Professor John Reinitz. In this paper I focus on the ideas and data experiments. Kenneth Barr provides key insights into how model functions an specific details associated with each model. Figure 4.1 is a live image taken by Kenneth in one of his experiments.

4.2 Introduction

A central unsolved problem in biology is to understand how DNA specifies how genes turn on and off in multicellular organisms. The "universal expressibility" of deep neural nets (DNNs), combined with standardized and effective optimization methods such as stochastic gradient descent (SGD), suggests that they might be a valuable tool in this undertaking. At the same time, the applicability and acceptance of DNNs in solving problems in natural science has been limited by the uninterpretability of their internal computations. We address both of these areas in this report by describing the reimplementation of a specific and highly predictive model of transcriptional control as a DNN. This work is a demonstration that DNN techniques can be applied to the apparently purely chemical problem of transcriptional control. Although applied here to a very specific problem in early embryo of *Drosophila*, we discuss below the reasons why the results here should generalize well to the study of transcription in other organism including humans. The work presented here also provides an example of a DNN in which the internal structure is well understood and which may be of value to the machine learning field.

Deep learning has been widely deployed in genomics and systems biology over the last few years ([54, 4, 140, 185, 42, 218, 207, 93, 168, 167, 7]). Many of the developed tools have been highly successful in classification problems such as the identification of binding sites, open regions of chromatin, and the location of enhancers. Deeper understanding requires more quantitative studies. One recent example that goes beyond classification concerns a fully quantitative and highly predictive DNN model of the role of untranslated RNA leader sequences in gene expression in yeast ([54]), We believe that these studies have two sets of limitations. First, they take a universal expressibility approach without much understanding and interpretation of the underlying chemical and biological mechanisms giving rise to to the phenomena under study. This limits the contributions DNNs can make to increasing human understanding of fundamental biological processes. Here we consider a DNN in which each layer has a specific chemical or biological interpretation. Studies of regulatory DNA with

DNNs have treated only the sequence itself, but in metazoa (multicellular animals) different cell types have very different gene expression states although they contain the same DNA. For example, one recent study used a sequence based DNN classifier of open and closed chromatin regions to find binding sites for tissue specific transcription factors, but the actual binding state of these factors was not part of the model ([168]). Here we consider a DNN in which state is described not only by the sequence, but also by the set of regulatory proteins present in each cell, and the state of binding of each factor constitutes a specific layer of the model. We now describe the problem to be solved in both mathematical and biological terms.

The "expression" of a gene is the rate at which it synthesizes its ultimate product, which may be an RNA or protein. Here we focus on genes transcribed by RNA polymerase II, which transcribes all genes other than those coding for ribosomal or transfer RNA. Although the expression of these genes can be affected at the level of RNA splicing or translation of protein, regulation of expression takes place chiefly at the level of transcriptional control. We thus consider the rate at which mRNA (the "transcript") is synthesized from a complementary DNA template. This rate $d[\text{mRNA}]/dt = f(\boldsymbol{D}, \boldsymbol{v})$ where $\boldsymbol{D} = (D_i)$, where each $D_i \in \{A, C, G, T\}$ is a base in the sequence of regulatory DNA, and $\boldsymbol{v} = (v_1, \ldots, v_a, \ldots, v_n)$, where each v_a is the nuclear concentration of a regulatory DNA binding protein known as a "transcription factor" (TF). The machine learning task is to learn the function f from a series of observations $(\boldsymbol{D}_j, \boldsymbol{v}_{jk})$ of the expression of sequence j in cell type k, where $k \in \{1, \ldots, M\}$. The essence of the scientific problem is that each sequence $\boldsymbol{D}_j$ must express correctly in each cell type, reflecting the fact that in a multicellular organism, different sets of genes are expressed in different cell types, but each cell type contains the same DNA.

Regulatory DNA is noncoding DNA which can be upstream (5'), downstream (3'), or within (intronic) the complementary mRNA template. TFs bind to DNA in its double stranded form. In metazoans, regulatory DNA is frequently much larger than the coding portion of the gene. The regulatory DNA contains segments of 500 to 1000 base pairs (bp) called "enhancers", each of which directs expression in a particular domain or tissue type.

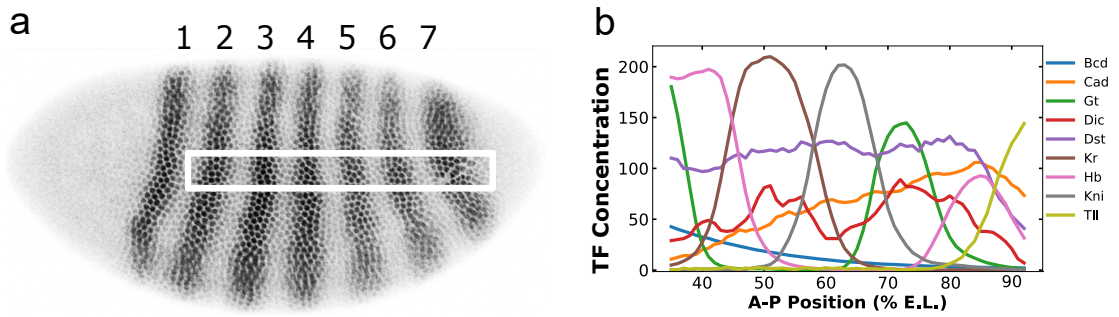


Figure 4.1: (a) shows a *Drosophila* embryo about 3 hours after fertilization which has been stained for Eve protein as described ([225]). Anterior is to the left and dorsal is up. The dark shades indicate the concentration of Eve protein and the stripes are numbered. The white box is the one dimensional region of interest used to generate the data ([110]). (b) The TF concentrations found across the embryo ([225]). In the graph, 58 data points are shown, corresponding to 58 nuclei on the A-P axis. Each nucleus is 1 % E.L. in size. The identity of TFs is shown in the key; the horizontal axis shows position in percent egg length (E.L.) and the vertical axis shows protein concentration.

In this study, we consider a gene called *eve* which acts in the early embryo of the fruit fly *Drosophila melanogaster*, at which time it forms a pattern of seven stripes as shown in Figure 4.1. The entire gene is 16.5 kilobases (kb) of DNA in length ([78]), but the mRNA transcript is only 1.5 kb (Figure 4.2). We consider the action of *eve* from 1 to 3 hours after the start of embryonic development. At this time the embryo is a hollow ellipsoid of cell nuclei that can be treated like a naturally grown gene chip in which $d[\text{mRNA}]/dt$, $\boldsymbol{D}_j$, and $\boldsymbol{v}_{jk}$ are fully observable at a quantitative level. The embryo contains two orthogonal axes in the anterior-poster (A-P) and dorsal-ventral (D-V) directions. In the central portion of the embryo, gene expression on the two axes is uncoupled, so cell type and gene expression can be visualized by plotting relevant state variables in one dimension.

We have chosen a specific dataset involving the expression driven by fused and unfused enhancers for *eve* stripes 2 and 3 in normal and revered orientation (see figure 4.2). Because the fused enhancers drive a novel pattern compared to the unfused enhancers, this dataset provides both a stringent and highly informative test of capabilities of theoretical models. In particular, it is known that *eve* stripe 2 is repressed at its borders by TFs known as quenchers which operate over a range of 150 bps ([223, 220, 92]). Multiple quenchers must

bind in order to elicit repression. It is known that Bicoid(Bcd) and D-STAT(Dst) are key activators of stripes 2 ([220]) and 3 ([221]) respectively, and that Bcd binding to DNA involves cooperativity ([31, 130]). Moreover, it is known that Hunchback(Hb) acts as a repressor on stripe 3 ([221]) and as an activator on stripe 2 ([220]), and that it is converted from a quencher to an activator by nearby sites occupied by Bcd or Caudal (cad) ([219]). As we explain below, this dense set of prior knowledge is not required for general modelling of transcription, but it provides a useful tool for selecting a particular thermodynamic model for reimplementation as a DNN.

In this paper, we show that DNNs can be used to generate a predictive model of gene expression. We chose a model for reimplementation as a DNN as follows. The ability to predict gene expression from DNA sequence is limited to so-called thermodynamic transcription models ([191, 109, 214, 116, 64, 98, 119, 161, 209, 211, 18, 14, 13]). These models are defined by the fact that occupancy of DNA by TFs is calculated using thermodynamics, and phenomenological rules are used to calculate the transcription rate from the configuration of bound factors. Thus, all thermodynamic model compute the occupancy of TF binding sites from the concentration of TFs and DNA sequences, and are hence capable of predicting expression from DNA sequences not used for learning. Thermodynamic models can be further classified according to which chemical and regulatory mechanisms are included. Not all thermodynamic models consider the role of cooperativity in DNA binding([109, 214, 116, 64, 98, 161, 209, 211, 18]), but some do ([214, 119, 14, 13]). It is our experience (Kim and Reinitz unpublished data) that failure to consider the cooperativity of Bcd produces pattern defects in stripes 2 and 3 using the data set considered here. In addition, thermodynamic models differ in terms of the processing steps performed after calculating the state of TF binding to DNA. The simplest sum activation and repression in a manner that does not take into account the limited range of action of quenchers, ([214, 116, 18]). Such models cannot account for the narrow interstripes between the stripes of *eve* expression. Because *eve* stripes are known to be formed by repression by gap genes,

the limited range of gap gene repression prevents repression at one enhancer from repressing expression at a nearby enhancer: thus, Hb bound at 3 enhancer is unable to repress expression from the stripe 2 enhancer. Many published models take into account the short range effect of quenchers. ([191, 109, 64, 98, 161, 211]), but do not include in addition the ability of repressors to be transformed into activators by other bound factors [119, 13]. The ability to model coactivation is essential for *eve* because it is necessary for Hb to function as a coactivated activator for stripe 2. Indeed, the absence of coactivation is the probable reason why one effort to model many enhancers failed to generate any expression from stripe 2 ([214]). Among models of individual enhancers, we chose [119] for reimplementation as an DNN because it has a sufficiently rich set of mechanisms with which to model stripes 2 and 3 and is thus a suitable test bed. It is the simplest model for such testing in that it does not contain additional mechanisms required to model a whole locus rather than an enhancer ([209, 14]).

Like DNNs, thermodynamic transcription models have a feedforward structure that can be described in layers. Unlike DNNs, thermodynamic models arise from chemistry and at least superficially have a different mathematical structure than DNNs. The form of the resulting equations makes back propagation and hence SGD difficult or impossible because of the need to hand code complex partial derivatives, so these models are optimized by zero order methods such as Simulated Annealing or Genetic Algorithms. Despite this apparent mathematical distinction, we were able to translate the chemical model of transcription into a standard DNN form that could perform rapid learning by SGD.

Thermodynamic transcription models differ from existing DNNs used in genomics and systems biology because they compute the different levels of mRNA expression in different cell types by computing it from regulatory DNA sequence and TF concentration. They also differ from purely phenomenological models that do not use any sequence information; such models have been applied to the dynamics of a small network using a variant of a recurrent Hopfield neural net ([192, 108, 157]), and to the expression of enhancers using

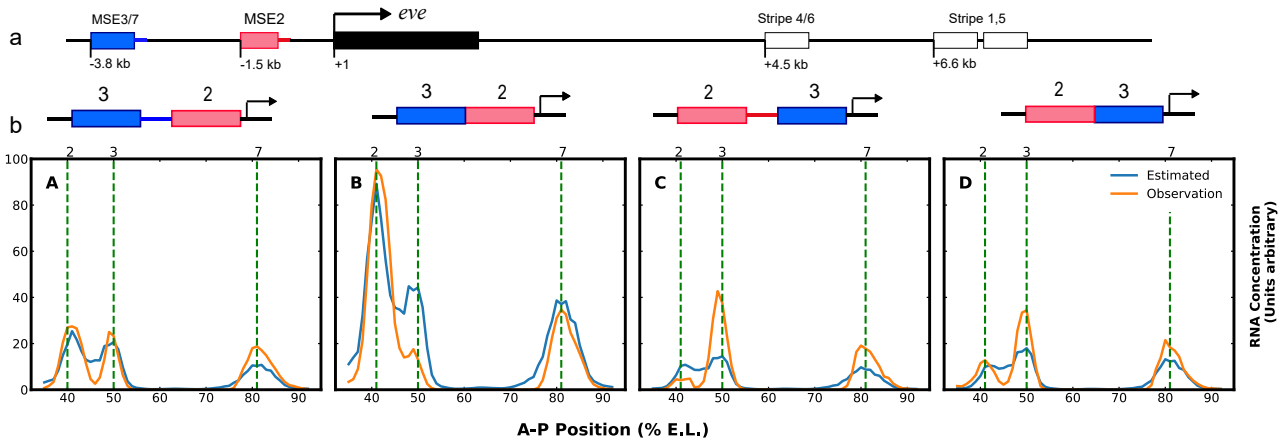


Figure 4.2: (a): The figure shows a diagram of the *eve* locus. The transcript is indicated by black box; enhancers are indicated by pink, blue or white boxes and are labeled with the *eve* stripe that they drive. (b): Fusion constructs that are used in the training process. The blue box represents the MSE3/7 enhancer and the red box represents the MSE2 enhancer. The pair of constructs over graphs A and B differ by only 358 pairs of spacer between MSE3 and MSE2, and those above graphs C and differ by only 155bp. The four graphs in (b) shows the data used for training and outcome from the model after 200 epochs using Adam. In each graph, 58 data points are shown, corresponding to 58 nuclei on the A-P axis. Each nucleus is 1 % E.L. in size. For reference, the quantitated average positions of stripes 2, 3, and 7 ([14] and see Figure 4.1) are shown by vertical dashed lines labeled by stripe identity. The orange lines show the observed data from the experiments ([119]) and the blue lines show the model output. The presence or absence of the spacers between MSE2 and MSE3 cause a large difference in the expression pattern generated. The overall RMSE is of the training is calculated to be 6.89; This is on the order of the experimental error in the data [119]. The R^2 of this fit is 0.83

logistic regression ([104]). Models in this class cannot predict gene expression from DNA sequence.

The resulting model is, to our knowledge, one of the first fully interpretable DNNs with an exact interpretation for each of the unknown parameters. It is also an example of a biologically validated DNN that is not perceptron based. The parameters used to calculate the position and affinity of DNA binding sites for TFs (see Section 4.3.1) are obtained from independent experiments, resulting in a very small number of parameters for an extremely deep network.

Below, in Section 4.3, we will describe the function of each layer and the interpretation of

parameters. In section 4.4, we focus on the training and evaluation of performance. Finally, in section 4.5, we discuss the scientific implications of our results.

4.3 Understanding each layer

The model's input is DNA sequence and TF concentration. The TFs have functional roles which, in the present application, are known from independent experiments. Activators activate transcription, quenchers suppress the action of activators over a limited range, and certain activators can also convert nearby quenchers into activators. In each of these regulatory mechanisms, multiple bound TFs are required to perform a regulatory action. We perform this calculation as follows. Binding site locations and affinities are determined from sequence. Together with TF concentrations, this information is used to calculate the occupancy of each binding site. We then calculate the effects of coactivation, followed by the effects of quenching. The total amount of remaining activation is then summed and passed through a thresholding function. Each step in this calculation corresponds to one or more of the layers shown in Figure 4.4. We describe these layers below.

4.3.1 Computing Fractional Occupancy

Identifying the binding sites

An indicator representation of DNA sequence is used as input. The column index is the base pair position number in the sequence. The row index indicates which of the 4 bases (A,C,G,T) is observed. For example, if we have a sequence of ACTTGTTA, the corresponding matrix is

$$\begin{pmatrix} \text{A} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \text{C} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{G} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \text{T} & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

The nine TFs of interest are Bicoid (Bcd), Caudal (Cad), Drosophila-STAT (Dst), Dichaete (Dic), Hunchback (Hb), Krupple (Kr), Knirps (Kni), Giant (Gt), and Tailless (Tll) (Figure 4.1b). The identification and affinity characterization of binding sites for TF a requires a (PWM_a). Chemically, the PWM represents an additive model of binding in which the Gibbs free energy ΔG of binding is the sum of the free energies of binding to each nucleotide. Statistically, the PWM score can be regarded as the likelihood of finding a binding site at a given position, calculated using a variational approximation of the likelihood using only the marginal likelihood of each base. The physical size of the binding site, which is important for calculating competitive interactions between TFs, can usually be read from DNAse I footprints [220, for example], and is larger than the set of nucleotides that are specifically recognized by the TF. We take this fact into account by padding the edges of the PWM with zeros to reflect their physical size. Most of our Position Weight Matrices (PWMs) were obtained from high quality SELEX([204, 176]) or bacterial one-hybrid data ([171]); complete details are given elsewhere ([119]). Because these data were both reliable and independent, we chose not to allow adjustments of PWM values to fit the data, as has been done in other studies ([214]). In the context of DNNs, PWMs can be understood as a convolution kernel in which the number of columns represent the number of nucleotides of DNA in physical contact with a bound TF. Unlike many convolution matrices used in deep learning, PWMs make direct experimental predictions about DNA properties, a fact used in a previous study to experimentally confirm PWMs obtained by deep learning techniques ([54]).

The resulting score, denoted by $S_{i,i+m;\,a}$, is an affine transformation of the the free energy $\Delta G_{i,i+m;\,a}$ of TF a binding to a site extending from base i to base $i+m$. In many cases, including most of the TFs considered here, m can be read off directly from DNAse I footprints ([220, for example]). TFs physically bind in the major groove of the DNA double helix but the PWM is convolved with only a single strand. We compensate for this fact by also scoring the complementary strand, in which bases are replaced by their complements

(A $\rightarrow$ T; C $\rightarrow$ G; T $\rightarrow$ A; G $\rightarrow$ C), and orientation is reversed. Scoring each strand results in a $1 \times n$ array of scores S_a, where n is the sequence length. At each base position, we set the score to be the larger of the two scores at that position.

We next calculate the equilibrium affinity $K_{i,i+m;\,a}$ of each binding site. If $\Delta G_{i,i+m;\,a}$ were known in units of kcal/mole, then $K_{i,i+m;\,a} = \exp(\Delta G_{i,i+m;\,a}/RT)$, where R is Boltzmann's constant and T the absolute temperature. $S_{i,i+m;\,a}$ is related to $\Delta G_{i,i+m;\,a}$ by an affine transformation $ax+b$, or more precisely, $S-S_{\max}/\lambda$ in which $S_{\max}$ is found from experiment and λ is learned by training as follows. Here, $S_{\max;\,a}$ is the maximum possible score for any particular TF a. $S_{\max,a}$ is determined experimentally from the tightest possible binding sequence. We only consider binding to sites with scores greater than zero, so we indicator function to assure that below threshold sites disappear, so that

$$K_{i,i+m;\,a} = \exp\left(-\frac{(S_{i,i+m;\,a} - S_{\max;\,a})}{\lambda_a}\right) \mathbb{I}(S_{i,i+m;\,a} > 0), \tag{4.1}$$

where λ_a is a learnable parameter for each TF a. The indicators, one for each TF a, give rise to what is essentially a `ReLU`, and is hence denoted as "ReLU Exponential Activation" in Figure 4.4. The $K_{i,i+m;\,a}$, arranged according to positions on the DNA sequence, produce another $1 \times (n-m)$ vector K_a. Adjusting for the size of m, we further concatenate K_a for all TFs to produce K, which we use for the calculation of fractional occupancy.

Belief propagation and partition function computation

Moving from affinity $K_{i,i+m}$ to fractional occupancy requires the consideration of all possible states in which the TFs can bind on the DNA. The fractional occupancy $f_{i,i+m;\,a}$ denotes the average occupancy of the site at equilibrium. Its calculation requires consideration of interactions between sites. Two overlapping sites cannot be occupied at the same time as illustrated in Figure 4.3a, and in some cases a TF bound at one site increases the binding affinity at a nearby site by a factor of w_{ij}^{coop}. In the present application, cooperativity only

occurs between pairs of bound Bcd less than 60 bp apart as shown in Figure 4.3a ([154, 32]). The resulting mathematical structure is, in essence, a Markov random field with known conditional probabilities $K_{i,i+m;\,a}$ at each position $(i, i+m)$ for each TF a. Using this interpretation, $f_{i,i+m;\,a}$ denotes the marginal probability of finding the specific protein a at the site $(i, i+m)$ at a given instant. The calculation of $f_{i,i+m;\,a}$ is best performed by computing the partition function Z, which can be calculated by a fast, recently discovered algorithm ([14, Appendix S1]). The algorithm is essentially a form of Belief propagation ([122]) and is given in Algorithm 3, which is also thus designated in Figure 4.4. The fractional

Algorithm 3 The Algorithm for Fractional Occupancy

Initialize $Z_N^- = 1$ and $Z_0^+ = 1$
for $i \leftarrow 1$ *to* N **do**
  $q_{i,a} = [TF]_a K_{i,a}$
  for $a \in \{$*Transcription Factors*$\}$ **do**
    $Z_{i,a}^{+nc} = q_{i,a} Z_{i-m-1}^{+}$
    $Z_{N-i,a}^{-NZ} = q_{N-i,a} Z_{N-i+m+1}^{-}$
    $Z_{i,a}^{+c} = \sum_{j=m+1}^{c_d} w_{ij}^{\text{coop}} q_{i,a} q_{i-j,a} Z_{i-j-m-1}^{+}$
    $Z_{i,a}^{-c} = \sum_{j=m+1}^{c_d} w_{ij}^{\text{coop}} q_{N-i,a} q_{N-i+j,a} Z_{N-i+j+m+1}^{-}$

    $Z_i^+ = \sum_a Z_{i,a}^{+NT} + Z_{i,a}^{+c}$
    $Z_{N-i}^- = \sum_a Z_{i,a}^{-nc} + Z_{i,a}^{-c}$
  end
end
return Z_a^{+NS},Z_a^{+c},Z_a^{-nc},Z_a^{-c}, Z_0^-

occupancy is then given by

$$f_{i,i+m;\,a} = \frac{Z_{i,i+m,a}^{+nc} Z_{i,i+m,a}^{-c} + Z_{i,i+ma}^{-nc} Z_{i,i+m,a}^{+c} + Z_{i,i+ma}^{-nc} Z_{i,i+m,a}^{+nc}}{Z_0^-}. \tag{4.2}$$

We discovered that Algorithm 1 can be reimplemented as an bidirectional Recurrent Neural Network (RNN) with linear activation units. At each recurrent unit, indexed here by i, $q_{i,a}$ is the new input, $Z_{i,a}^{-nc}, Z_{i,a}^{-c}, Z_{i,a}^{+nc}, Z_{i,a}^{+c}$ are outputs, and Z_i^+ and Z_{N-i}^- are the information passed to next recurrent unit. In this RNN, the only unknown is w_{ij}^{coop} which

is found by training on the data.

4.3.2 TF-TF Interactions

Interactions between bound TFs follow phenomenological rules. A central feature of the *cis*-regulatory DNA of metazoan genes is the fact that biological function is encoded in multiple binding sites ([223, 220, 221]). This fact is expressed mathematically in the phenomenological equations below. In the present application, bound TFs have specific roles derived from specific experimental results, although this approach also works if the roles are not known a priori ([18]). Here, Bcd, Cad, Dst and Dic are activators; Hb, Tll, Kni, Kr and Gt are quenchers; and both Bcd and Cad are coactivators of Hb. We now describe the actions of each class of TF in the order which we compute them. The ultimate goal of this computation is to obtain the summed action of activators after their contribution has been increased by coactivation and diminished by quenching.

Coactivators

Coactivators turn a nearby quencher into an activator as schematically illustrated in Figure 4.3b. This action is described by the equation [119, 14]

$$\hat{f}_i^{Q_C} = f_i^{Q_C} \prod_{j=i-k}^{i+k} (1 - d_c(j) E_C^{Q_C} f_j^C), \tag{4.3}$$

where $\hat{f}_i^{Q_C}$ is the portion of activator fractional occupancy created from the total fractional occupancy $f_i^{Q_C}$. $d_c(j)$ is a convolutional kernel describing the distance dependence of coactivation. $d_c(j) = 1$ if $\mid j - i \mid$ less than 156 bp. In this application, $k = 206$. As $\mid j - i \mid$ increases to 206 bp, $d_c(j)$ decreases linearly to 0. $E_C^{Q_C} \in [0, 1]$ denotes the relative strength of the coactivators. f_j^C simply refers to fractional occupancy of Bcd and Cad since they are the only coactivators in this setting.

It is easy to observe that equation (4.3) is the first term of a Taylor expansion of

$$\begin{aligned} \hat{f}_i^{Q_C} &\approx f_i^{Q_C} \prod_{j=i-k}^{i+k} (1 - E_C^{Q_C} f_j^C)^{d_c(j)} \\ \Rightarrow \hat{f}_i^{Q_C} &\approx \exp\left(\sum_{j=i-k}^{i+k} d_c(j) \log(1 - E_C^{Q_C} f_j^C) \right) f_i^{Q_C}. \end{aligned} \tag{4.4}$$

This can be turned into three convolutional linear activation units, so that

$$\begin{aligned} y_j &= \log(1 - E_C^{Q_C} f_j^C); \\ z_i &= \exp\left(\sum_{j=i-k}^{i+k} d_c(j) y_j \right); \\ \hat{f}_i^{Q_C} &= f_i^{Q_C} z_i. \end{aligned} \tag{4.5}$$

This is schematically designated by the pink box labeled "co-activators" in Figure 4.4

Quenchers

As their name suggests, quenchers suppress activation by all activators, both native and those subject to coactivation, as schematically indicated in Figure 4.3b and 4.4. Their action is local and occurs only within around 100 to 150 base pairs ([99]). The basic mathematical formulation of the effect of nearby quenchers on the activator at position i is given by ([119])

$$\hat{f}_i^A = f_i^A \prod_{j=i-k}^{i+k} (1 - d_q(j) E_A^Q f_j^Q), \tag{4.6}$$

where f_i^A is the fractional occupancy of activators whether coactivated or not, $\hat{f}_i^A$ is the fractional occupancy of activators after quenching, f_j^Q is the fractional occupancy of a quencher bound at position j. $E_A^Q \in (0, 1)$ is the strength of quencher Q on activator A and $d_q(j)$ is a convolutional kernel representing the range of quenching on the DNA strand. $k = 150$ bp, and $d_q(j) = 1$ when $\mid j - i \mid \leq 100$ and goes linearly down to 0 from 100 to 150 bp. Using a

mathematical argument similar to that of the previous section, we can write

$$\hat{f}_i^A \approx \exp\left(\sum_{j=i-k}^{i+k} d_q(j)\log(1-E_A^Q f_j^Q)\right) f_i^Q. \tag{4.7}$$

This gives three convolutional linear activation units.

$$\begin{aligned} y_j &= \log(1-E_A^Q f_j^Q); \\ z_i &= \exp\left(\sum_{j=i-k}^{i+k} d_q(j) y_j\right); \\ \hat{f}_i^Q &= f_i^A z_i. \end{aligned} \tag{4.8}$$

Activation

Last, we sum the fractional occupancies of all activators remaining after the previous two steps. We consider the activators to lower the energy barrier in a diffusion limited Arrhenius rate law ([14]), which has the mathematical form of a sigmoidal thresholding function. This results an fully connected layer, shown schematically in Figure 4.4, which yields the mRNA synthesis rate. In the experimental system used, mRNA has a lifetime much shorter than the time required to change transcription rates, so that the mRNA concentration [`mRNA`], an experimentally observable quantity, is given by

$$\begin{aligned} [\text{mRNA}] &\propto \frac{d[\text{mRNA}]}{dt} \\ &= R\left(\frac{\exp\left(\sum_{j\in\{A,Q_C\}} E_j \sum_i \hat{f}_i^j - \theta\right)}{1+\exp\left(\sum_{j\in\{A,Q_C\}} E_j \sum_i \hat{f}_i^j - \theta\right)}\right). \end{aligned} \tag{4.9}$$

Here $E_j > 0$ represents the activating strength of each activator and is obtained by training on the data. θ, also obtained by training, is the amount of activation in the absence of activator and R is the maximum synthesis rate.

4.4 Implementation, Training, and Results

We train on the target

$$[\text{mRNA}]_{\texttt{train}} = \frac{[\text{mRNA}]_{\texttt{cell}}}{\sqrt{(\sum_{\texttt{cells}}[\text{mRNA}]^2_{\texttt{cell}})}}, \tag{4.10}$$

using an L2 norm, minimizing the loss function $L = \|[\text{mRNA}]_{\texttt{train}} - [\text{mRNA}]_{\texttt{model}}\|_2^2$.

We implemented and trained this model in Keras with a TensorFlow back end ([50, 1]).The resulting architecture is shown in Figure 4.4. The figure shows 9 layers, essentially one for each class of box. The actual implementation in Keras, which performed the precise computation described above, was deemed by Keras to contain 223 layers. This large number of layers is a consequence of the fact that we did not use any of the conventional architecture([124, 107]), and hence needed to use the `Lambda` functions in Keras to represent some of the activation functions and PWM convolutions. Algorithm 3 was implemented as a special layer in Keras with two `rnn` functions. Our implementation contains 52 unknown parameters.

The training data and PWMs was as previously described ([119]), although training data was limited to the fusion constructs M32, M3_2, M23, and M2_3. The model was trained using a single `Intel Core i7-8700K` CPU. The training data contains 232 unique observations of $[\text{mRNA}]_{\texttt{train}}$. Each of these observations is associated with the DNA sequence that drives the expression together with the concentration level of TFs that is characteristic of the position of the observed nucleus in the embryo. The model was trained using Adam ([120]), for 200 epochs, with Nesterov momentum as implemented in Keras. Training took approximately 2 hours. This compares with several days of serial simulated annealing before Algorithm 3 was devised ([119]), and is about equivalent to the time taken by code using Algorithm 3 running in parallel with the loss function for each construct computed on a separate core. However, optimization by simulated annealing requires several million evaluations of the loss function. while the implementation use here required about 10,000. This is

indicative of an algorithmic speedup on the order of 100. The earlier work used about 10000 lines of compiled C++ while this work uses less than 1000 lines of Python. However, the mismatch between algorithmic and wall clock speedup indicates that there is considerable scope for improvement of the TensorFlow backend for this type of problem. In addition, This implementation is more modular that that described in [119]. The use of Keras enable us to break the model up into blocks represented by proteins and interactions, which were stacked on top of one another with minimal modification to the code. This will allow for greater flexibility when producing similar models for other organisms or for extending model to different biological functions, such as chromatin accessibility.

The results of the training are shown in Figure 4.2. It is important to note that this model tends not to over-fit the data since the number of parameters is only 52 compared to the 232 nuclei we used as the training set. Nevertheless, the RMSE, an approximate measure of the average error between the behavior of the model and a quantitative observation, is 6.89, approximately the uncertainty level of the data itself, which is accurate to about 5 to 10%. This quality of fit is less than the RMSE of 2.3 reported in [119], but this study also adjusted parameters controlling the range of co-activation and quenching, as well as the minimum PWM score to include a binding site, which was set to 0 here. Although the reduction in degrees of freedom resulted in a poorer fit, it was nevertheless within experimental error. Comparison to other studies with thermodynamic models are complicated by the diverse set genes selected for modelling. However, we note that a comparison to one prominent published study ([214]) is impossible because the model used failed to give any expression from stripe 2. Another study using comparable data ([98]) reported values of correlation coefficient R between 0.55 to 0.60, while the R value in our study is 0.91.

We tested the predictive power of our DNN by confronting it with set of enhancer sequences which the model has not previously seen. The test sequences must be capable of driving expression in *D. melanogaster* embryos, so that the same TF dataset used for training can be employed to calculate the predicted expression along the A-P axis. In Figure

(4.6), we show the predicted expression of four enhancer sequences, each of which consists of 58 nuclei and hence 58 separate predictions. We chose these enhancers for presentation because they are exceptionally stringent tests in the sense that they involve enhancers with no DNA homology with those used in the training set, either because they are from different genes or distant species. 17 additional predictions are given in the Supplementary Material.

We comment in detail on the following predictions. When considering the accuracy of these predictions, it is important to note that the experimental data used for the comparison is in the form of non-quantitative images, which are referred to by specific figure in the citation below. Because the comparison is made to non-quantitative data, it is not possible to assess the quantitative accuracy of the prediction, but only the spatial position. *run_str3_7* is the enhancer of the *runt* gene of *D. melanogaster* that drives *runt* stripes 3 and 7, each of which is about 2 % E.L. anterior of the corresponding eve stripes. The positions of the stripes are accurate [121, Figure 3K and 3D]. *eve_S2E(cyn)* and *eve_S2E(spp)* are respectively the *eve* stripe 2 enhancers from *Sepsis cynipsea* and a different but unidentified *Sepsis* species. *eve_s37E(pun)* is the stripe 3/7 enhancer from *Sepsis punctum*. The *Sepsis* genus is a member of the family Sepsidae, not Drosophilidae, and are so distant from *D. melanogaster* that there is no sequence homology with the *D. melanogaster* enhancer longer than a pair of overlapping binding sites ([97, 96]). Nevertheless, all *Sepsis* enhancers that have been assayed in *D. melanogaster* embryos drive expression of stripe 2 1-2% E. L. posterior to their native position and stripes 3 and 7 at the native position ([97, Table 2]). For *eve_S2E(cyn)*, the predicted stripe 2 is of the correct width and 2 % E.L. more posterior than reported. In Sepsidae, stripe 2 enhancer also drives expression of stripe 7. For *eve_S2E(cyn)*, our prediction of stripe 7 expression is completely accurate. For *eve_S2E(spp)*, the predicted stripes are in the correct location, but stripe 2 is significantly wider than observed. For *eve_S37E(pun)*, our prediction for stripe 3 is about 2 nuclei posterior and twice as wide as observed, and for stripe 7, our prediction is completely correct. These predictions demonstrate the generalization capabilities of the DNN implementation of the model. The quality of prediction is comparable

to but slightly poorer than the original implementation [119, Figures 4 and S5].

4.5 Discussion

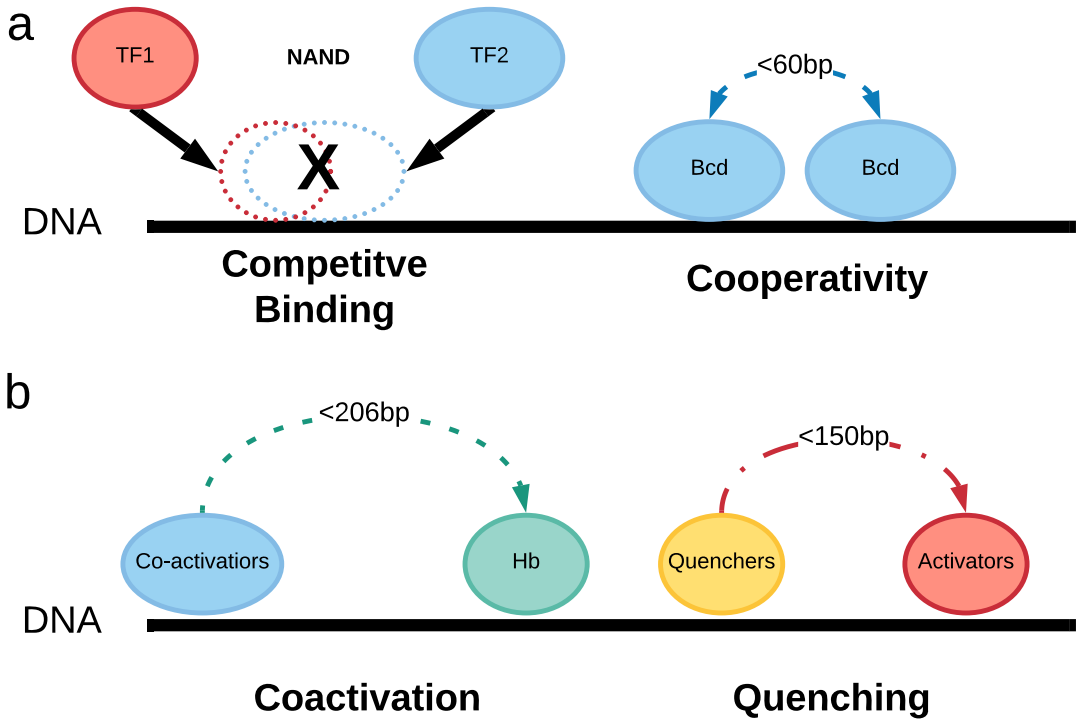
The work presented here constitutes a strong proof-of-concept for the proposition that DNN can be extremely useful for the construction of precise models of metazoan transcriptional control. Specialized properties of the early *Drosophila* embryo made it uniquely advantageous for the study of the fundamental properties of metazoan transcription before the onset of genomics. For that reason, it is natural that thermodynamic models of metazoan transcription were developed in *Drosophila* ([191, 109, 214, 116, 64, 98, 119, 161, 209, 211, 14, 13]). As explained in the introduction, we selected what is arguably the most accurate and predictive enhancer level thermodynamic model for reimplementation as a DNN. We have shown that a simpler reimplementation of this model performance almost as well as the original([119]). The reimplementation is much simpler to code than the original model, amounting to about 600 lines of Keras compared to about 10,000 lines of C++. Implementation in Keras also provides a numerical advantage by permitting the use of back propagation (BP) and stochastic gradient descent for optimization without the need to hand code partial derivatives. As reported above, the use of SGD provides an algorithmic speedup of about 100. Although some issues connected to the wall clock speed of the Keras implementation remain, these results suggest that models of this type could be scaled up to much larger datasets. A possible limitation of such generalization is the extensive prior knowledge of details of *Drosophila* transcription that were used in this study.

There are ample reasons for believing that this lack of prior knowledge can compensated for by increased quantity and quality of computation. In the absence of prior knowledge of the functional roles of TFs, a model very similar to the one we reimplemented here applied to erythropoesis in mouse ([18]) considered all possible combination of TF functional roles, and selected the combination which minimized the loss function. The resulting model, while computationally expensive, proved quite predictive ([193]). This shows that extensive prior

knowledge is not required.

Until recently, a more serious limitation was the absence of datasets that contained not only information about the sequence but also the concentration of TFs and levels of reporter expression. Such datasets, on a genomic scale, have begun to be available for flies and mammals, including humans ([6, 151, 222, 181, 236]). These genomics datasets are much larger than that used in the mouse study ([18]). However, the factor of 100 algorithmic speedup, if converted to wall clock speedup, provides a way forward. The introduction of reinforcement learning techniques may provide further computational efficiency in determining the functional roles of TFs.

With respect to Deep Learning, our model constitutes an example of a fully interpretable DNN that is not merely biologically plausible but biologically validated ([119]). It is our hope that this example will provide insights into the interpretabilities of DNNs in general, a problem that has received wide attention in the community ([81, 38, 22, 137, 155, 247]).



c

Regulatory Mechanism	Competitive Binding	Cooperativity	Coactivation	Quenching
Range of Action	Overlapping Binding Sites	<60 bp	<206 bp	<150 bp
Participating TFs	All TFs	Bcd on Bcd	Bcd or Cad on Hb	All quenchers (Hb, Kr, Gt, Kni, and Tll) on any activator (Bcd, Cad, Dst Dic, and activated Hb)

Figure 4.3: (a): Diagram of chemical interactions of the transcription factors on the DNA strain that are considered in Algorithm 3, namely competitive binding (all TFs) and cooperativity (Bcd only). (b): Diagram of phenomenological interactions between bound TFs. Coactivators (Bcd and Cad only) act on quenchers (Hb only in this application) and turn them into activators. Quenchers (Hb, Kr, Kni, Gt ,and Tll) act on activators (Bcd, Cad, Dst, Dic and activated Hb) to quench their activating power. (c): Table summarizing regulatory interactions among the TFs in the DNN.

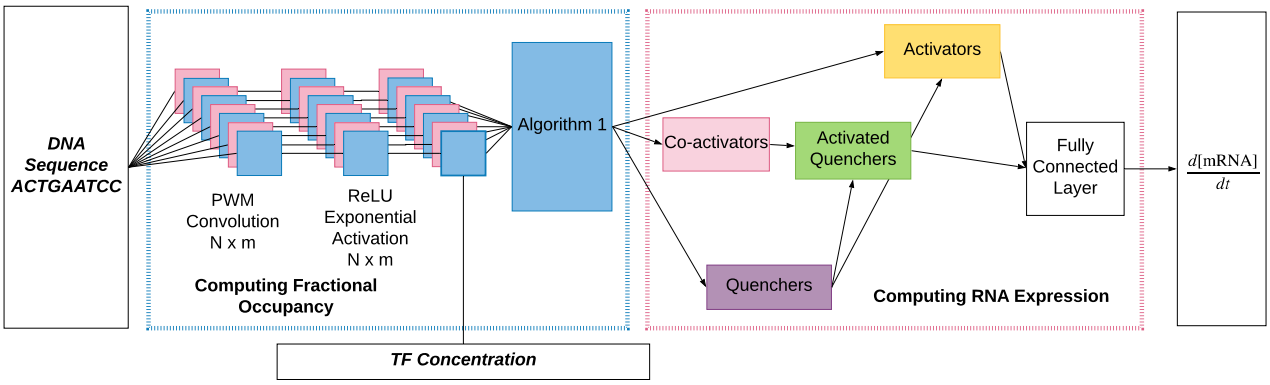


Figure 4.4: This is a graphical representation of the DNN. (Left) Chemical calculations. The computation can be seen as the DNA going through a convolution and then passing through a `ReLU` and then the the $(\exp(\cdot))$ activation function shown in equation (4.1). (Right) The graphical representation of interactions between bound TFs. Coactivators activate quenchers, quenchers quench activation, and activators combined together in a fully connected layer produce mRNA.

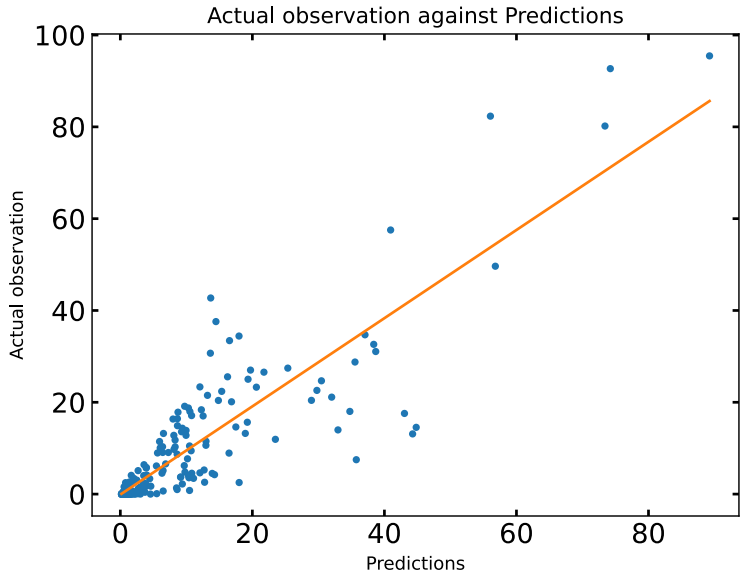


Figure 4.5: The figure shows [`mRNA`] from the four constructs in the training data against the prediction of the model. The correlation coefficient R is 0.91.

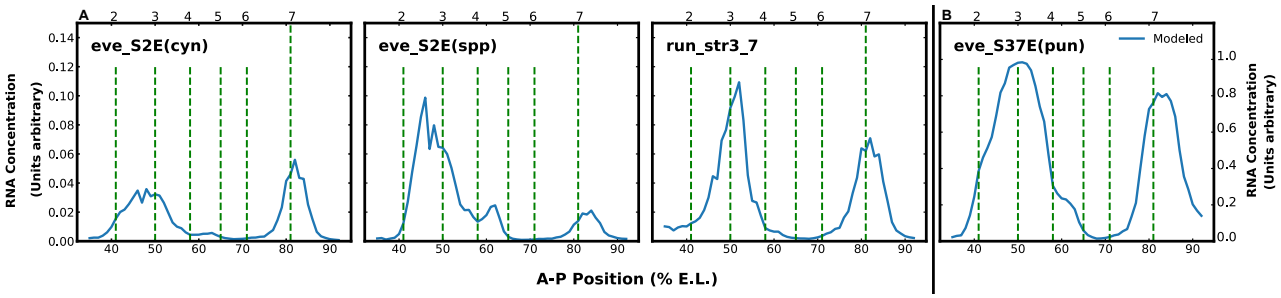


Figure 4.6: The figure shows four examples of predictions driven by enhancers not used for training. The location of eve stripes 2 through 7 are shown by vertical dashed lines. The vertical axis shows predicted mRNA concentration; note that the scale for the three graphs to the left of the vertical line differs from the graph on its right. The horizontal axis shows A-P position in % E.L. The enhancers shown are described in the text.

CHAPTER 5

APPLICATION OF NON-PARAMETRIC BAYESIAN INFERENCE TO TRANSCRIPTIONAL BURSTING

5.1 Abstract

Transcriptional bursting refers to the discovery that transcription occurs in random bursts. One can model this process with a Markov model with multiple states. The model describes the waiting time between RNA polymerase binding events as a mixture of exponential distributions where both the number of exponentials in the mixture, their relative size and their means are all unknown. In this chapter we introduce a Bayesian non-parametric method for data inference. This algorithm gives us a set of posterior samples that we can use to compute the mean and variance of the desired parameters. In addition, we are able to prove that as the number of observation goes to infinity, this method will converge to the correct answer.

5.2 Introduction

Quantitative data describing the intensity of the observed burst of transcription [228] as a function of time can be deconvolved into a distribution of waiting times between successive transcription initiation events, at each of which an RNA polymerase molecule binds and begins to elongate a transcript. It has been shown that the distribution of these waiting times $T \in \mathbb{R}^+$ can be modelled by a mixture of exponential distributions with a density $p(t)$ given as

$$p(t) = \sum_{j=1}^{N} A_j \lambda_j \exp\left(-\lambda_j t\right) \tag{5.1}$$

where parameters A_j, λ_j, and N are unknown but subject to the condition that $\sum_{j=1}^{N} A_j = 1,\ N > 0, A_j > 0, \lambda_j > 0$ [228]. Here, we denote this probability measure

of the random variable T as P. Using the probability measure P, there is a corresponding survival function

$$S(t) = 1 - P([0,t]) = \sum_{j=1}^{N} A_j \exp\left(-\lambda_j t\right)$$

induced by the parameters $(N, \mathbf{A}, \boldsymbol{\lambda})$. The goal of this chapter is to present a method to find the probability measure P and the associated $S(t)$ using a Bayesian approach.

5.2.1 Notation

In this chapter, we present a Bayesian inference approach to P through find the posterior distribution. Our starting point is a set of waiting times between successive binding event of polymerase t_i, $i \in \{1, \cdots, n\}$. Each t_i is independently sampled from a true underlying model of a mixture distribution with a density function given as

$$p_0(t) = \sum_{j=1}^{N_0} \lambda_{j0} A_{j0} \exp\left(-\lambda_{j0} t\right). \tag{5.2}$$

In our analysis, we assume that N_0, A_{j0}, λ_{j0} are unknown. This true underlying probability measure is denoted as P_0

5.2.2 Motivation and Approach

As number of states N is unknown, a maximum likelihood estimation of the parameters with no restriction breaks down because increasing the number of states N increases the likelihood, giving rise to unphysically large value of N. One approach is to constraint N artificially, but such constraints do not provide any mathematical and physical rigor. Tantale at el [228] proposed setting N to be the minimum value such that the best-fit survival function $S_{\mathrm{MLE}}(t)$ falls within 95% confidence interval of a non-parametric estimation of $S(t)$ using the Kaplan-Meier estimator. Using Bayesian analysis, one can place a prior on N to constrain it in a very natural way. This scenario is a type of non-parametric Bayesian inference. One

key advantage is that we are able to prove that the posterior will converge to the correct distribution when the number of observations n goes to infinity.

Non-parametric Bayesian inference deals with regimes where the dimension can be as large as needed, but finite with probability 1. In particular, non-parametric Bayesian methods have been successful in clustering [175, 55], fitting to densities [141, 61] and non-parametric regression problems [57]. We select as the prior a Dirichlet Process with a central gamma distribution. We use this stochastic process as a prior over λ_j. While there is no closed form formulation of its density, there exists an analogous process which can be used to sample from this prior called a stick breaking process [216]. The prior is constructed as follows.

Construction of the prior:

1. Start with the "stick" $(0, 1]$
2. Set $c_0 = 0$
3. For $j = 1, 2, \cdots$
   (a) Sample $\alpha_j \sim \texttt{Beta}(1, \alpha)$
   (b) Define $c_{j+1} = c_j + \alpha_j(1 - c_j)$
   (c) Sample $\lambda_j \sim \text{Gamma}(a, b)$
4. Denote the distribution as

$$\sum_{j=1}^{\infty} \lambda_i (c_{j+1} - c_j) \exp(-\lambda_i t)$$

Note the following. First, the number of λ_j in the prior is infinite. Second, the weights that are given to each λ_j are distributed according to an infinite extension of the Dirichlet distribution. α can be seen as a sparsity parameter denoting the rate of which the weights decay: the smaller the α, the faster the weights decay. Thus, α controls the prior for N. a

and b, the gamma distribution parameters, are the priors for λ_j. a and b do not affect overall consistency. However, in practice, the choice of a and b do affect the convergence rate.

To perform inference, one samples from posterior distributions parameterized by $\mathbf{A}$ and λ using the Monte Carlo algorithm. We shall prove that the stochastic process in the posterior distribution converges to P_0 as the number of samples reach infinity. This gives us a theoretical guarantee that our approach will lead us to the true solutions. For actual implementation, we introduce a class of Monte Carlo algorithm called Block Gibbs sampling and we demonstrate its effectiveness through simulations studies and experiments on real data .

The outline of Chapter is as follows: In section 5.3, we will show the consistency of the method and prove that this method converge to the correct distribution. In section 5.4, we describe the sampling algorithm for the posterior distribution. We perform numerical experimentation of our method in section 5.5 and, we will conclude with a detailed discussion in section 5.7.

5.3 Theoretical consistency

In Bayesian inference, it is necessary to prove that the posterior distribution will converge to P_0 in probability. The posterior samples are sets of tuples $(N, \mathbf{A}, \boldsymbol{\lambda})$ which induce a distribution $P \mid t_1 \cdots t_n$ by equation 5.1. In this analysis, we show convergence in distribution and not parameter value. This is because N can be arbitrarily large. As such, we note that the parameters in P are not strictly identifiable, because (e.g.) for any (λ_j, A_j),

$$A_j \lambda_j \exp(-\lambda_j t) = 0.5 \times A_j \lambda_j \exp(-\lambda_j t) + 0.5 \times A_j \lambda_j \exp(-\lambda_j t).$$

Here, the two set of parameters are equivalent. i.e. $\{(\lambda_j, A_j)\} \equiv \{(\lambda_j, 0.5A_j), (\lambda_j, 0.5A_j)\}$. This implies that the parameters in P are not strictly identifiable unless one restricts N and provides conditions on λ_j in the modeling process after which it is identifiable [230]. A

theoretical analysis on how each parameter $(N, \mathbf{A}, \boldsymbol{\lambda})$ concentrates to the true value is not well defined although in practice, one can place a posterior weight on each value of N and select the most probable N. Therefore, in this analysis, we compare P_0 to the distribution $P \mid t_1, \cdots t_n$ induced by the posterior.

To set up this problem rigorously, let $\mathcal{P}$ be the space of mixtures of exponential distributions, $\mathcal{F}_{\mathcal{P}}$ the σ-algebra generated by $\mathcal{P}$, and Π probability measure assigned on this σ-algebra. We define the tuple $(\mathcal{P}, \mathcal{F}_{\mathcal{P}}, \Pi)$ as the probability space on the set of distributions. Let TV be the total variation distance $TV(P_0, P) = \sup_{A \in \mathcal{F}_{\mathbb{R}}} \mid P_0(A) - P(A) \mid$ where $\mathcal{F}_{\mathbb{R}}$ denotes the Lebesgue σ-algebra.

Definition 5.3.1. *(Definition 6.1 [87]) Let U_ϵ be the TV neighborhood of distance ϵ centered at P_0 ($U_\epsilon^c = \mathcal{P} \backslash U_\epsilon$). The posterior distribution is said to be weakly consistent with respect to TV at P_0 if and only if, for any $\epsilon > 0$,*

$$\Pi(U_\epsilon^c \mid t_1, t_2 \cdots t_n) \to 0$$

in P_0-probability as $n \to \infty$. The posterior distribution is said to be strongly consistent at P_0 if this convergence is in the almost-sure sense.

Here, we will prove strong consistency. This implies that all probability weight will be placed on top of a set of correct distributions as the number of data points go to infinity. Specifically, we prove:

Theorem 5.3.1. *Let Gamma(a, b) be a gamma distribution parameterized by the shape parameter a and rate parameter b. The posterior is strongly consistent with respect to a Dirichlet Process Prior with a central distribution Gamma(a, b) if $a \geq 2$ and $b \geq 2$.*

The full proof is given in Appendix section C.1. The basic idea of the proof is the following. There is theorem which states that strong consistency is achieved for any mixture distribution if 4 conditions hold [87, Theorem 7.15]. The 4 conditions can be summarized as

follows: For any large integer m: First, the density of the mixture components is Lipschitz with respect to λ and there is a Lipschitz constant a_m. Second, there is a bounded parameter space $\lambda \in \Lambda_m \subset \mathbb{R}^+$. Third, the size of Λ_m is well controlled in relation to a_m. Fourth, the prior concentrates on Λ_m as m goes to infinity. In the proof, we show that the 4 conditions apply to our probability model P and using the result of [87, Theorem 7.15], there is strong consistency.

5.4 Gibbs Sampling Procedure

5.4.1 Algorithm

In Bayesian methods, one samples from the posterior distribution through the use of Markov Chain Monte Carlo (MCMC). Gibbs Sampling is a variation of MCMC, where at each step, each variable is sampled conditioned on fixing all other variables. As a result, the acceptance probability for each step is always 1. A key advantage of a Dirichlet Process prior is that all conditional distributions have closed form and therefore can be sampled directly using fast computation tools [69]. In fact, to specify the algorithm, it is sufficient for us to specify the exact conditional distributions of variables A_j, λ_j and N. In Algorithm 5.4.1, we define $v_i \in \{1, \cdots N\}$ as a random variable that assigns sample t_i to the cluster centered at λ_{v_i}. For example, if $v_1 = 2$, the data point t_1 has been assigned to the cluster centered at λ_2. The notation v suggests the idea of a 'vote'. In fact, Algorithm 5.4.1 is a specific implementation of the Blocked Gibbs Sampling algorithm by Gelman [82] which we adapt for transcription bursting.

Another perspective is to consider Algorithm 5.4.1 as a probabilistic implementation of the K-means clustering algorithm. Instead of a fixed number of centers, as in the case of the K-means clustering algorithm, the Dirichlet Process places a prior on the number of centers N. In fact, if one fixes the number of centers and replaces the sampling in step 2 by picking the cluster with the highest probability and further replaces the sampling in step 5 by setting

the center λ_j to be the mean of the conditional distribution, we are essentially left with a K-means clustering algorithm. The centers are added sequentially. At each iteration, the algorithm first proposes a λ_{new}. Then each data point t_i will be assigned to a center (λ_j) by sampling v_i from multinomial distribution. If a center has at least one data point being assigned to it, then it will be accepted. Centers with no assignments are eliminated. The remaining centers (λ_j) are updated by the t_i's that are assigned for the center. At the end, N is taken as the number of the centers that remains and we update $\mathbf{A}$ by sampling from a Dirichlet distribution based on the number of data points that has been assigned to each center.

Algorithm 5.4.1:

Initialize with the set $\lambda = \{\lambda_1\}$, $\mathbf{A} = \{1\}$, and $N = 1$. Repeat the following.

1. Sample $\lambda_{\text{new}} \sim \text{Gamma}(a, b)$.

2. For each data sample i, assign the data point t_i to cluster

$$v_i \mid A_j, \lambda_j \sim \text{multinomial}\left(\frac{A_1\lambda_1 \exp(-\lambda_1 t_i)}{\sum_i A_j\lambda_j \exp(-\lambda_j t_i)}, \cdots, \frac{A_{\text{new}}\lambda_{\text{new}} \exp(-\lambda_{\text{New}} t_i)}{\sum_j A_j\lambda_j \exp(-\lambda_j t_i)}\right).$$

3. $m_j = \sum_i \mathbb{I}(v_i = j)$, the number of data points assigned to cluster j.

4. $N = \sum_j \mathbb{I}(m_j > 0)$, the number of clusters with at least one data point assigned.

5. Sample $\lambda_j \mid v_i, t_i \sim \text{Gamma}\left(a + m_j, b + \sum_{v_i=j} t_i\right)$.

6. Sample $A_1, A_2, \cdots A_N, A_{\text{new}} \sim \text{Dirichlet}\left(m_1 + 1, m_2 + 1 \cdots, \alpha\right)$.

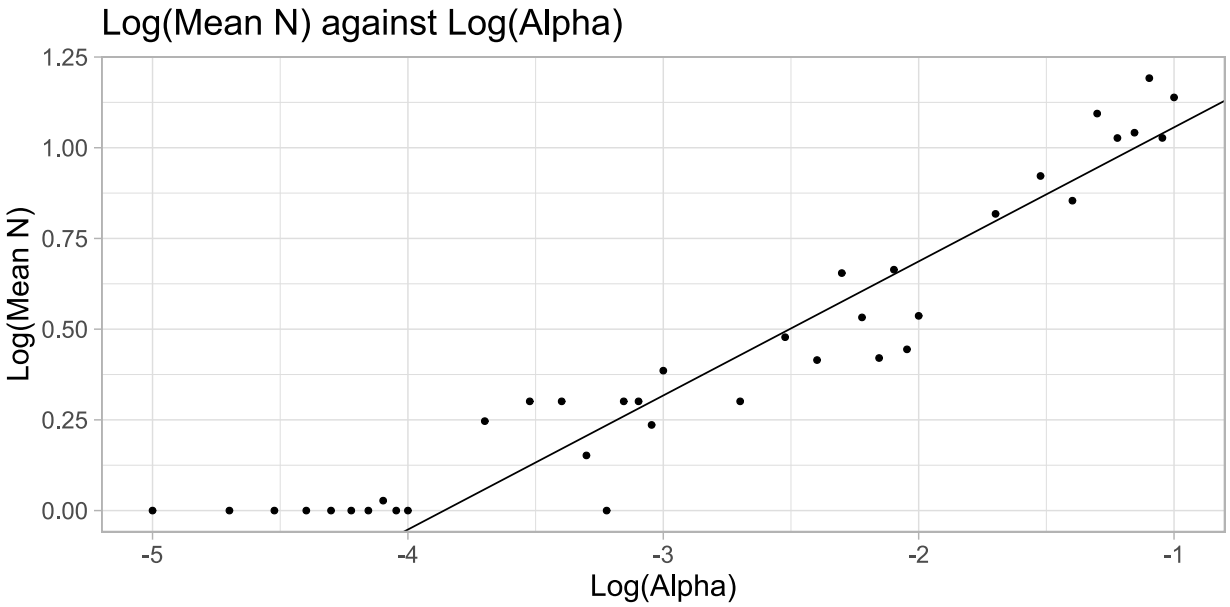


Figure 5.1: A plot of $\log(\bar{N})$ against $\log(\alpha)$ in a Monte Carlo run. We have 2000 MCMC iterations and we take the first 1000 iterations as burn-ins. Here we observe that when $\log(\alpha)$ is small, N is 1 ($\log(N) = 0$). When $\log(\alpha)$ increases past threshold of about 1×10^{-4}, N increases. The relationship between $\log(\alpha)$ and $\log(N)$ is close linear after $\log(\alpha)$ hits the threshold.

5.4.2 Block Gibbs sampling algorithm in action

We will demonstrate this method using the following simulated dataset. The dataset consists of 20000 data points t_i coming from the mixture distribution $P_0 = 0.5\exp(-0.1t) + 0.5\exp(-t)$ (i.e. $\boldsymbol{\lambda} \in (0.1, 1)$ and $\mathbf{A} \in (0.5, 0.5)$). Using this dataset, we will elucidate the possible effects of α, a, and b on the outcome of the algorithm.

Choice of α

In Algorithm 5.4.1, α determines the value of A_{new} because the mean of A_{new} is given by $\alpha/(n+N)$. Thus, the smaller the α, the less likely it is that a t_i is going to be assigned to the cluster centered at λ_{new}. In other words, α is a sparsity parameter which controls overfitting in terms of N.

We demonstrate this effect of α with the following numerical experiment. Algorithm 5.4.1 was performed using different values of α. The prior of λ_j is set to Gamma$(0.1, 1)$.

$\log_{10}(\alpha)$	-5	-4	-3	-2	-1
N_{mode}	1	1	2	3	14
$\hat{\mathbb{P}}(N_{\text{mode}})$	1.00	1.00	0.57	0.56	0.22

Table 5.1: A table of $\log_{10}(\alpha)$, Mode N_{mode}, the probability $\hat{\mathbb{P}}(N_{\text{mode}})$ estimated by taking the number of samples with the given value of N divided by the total.

The algorithm is run for 2000 iterations and we remove the first 1000 samples as burn-in. We summarize some of our results in Figure 5.1, Figure 5.2, Table 5.1, and Table 5.2.

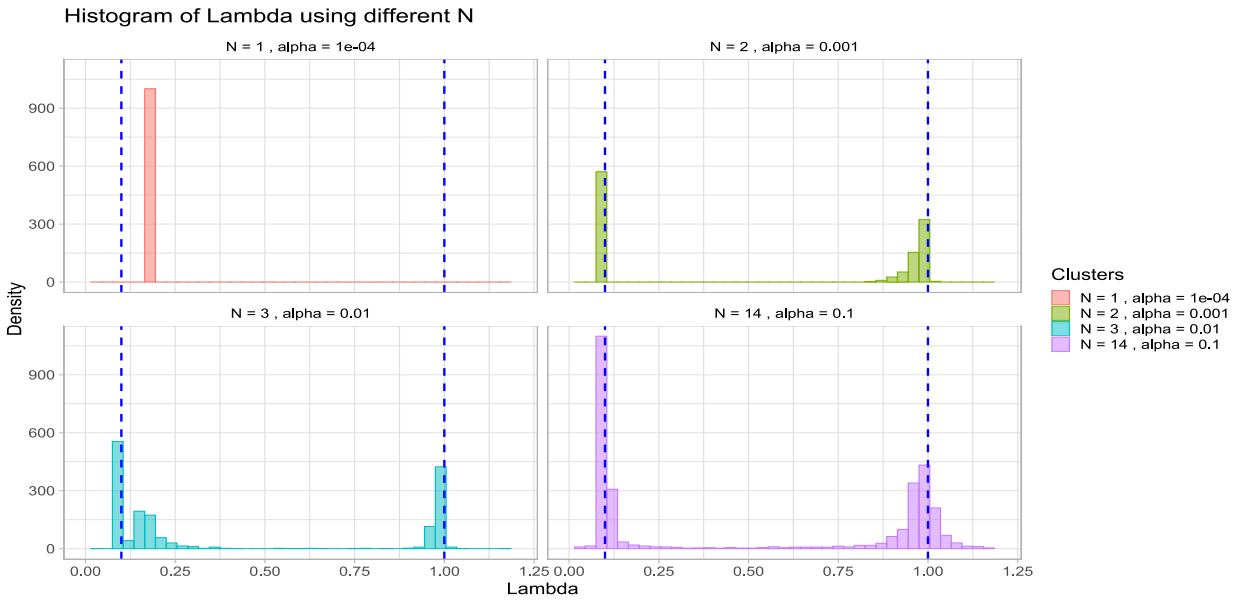


Figure 5.2: The histogram of posterior $\log(\lambda)$ samples for different N and α. The underlying distribution is $P_0 = 0.5\exp(-0.1t) + 0.5\exp(-t)$. The correct values of λ's are shown in dash lines. When α is small ($\alpha = 10^{-4}$), there is one cluster of λ. When α increases, N increases. However, one can observe that there are clusters of λ. In addition, λ samples are concentrated on 0.1 and 1.

In Figure 5.1, we plotted the log of mean posterior N against $\log(\alpha)$. When α is very small, one observes that posterior $N = 1$ with probability 1. In this Figure, there appears to be a critical point at $\log(\alpha) = -4$ where from this value onward, the posterior mean N ($\bar{N}$) increases with increasing α. In fact, there is evidently a linear relationship between $\log(\bar{N})$ and $\log(\alpha)$. Indeed, this shows that that α can be interpreted a sparsity parameter which control N.

$\hat{\mathbb{E}}(\lambda_j)$	1.01	0.96	0.20	0.11	0.00
$\hat{\sigma}(\lambda_j)$	0.18	0.05	0.11	0.07	0.03

Table 5.2: A table of $\hat{\mathbb{E}}(\lambda_j)$, the posterior mean of λ_j and $\hat{\sigma}(\lambda_j)$, the posterior standard deviation of λ_j. In this experiment, $\alpha = 0.05$

Table 5.1 demonstrates the effect of α on N. The table shows $\log(\alpha)$, the posterior mode N_{mode}, and its estimated probability $\hat{\mathbb{P}}(N_{\text{mode}})$. Again, we notice that as α increases, N_{mode} increases but $\hat{\mathbb{P}}(N_{\text{mode}})$ decreases. Increasing α also disperses the probability weights of N in the posterior.

It is evident that if $\alpha \geq 0.01$, N_{mode} is higher than the underlying N_0. At the same time, one can observe that λ_i estimates start to overfit. We show this in Table 5.2 for the case where $\alpha = 0.05$. Here, we can observe that the intervals $\hat{\mathbb{E}}(\lambda_j) \pm \hat{\sigma}(\lambda_j)$ overlap each other. This is especially visible for the case $\hat{\mathbb{E}}(\lambda_j) = 1.01$ and $\hat{\mathbb{E}}(\lambda_j) = 0.96$. λ_j estimates that are so close to each other strongly indicates overfitting, in the sense of empirically degenerate values of λ_j. In this case, using a smaller α is recommended.

This degeneracy is further illustrated in Figure 5.2. Despite the fact that N_{mode} is much larger than $N_0 = 2$, the posterior λ_j form two clusters once α is sufficiently large. Evidently, the clusters of λ_j samples concentrate around the true values of 0.1 and 1. No new clusters of λ_j appear even though the estimate of N is greater than 2. This tells us that as long as there is no overfitting, the Gibbs sampling algorithm is able to return the right set of λ_j's.

Using Algorithm 5.4.1, one collects posterior samples as outputs. As a key advantage, one can ask any questions about the posterior distribution through directly estimating the statistics. In fact, using the posterior samples, one can estimate the most probable value N, the mean of λ_j's, and their standard deviations conditional on each N. These estimates gives us single value estimators for the tuple $(N, \boldsymbol{\lambda}, \mathbf{A})$ and their errors.

5.4.3 Workflow for the application of the method

A practical implementation of Algorithm 5.4.1 requires the user to choose an α. Since there is no suggestion of a good α from theoretical analysis, we propose a scan through various α and find the largest α that does not result in overfitting. To deal with a large number of datasets to process at one time, it is not feasible for the user to manually scan through the distribution of posterior λ's for various levels of α (as been done in Figure 5.2). One needs to consider a criterion for overfitting that can be identified through a numerical comparison. Here, we consider overfitting to occur when

$$\hat{\mathbb{E}}(\lambda_k) \in (\hat{\mathbb{E}}(\lambda_j) - s\hat{\sigma}(\lambda_j), \hat{\mathbb{E}}(\lambda_j) + s\hat{\sigma}(\lambda_j)) \text{ for } k \neq j,$$

where s is sensitivity parameter. A small s means that we accept a much denser spread of λ_j. we found that $s = 2$ gives good performance. We choose the largest α that such that we observe no sign of overfitting.

This gives a practical workflow to be implemented for automated data analysis as described in Algorithm 5.4.3. One starts off with a small α and increases α incrementally until overfitting is observed. For each run (associated with a single α value), one obtain a set of posterior samples $(N, \boldsymbol{\lambda}, \mathbf{A})$. A good choice for the estimator of N is the posterior mode N_{mode}. Conditioning on the posterior mode, one then collect the set of $\boldsymbol{\lambda}$ and $\mathbf{A}$ to compute the posterior mean $\hat{\mathbb{E}}(\lambda_j)$ and posterior standard deviation $\hat{\sigma}(\lambda_j)$. With these estimates, we can judge if there is overfitting in the λ_j's and we stop the algorithm once overfitting has occurred.

Algorithm 5.4.3:

For each α:

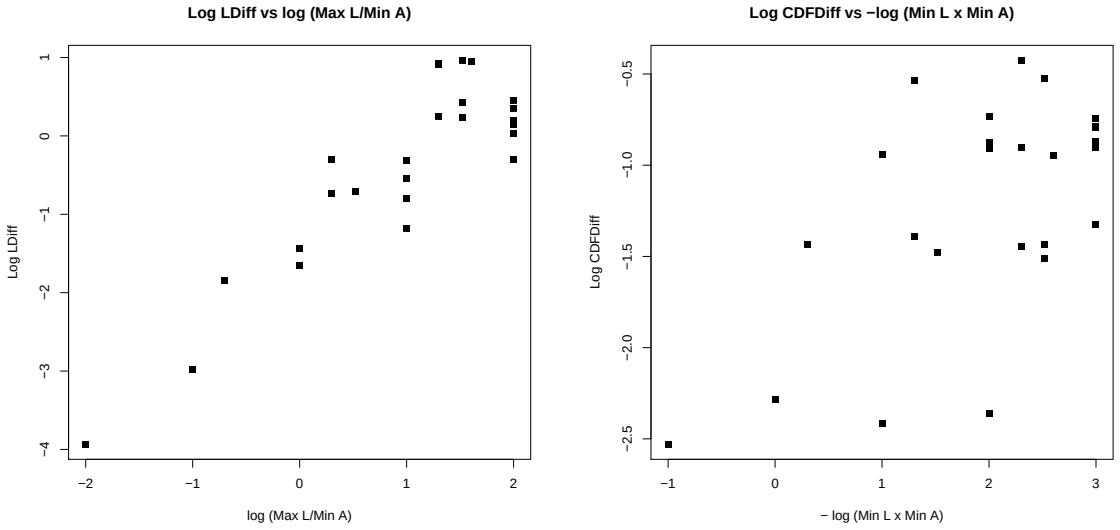
1. Run Algorithm 5.4.1 with burn-in.
2. Collect samples of N, $\mathbf{A}$, and λ.
3. Compute $\hat{N}$, $\hat{\mathbb{E}}(A_j)$, and $\hat{\mathbb{E}}(\lambda_j)$ by taking the mode N_{mode}, posterior mean of A_j and λ_j samples conditional on N_{mode}.
4. Compute the standard deviation $\hat{\sigma}(\lambda_j)$.
5. Check for overfitting using the criterion

$$\hat{\mathbb{E}}(\lambda_k) \in (\hat{\mathbb{E}}(\lambda_j) - s\hat{\sigma}(\lambda_j), \hat{\mathbb{E}}(\lambda_j) + s\hat{\sigma}(\lambda_j)) \text{ for } k \neq j.$$

5.5 Experimentation and simulation

To study performance of the workflow, we conducted a comprehensive simulation experiment to study the effects of $N_0 = 1, 2, 3$, and 4 with a variety of values of $\mathbf{A}_0$ and $\boldsymbol{\lambda}_0$ given in Table 5.3.

For each of these scenarios, we build 50 simulation data-sets, each consisting of 20000 data points t_i. We consider balanced and skewed spread of $\mathbf{A}_0$ and λ_0 where component λ_js can differs by orders of 10, 100, and 1000. The Gibbs sampling algorithm is implemented in `R`-code that is packaged in a `docker` container[1].



(a) (b)

Figure 5.3: Errors in different set-ups. In Figure 5.3(a), we plot $\log_{10}(\mid \lambda - \hat{\mathbb{E}}(\lambda) \mid)$ against $\log_{10}(\max \lambda_i / \min A_i)$. In Figure 5.3(b), we plot $\log_{10}(\max_t \mid \hat{F} - F \mid)$ against $\log_{10}(\max \lambda_i \times \min A_i)$.

In this simulation study, we use the following set-up for our MCMC run. The number of MCMC iterations is 5000 with 1000 discarded as burn-in. α was scanned through monotonically increasing values ranging from 1×10^{-5} to 1 such that the values of $\alpha = x \times 10^y$ where x was stepped through x values of 1, 3, 5, and 7 and y takes on values from –5 to –1. The outcome of the experiments is recorded in Table 5.3.

1. The code can be found in `https://github.com/yiliu9090/GibbsSamplingForInferenceR`

N	$\mathbf{A}$				$\boldsymbol{\lambda}$				NTF	NDiff	std	ADiff	std	LDiff	std	CDF Diff	std
1	1				0.01				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	1				0.1				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	1				1				0.08	0.08	0.27	0.02	0.06	0.04	0.10	0.01	0.01
1	1				10				0.08	0.08	0.27	0.00	0.00	0.29	0.72	0.00	0.00
2	0.5		0.5		0.01		0.1		0.00	0.00	0.00	0.05	0.01	0.01	0.00	0.04	0.01
2	0.1		0.9		0.01		0.1		0.00	0.00	0.00	0.72	0.03	0.02	0.00	0.14	0.01
2	0.5		0.5		0.01		1		0.00	0.00	0.00	0.01	0.01	0.50	0.09	0.13	0.04
2	0.1		0.9		0.01		1		0.10	0.10	0.30	0.39	0.11	0.07	0.06	0.16	0.02
2	0.5		0.5		0.01		10		0.08	0.08	0.27	0.15	0.26	8.48	2.44	0.37	0.16
2	0.1		0.9		0.01		10		0.00	0.00	0.00	0.08	0.01	0.50	0.15	0.05	0.01
2	0.5		0.5		0.1		1		0.12	0.12	0.33	0.10	0.19	0.18	0.07	0.04	0.01
2	0.1		0.9		0.1		1		1.00	1.00	0.00	0.81	0.06	0.48	0.14	0.12	0.01
2	0.5		0.5		0.1		10		0.00	0.00	0.00	0.08	0.02	8.26	1.28	0.29	0.07
2	0.1		0.9		0.1		10		0.46	0.46	0.50	0.50	0.17	1.40	0.89	0.19	0.04
2	0.5		0.5		1		10		0.08	0.08	0.27	0.04	0.13	1.78	0.73	0.04	0.02
2	0.1		0.9		1		10		1.00	1.00	0.00	0.74	0.00	2.81	0.16	0.12	0.01
3	0.3	0.3	0.4		0.01	0.1	1		0.04	0.04	0.20	0.11	0.08	0.19	0.04	0.03	0.01
3	0.1	0.1	0.8		0.01	0.1	1		0.86	0.86	0.35	0.47	0.08	0.16	0.08	0.13	0.01
3	0.3	0.3	0.4		0.01	0.1	10		0.46	0.46	0.50	0.51	0.12	9.11	1.58	0.30	0.08
3	0.1	0.1	0.8		0.01	0.1	10		0.04	0.04	0.20	0.51	0.09	1.59	0.39	0.16	0.03
3	0.3	0.3	0.4		0.01	1	10		0.00	0.00	0.00	0.06	0.01	2.69	0.30	0.04	0.00
3	0.1	0.1	0.8		0.01	1	10		0.00	0.00	0.00	0.76	0.10	2.23	0.15	0.18	0.01
3	0.3	0.3	0.4		0.1	1	10		0.00	0.00	0.00	0.06	0.01	1.72	0.29	0.03	0.00
3	0.1	0.1	0.8		0.1	1	10		0.78	0.78	0.42	0.45	0.05	1.58	0.18	0.13	0.02
4	0.25	0.25	0.25	0.25	0.01	0.1	1	10	0.92	0.92	0.27	0.56	0.12	8.88	0.91	0.11	0.02
4	0.1	0.1	0.1	0.7	0.01	0.1	1	10	1.00	1.00	0.00	0.61	0.08	1.06	0.34	0.12	0.01

Table 5.3: Results from the simulation study. We conduct simulation for different values of N, $\mathbf{A}$, and λ. Then we record the following statistics. NTF: The proportion of fits that gives incorrect values of N. NDiff: Average difference between estimated $\hat{N}$ and true N. ADiff: Average absolute difference between the estimated $\hat{\mathbf{A}}$ and true $\mathbf{A}$. ADiff $= \mid \hat{\mathbf{A}} - \mathbf{A} \mid$. LDiff: Average absolute difference between the estimated λ and true λ. LDiff $= \mid \hat{\lambda} - \lambda \mid$. CDFDiff: Average maximum difference between the CDF functions $F(t) = \sum_{j=1}^{N} A_j \exp(-\lambda_j t)$ and $\hat{F}(t) = \sum_{j=1}^{\hat{N}} \hat{A}_j \exp(-\hat{\lambda_j} t)$ for $t \in \{0.01, 0.02 \cdots 500\}$ i.e. $\max_t \mid \hat{F}(t) - F(t) \mid$.

From Table 5.3, one notices that the general performance of the method is strong for cases where the clusters' size is balanced. As expected, the method works well on small N_0 and is able to estimate N_0 rather well even for $N_0 = 3$. For $N_0 = 4$, the method tends to underestimate N, but the survival curve itself is well-estimated because $\max_t \mid \hat{F}(t) - F(t) \mid$ is small.

In Figure 5.3(a), we study how well the method is able to recover λ_j's by looking at the errors associated with the estimates. We observe an interesting effect in which $\log(\max \lambda_i / \min A_i)$ appears to be linearly dependent on the $\log(\mid \lambda - \hat{\mathbb{E}}(\lambda) \mid)$. It appears that $\log(\max \lambda_i / \min A_i)$ is a measure of the difficulty of the problem and the larger $\log(\max \lambda_i / \min A_i)$, the larger $\log(\mid \lambda - \hat{\mathbb{E}}(\lambda) \mid)$. A similar observation is made for $\log(\max \lambda_i \times \min A_i)$ and $\log(\max_t \mid \hat{F}(t) - F(t) \mid)$ in Figure 5.3(b). In fact, $\max_t \mid \hat{F}(t) - F(t) \mid$ is an empirical estimate of Total Variation (TV) distance. Therefore, we have that $\log(\max \lambda_i \times \min A_i)$ is another measure of the difficulty of this problem.

5.6 Demonstration on Real Data

We apply this method to a Drosophila dataset [183]. The paper investigate two key Drosophila developmental promoter motifs, the TATA box (TATA) and the Initiator (INR) [183]. Using live imaging in Drosophila embryos and new computational methods, it demonstrate that bursting occurs on multiple timescales ranging from seconds to minutes [183]. The work described implemented a machine learning method that deconvolves single nucleus mRNA production from live Drosophila embryos to detect all single polymerase initiation events. The outputs are waiting times t_i.

Because the waiting times are extracted in a deconvolution that makes certain assumptions, the waiting times occur in multiples of 1/3. This does not reflect the true distribution coming from a mixture of exponential distribution. We prepossesses data by jittering the values: i.e. we added a random deviants Uniform$(-b, b)$ to each value, where b is the smallest interval between any two values in the data set. This is smoothing process where we remove points with extreme probability weight. For example, the smallest value that can be recorded is 1/3 and there is a very heavy weight placed on this value. The jittering process then smoothes the empirical distribution by removing these extreme probability weights.

Since we do not have the underlying truth, one way to visualize how good our workflow is fitting the data is to plot the predicted survival function $S_{\text{predict}}(t)$ against an empirical survival function $S_{\text{empirical}}(t)$. We plot this in Figure 5.4. We observe that the output model generally fit the survival model well.

5.7 Conclusion

We introduced a new inference method for transcriptional bursting which is theoretically verified. While this method has a strong theoretical basis, there are still interesting extensions to consider. Specifically, there are cases where it is known that λ_j are orders of magnitude apart (i.e. $\lambda_k/\lambda_j \geq 10$ for $j \neq k, \lambda_k > \lambda_j$). One can consider introducing this into the sampling algorithm to achieve this effect.

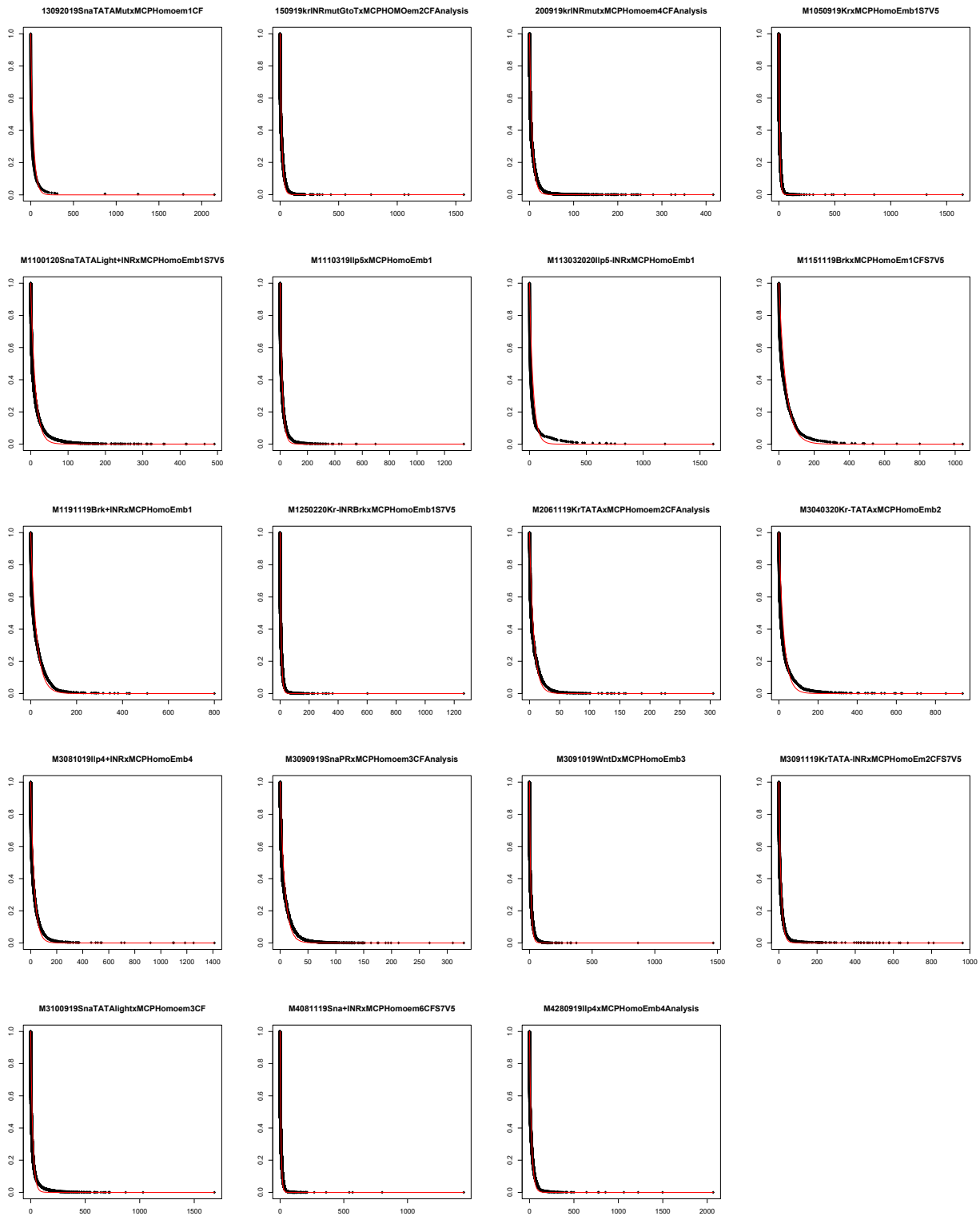


Figure 5.4: A figure of empirical survival function $S_{\text{empirical}}(t)$ against predicted survival function $S_{\text{predict}}(t)$. The black color data points are the empirical survival function and the red line is the predicted survival function using the estimated parameters $(N_{\text{mode}}, \hat{\mathbb{E}}(\lambda_j), \hat{\mathbb{E}}(A_j))$.

CHAPTER 6

DISCUSSION

6.1 Chapters Overview

The goal of this thesis is to explore issues of machine learning interpretability through Bayesian methods. Here, we considered two key methodologies of non-parametric Bayesian inference and demonstrated how Bayesian inference can be applied to key estimation problems in transcription. The output of Bayesian computation is set of posterior samples by means of which we compute key estimators. In fact, we can compute estimators for higher order moments of the posterior without the need to develop new estimation procedures.

In Chapter 2, we consider a weak form interpretability for non-parametric models: variable selection. We proved the consistency of Bayesian method for model-free variable selection and proposed an Approximate Bayesian Computation (ABC) algorithm for fast posterior sampling. The new ABC algorithm has the importance feature of being able to sample from the space of functions rather than just the real line. We expanded the uses of Bayesian methods to a larger set of non-parametric problem. However, ABC has a key computational weakness. To sample using the ABC algorithm, one has to re-sample from the prior at each iteration. There is room for efficiency improvement as one can learn from previous samples.

Following this idea, in Chapter 3, we improve the speed ABC sampling using a reinforcement learning algorithm commonly known as Thompson Sampling. Thompson Sampling does a Bayesian update at each iteration, therefore improving the speed of convergence. Under mild conditions, this algorithm converges to an optimal solution at the optimal rate. This new algorithm, named Thompson Variable Selection (TVS), allows us to conduct rapid variable selection on even in the case where the number of samples goes to infinity. In fact, we propose dividing the data into small subsets and iterate through the entire dataset subset by subset. The use of this method permits variable selection for datasets of 20000 data points. Our simulation studies shows that TVS method improve the speed variable selection

by factor of 10 compared to Bayesian Tree method without losing performance in terms of accuracy.

We considered interpretability arising from parameter estimation in Chapter 4. The specific application that we are interested in is transcriptional control. We are specifically interested in the *eve* gene in *D. melanogaster* because we able to observe its transcription process in a relatively isolated environment. Because of this, experiments are easier to do and underlying mechanism can be studied. However, even in this relatively simple case, it is difficult to model transcriptional control due to high level of complexity when proteins interacts with DNA sequences. We showed an one-to-one mapping of a thermodynamic model for transcription to a non-trivial DNN. This connection between the two models not only demonstrate a case where a DNN is fully interpretable but allows us to leverage the industrial level computational tools such as `Tensorflow` for fast gradient based optimisation methods without the need to build huge systems of software.

In Chapter 4, we did not address the phenomenon transcriptional bursting. We fill in this gap by looking at modeling transcription bursting in Chapter 5. The number of states in transcriptional bursting is unknown. Therefore, for parameter estimation and inference, we propose a non-parametric Bayesian method where we put a Dirichlet process prior on the number of states and its associated parameters. This machine learning framework has two key advantages. First, there is theoretical consistency which we have proved in Chapter 5. Second, the conditional distribution of all the parameters are in closed form and we have a very fast computation of posterior distribution using a block Gibbs sampling approach.

6.2 Revisiting Interpretability

In this thesis I have followed the work of Lipton and Fan in considering interpretability in terms of Simulatability, Decomposability, and Algorithmic Transparency [146, 65]. Here, Simulatability is the understanding over the entire model; Decomposbility is understanding the model in terms of its components; Algorithmic Transparency is to understand the training

process and dynamics of a model [146]. In this section, we will discuss how Chapter 2, 3, 4, and 5 looks into these factors.

Simulatability. We try to tackle these concepts using Bayesian Methods. One key advantage of Bayesian methods is that they can address the problem of Simulatablity as an estimation problem. This is demonstrated in Chapter 2 and 3. The Chapters tackle a weak form of Simulatablity by focusing only on finding a set of independent variables that has association with the dependent variables, i.e. variable selection. Variable selection is a very well studied idea for parametric models [233, 8] such as linear regression. Currently, in a frequentist formulation, variable selection is done through restricting the false discovery rate of the set of found variables [8, 34], for example by the method of knockoffs. However, in a Bayesian spike-and-slab formulation, one assigns a prior inclusion probability to each variable and compute the posterior inclusion probability [12, 200, 197, 205]. Rockova has extended the ideas of spike-and-slab prior to the Bayesian Forest formulations [202]. In Chapter 2, we added on Rockova's work by proving the consistency of the posterior inclusion probability for tree-based models.

Decomposability. Chapter 4 gives an interesting example of Deep Neural Network where each layer represent a thermodynamic or phenomenological formulation of known biological mechanisms. This DNN surprisingly fulfills both Simulatability and Decomposability. In general, Decomposability is not known in Deep Neural Network since the layers are heavily interconnected [65]. Past work focuses on visual interpretability of Deep Neural network [248, 249] but none has any interpretability for each parameter. As such, this is an interesting instance where a Deep Neural Network can be decomposed to its layers and one can understand the functions layers. Using this model, we can map the function of each layer to the chemical function of the proteins and DNAs. For the study of interpretability in DNNs, this provides an interesting use case that can be used for future research.

Algorithmic Transparency. This thesis focuses mainly on the issue of Algorithmic Transparency since we have introduced new algorithms in each chapter to solve key problems in the chapter. In Chapter 2, we have introduced an ABC algorithm that approximates the posterior distribution. We use this method to compute the posterior estimates of the inclusion probability. This method circumscribes the difficulty of computing likelihood and introduce a new way of looking at ABC type algorithms. ABC methods has been used for applications such as population genetics [15] but it is used only a small number of parameters. Previous work in ABC has been focus on improving its performance with respect to time [159] and applicability [2, 114, 172]. The split sample trick introduced in Chapter 2 builds new types of ABC algorithm which allows the dimension increase significantly. Interestingly, this allows us to combine gradient techniques with L0 methods in a natural way. However, how to prove that the approximation is valid is a question that is reserved for future research.

In Chapter 3, we increase the speed of ABC algorithm using reinforcement learning. The use of Thompson Sampling improves the convergence speed by taking into account outcomes from past iterations. In fact, variations of Thompson Sampling have been developed in last decade [123, 133, 243] but they have not been used for variable selection. We showed that this algorithm is a relaxation of spike-and-slab prior and its convergence rate can be proven to be optimal under weak conditions. Another interesting discovery is that this new algorithm make use of batch learning to improve speed. This is a major improvement of MCMC algorithms since computation speed is directly proportional to the number of samples n. Our new algorithm allows for much faster computation of the posterior distribution even though it is an approximation. As the task is only to select a set of variables, a faster algorithm can be useful in practise compared to an accurate computation of posterior distribution. However, if one wants to make estimation of higher order relationships, such as correlations, this method might not be appropriate.

In Chapter 4, we replace a simulated annealling algorithm with stochastic gradient descent. Stochastic gradient descent can be a much faster algorithm as it makes use of the first

order derivatives information. Having gradient information alleviate the curse of dimensionality. Indeed, one only requires hundreds of gradient computation compared to the 2 million function evaluation in simulated annealling. More importantly, we only make use of a well known SGD algorithm in Chapter 4 (ADAM) and did not explore newer versions of SGD that can be more suited for this problem and our computational hardware. With more SGD algorithms invented on a daily basis, one can expect that this optimization method can be improved in due course. In addition, we consider the speed of implementation. Whereas it takes 20000 lines of `C++` code to build the system for simulated annealling, it takes 600 lines of `python` code to implement the model in `Tensorflow`. The use `Tensorflow` will also help us utilize newly published optimization methods since most SGDs codes are written using a `Tensorflow` framework.

Finally, in Chapter 5, we introduced the block Gibbs sampling. This algorithm makes use of the fact that each parameter, conditioned on other parameters and data, has a closed form distribution. This sampling algorithm is fast and can be parallelized over the samples. One key contribution we show in Chapter 5 is the proof of consistency for our model. The proof of consistency shows that eventually we will reach the true solution. However, we did not manage to prove the rate of convergence of each parameter even though we prove the asymptotic convergence of our method.

As stated in Chapter 1, interpretability is indeed a difficult concept to tackle. This thesis essentially contributes solutions to the problem by providing new algorithms and interesting use case.

6.3 Future Directions

6.3.1 Probably approximately correct learning

Probably approximately correct (PAC) learning is a theoretical computer science approach to analyze Machine learning algorithms [237]. In PAC-learning, one characterizes the

flexibility of models by means of Vapnik–Chervonenkis (VC) dimension, i.e. the maximum number of data points that can be perfectly fitted by the machine learning algorithm. A model with finite VC-dimension has a low generalization error [163].

In fact, for tree and forest models, the VC-dimension is known. One therefore can compute the generalization bound of the ABC samples. One can compute the generalization bound of the posterior under the ABC formulation. However, the true difficulty is to understand how the generalization bound is linked to variable selection. If this can be done, one then can prove a stronger theory on ABC-variable selection. This would be an important development for theory of ABC-methods in general.

6.3.2 Stochastic Gradient descent as Monte Carlo

In Chapter 4, we have introduced stochastic gradient descent as an optimization method for our transcriptional control model. Interestingly, we note that stochastic gradient descent can be formulated as a Monte Carlo simulation for posterior sampling [156, 244]. In particular, using L2 distance as a loss function in Chapter 4 means that we are taking a Gaussian noise assumption. In fact, mathematically we are trying minimize the KL-divergence between the empirical conditional distribution (condition on DNA sequence, protein concentrations, and roles of the protein in the transcription process) and the conditional distribution generated by the model. This allows us to sample from the posterior distribution using stochastic gradient descent with a constant step size [156]. Using the samples collected, we are able to make statistical statements about the errors associated with each parameter.

In Chapter 4, we focus on estimating the parameters associated with different roles in transcriptional regulation. However, it is difficult to compute the higher moments associated with parameters of activation, quenching, and co-activation conditional on the data [147]. Bayesian methods can solve this problem through computing the higher moment estimators from the posterior samples but there is no clear sampling method for discrete values such as roles of each transcription factor. While these factors are known through independent

experiments for *Drosophila*, this is not possible for most modern genomic scale datasets.

A simple solution is to consider an algorithm similar to that developed in Chapter 2. The first step is to sample from a prior distribution the role of each transcription factor. The second step is to randomly split the data into two segments: one for training using stochastic gradient descent and the other for computing a set of ABC error ϵ. Lastly, one collects samples with the smallest ϵ.

However, one key hurdle lies in setting up a prior for the role of each transcription factor. One can give equal probability to each configuration of transcription factor roles but then the number of possible combination of roles is exponential to number of transcription factors. Proper understanding of the construction of priors is an essential problem that requires our attention.

6.3.3 Implementation of transcription model

Our implementation of the transcription model in Chapter 4 suffers from rather slow computational speed compared to more developed DNN system. One key reason is that `Tensorflow` is a relatively opaque system developed for convenient implementation of DNN. While it has strong enterprise support by Google, it is not designed for specialized DNN structure of which we saw in Chapter 4. In addition, Google do not provide any road map to how the code is run. A check on source code shows that it is designed such that functions and classes have multiple layers of embeddings and one finds it frustrating to narrow down on lines of the code has been run. It would more useful to adopt a more transparent system.

In fact, multiple attempts have been made to improve the training speed but we cannot determine where the bottleneck lies. Our first attempt improves hardware by using the `A100` GPU which is specifically designed for machine learning training. It did not improve the speed. We also investigate the proficiency of our code in `Tensorflow board` but the `Tensorflow board` shows that most of GPU is idle most of the time but is unable to pin point which line of code is the bottleneck.

`Pytorch` does come into our mind. It is another enterprise supported (by Meta) system which has gain traction in the recent years. Unfortunately, `Pytorch` has very poor support for its recurrent neural network implementation. It does not allow RNN to be customized beyond having matrix multiplication. As the Barr's algorithm is a very specialized algorithm that requires matrix multiplication, concatenation and dropping previous values. With the current iteration of `Pytorch`, an implementation of the transcription model is not possible.

The fact that we are using enterprise supported package demonstrates an interesting dilemma. In order the improve the speed of which one can implement the code (i.e. reduce the number of lines of code needed for implementation), one has to use an opaque system. However such systems are not designed so that one can improve the efficiency of training since the system is hard to profile on a line by line basis. Even with good profiling, editing the source code might prove to be fatal for the system since the multiple layers of embedding in the code might result in code breaking down after small changes. A possible direction forward is to consider the use of auto-differentiation packages in `Julia`. `Julia` is open-source with a strong community support. It is designed for highly efficient computing and seems to be more transparent. As an auto-differentiation package is sufficient for us to implement and optimize the model in an efficient manner, moving to `Julia` is an attractive idea. Furthermore, it is much easier to form collaborations with those who write the packages in `Julia` than employees who maintain the code in Google and Meta, a key factor to consider when one wants to embark on this project.

6.4 Conclusion

From the my perspective, trying to interpret models, especially with little scientific backing and much experimental noise, is always going to difficult. In fact, the integration of statistics into modern biology is often a compromise to find signals when actual experimentation is difficult if not impossible to do. To find useful signals, one often conduct drastic simplification of the actual physical interaction into linear models. This procedure is arguably

an attempt to understand first order Taylor expansion of the mathematical representation of the true underlying physical relationship at the point of interest.

However, as shown in Chapter 4, a predictive thermodynamic model can often be very complex and how much a linear model can be represent the true relationship between variables is debatable. With experimental noise, one would expect that there are many misrepresentation and biasness in the process. The author argues that there is a need to develop methods and theory to deal with different aspects of complex models such as parameter estimation and inferences under a wider set of assumptions. In this case, the model is not linear and the optimization problem associated with the estimation may not convex or even differentiable.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Simon Aeschbacher, Mark A Beaumont, and Andreas Futschik. A novel approach for choosing summary statistics in approximate bayesian computation. *Genetics*, 192(3):1027–1047, 2012.

[3] Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, 2012.

[4] Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33:831–838, 2015.

[5] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.

[6] C. D. Arnold, D. Gerlach, C. Stelzer, L. M. Boryn, M. Rath, and A. Stark. Genome-wide quantitative enhancer activity maps identified by STARR-seq. *Science*, pages 1074–1077, 2013. PMID:23328393 doi:10.1126/science.1232542.

[7] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Amr Alexandari, Sabrina Krueger, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, et al. Deep learning at base-resolution reveals motif syntax of the cis-regulatory code. *bioRxiv*, page 737981, 2019.

[8] Rina Foygel Barber and Emmanuel J Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.

[9] Stuart Barber, Jochen Voss, and Mark Webster. The rate of convergence for approximate Bayesian computation. *Electronic Journal of Statistics*, 9(1):80–105, 2015.

[10] Maria Maddalena Barbieri and James O Berger. Optimal predictive model selection. *The Annals of Statistics*, 32(3):870–897, 2004.

[11] Marilena Barbieri, James O Berger, Edward I George, and Veronika Rocková. The median probability model and correlated variables. *arXiv*, 2018.

[12] Marilena Barbieri, James O. Berger, Edward I. George, and Veronika Rockova. The median probability model and correlated variables. *Bayesian Analysis (to appear)*, 2020.

[13] K. A. Barr, C. Martinez, J. R. Moran, A. R. Kim, A. F. Ramos, and J. Reinitz. Synthetic enhancer design by *in silico* compensatory evolution reveals flexibility and constraint in *cis*-regulation. *BMC Systems Biology*, 11:116, 2017. PMID:29187214 PMCID:PMC5708098 doi:10.1186/s12918-017-0485-2.

[14] K. A. Barr and J. Reinitz. A sequence level model of an intact locus predicts the location and function of nonadditive enhancers. *PLoS One*, 12:e0180861, 2017. doi:10.1371/journal.pone.0180861. PMCID:PMC5513433.

[15] Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[16] James O Berger and Luis R Pericchi. The intrinsic Bayes factor for linear models. *Bayesian statistics*, 5:25–44, 1996.

[17] James O Berger and Luis R Pericchi. Training samples in objective Bayesian model selection. *The Annals of Statistics*, 32(3):841–869, 2004.

[18] E. Bertolino, J. Reinitz, and Manu. The analysis of novel distal Cebpa enhancers and silencers using a transcriptional model reveals the complex regulatory logic of hematopoietic lineage specification. *Developmental Biology*, 413:128–144, 2016. doi:10.1016/j.ydbio.2016.02.030. PMCID:PMC4878123.

[19] Anirban Bhattacharya, Antik Chakraborty, and Bani K Mallick. Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika*, 103(4):985, 2016.

[20] J. Bleich, A. Kapelner, E. George, and S. Jensen. Variable selection for BART: An application to gene regulation. *Annals of Applied Statistics*, 8(3):1750–1781, 2014.

[21] Justin Bleich, Adam Kapelner, Edward I George, and Shane T Jensen. Variable selection for BART: An application to gene regulation. *The Annals of Applied Statistics*, pages 1750–1781, 2014.

[22] Zvi Boger and Hugo Guterman. Knowledge extraction from artificial neural network models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 4, pages 3030–3035. IEEE, 1997.

[23] Leonard Bottolo, Sylvia Richardson, et al. Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis*, 5(3):583–618, 2010.

[24] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[25] Leo Breiman and Adele Cutler. Online manual for random forests, 2013.

[26] Leo Breiman, Jerome Friedman, RA Olshen, and Charles J Stone. *Classification and regression trees*. New York: Chapman and Hall, 1984.

[27] Philip J Brown, Marina Vannucci, and Tom Fearn. Multivariate Bayesian variable selection and prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(3):627–641, 1998.

[28] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning Theory*, 2009.

[29] Séebastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, 2013.

[30] C. Burns, J. Thomason, and W. Tansey. Interpreting black box models via hypothesis testing. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science*, 2020.

[31] D. S. Burz and S. D. Hanes. Isolation of mutations that disrupt cooperative DNA binding of the *Drosophila* Bicoid protein. *Journal of Molecular Biology*, 305:21–230, 2001.

[32] David S. Burz, Rolando Rivera-Pomar, Herbert Jaeckle, and Steven D. Hanes. Cooperative DNA-binding by Bicoid provides a mechanism for threshold-dependent gene activation in the *Drosophila* embryo. *The EMBO journal*, 17:5998–6009, 1998.

[33] Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: 'model-x' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2018.

[34] Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: 'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

[35] P. Carbonetto and M. Stephens. Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Analysis*, 7(1):73–108, 2012.

[36] Peter Carbonetto, Matthew Stephens, et al. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian analysis*, 7(1):73–108, 2012.

[37] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[38] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[39] Ismaël Castillo and Romain Mismer. Empirical Bayes analysis of spike and slab posterior distributions. *arXiv preprint arXiv:1801.01696*, 2018.

[40] Ismaël Castillo, Johannes Schmidt-Hieber, and Aad Van der Vaart. Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986–2018, 2015.

[41] Ismaël Castillo and Aad van der Vaart. Needles and straw in a haystack: Posterior concentration for possibly sparse sequences. *The Annals of Statistics*, 40(4):2069–2101, 2012.

[42] F. Celesti, A. Celesti, L. Carnevale, A. Galletta, S. Campo, A. Romano, P. Bramanti, and M. Villari. Big data analytics in genomics: The point on Deep Learning solutions. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 306–309, 2017.

[43] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[44] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, 2013.

[45] Hugh Chipman, Edward I George, and Robert E McCulloch. The practical implementation of Bayesian model selection. In *Model Selection*, pages 65–116. Institute of Mathematical Statistics, 2001.

[46] Hugh Chipman, Edward I. George, and Robert E. McCulloch. The Practical Implementation of Bayesian Model Selection. In *Institute of Mathematical Statistics Lecture Notes - Monograph Series*. Institute of Mathematical Statistics, 2001.

[47] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

[48] Hugh A Chipman, Edward I George, and Robert E McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[49] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[50] François Chollet et al. Keras, 2015.

[51] Richard Combes and Alexandre Proutiere. Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*, 2014.

[52] Laëtitia Comminges and Arnak S Dalalyan. Tight conditions for consistency of variable selection in the context of high dimensionality. *The Annals of Statistics*, 40(5):2667–2696, 2012.

[53] Katalin Csillery, Michael GB Blum, Oscar E Gaggiotti, and Olivier Francois. Approximate Bayesian computation (ABC) in practice. *Trends in ecology \& evolution*, 25(7):410–418, 2010.

[54] Josh T Cuperus, Benjamin Groves, Anna Kuchina, Alexander B Rosenberg, Nebojsa Jojic, Stanley Fields, and Georg Seelig. Deep learning of the regulatory grammar of yeast 5 untranslated regions from 500,000 random sequences. *Genome research*, 27(12):2015–2024, 2017.

[55] Hal Daumé III, Daniel Marcu, and William Cohen. A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, 6(9), 2005.

[56] David GT Denison, Bani K Mallick, and Adrian FM Smith. A bayesian cart algorithm. *Biometrika*, 85(2):363–377, 1998.

[57] Maria Anna Di Lucca, Alessandra Guglielmi, Peter Müller, and Fernando A Quintana. A simple class of bayesian nonparametric autoregression models. *Bayesian Analysis (Online)*, 8(1):63, 2013.

[58] Bradley Efron. Bayesian inference and the parametric bootstrap. *The Annals of Applied Statistics*, 6(4):1971, 2012.

[59] Bradley Efron. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge University Press, 2012.

[60] Bradley Efron and Robert Tibshirani. The problem of regions. *The Annals of Statistics*, pages 1687–1718, 1998.

[61] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588, 1995.

[62] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

[63] Conor Fahy and Shengxiang Yang. Dynamic feature selection for clustering high dimensional data streams. *IEEE Access*, 7:127128–127140, 2019.

[64] Walid D. Fakhouri, Ahmet Ay, Rupindar Sayal, Jacqueline Dresch, Evan Dayringer, and David N. Arnosti. Deciphering a transcriptional regulatory code: modeling short-range repression in the *Drosophila embryo*. *Molecular Systems Biology*, 6:341, 2010. PMID:20087339 PMCID:PMC2824527 doi:10.1038/msb.2009.97.

[65] Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021.

[66] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[67] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

[68] Joseph Felsenstein. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 39(4):783–791, 1985.

[69] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[70] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[71] Dean P Foster and Robert A Stine. α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):429–444, 2008.

[72] David T Frazier, Gael M Martin, Christian P Robert, and Judith Rousseau. Asymptotic properties of approximate Bayesian computation. *Biometrika*, 105(3):593–607, 2018.

[73] David T Frazier, Christian P Robert, and Judith Rousseau. Model misspecification in approximate Bayesian computation: consequences and diagnostics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2020.

[74] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. New York: Springer Series in Statistics, 2001.

[75] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

[76] Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[77] Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–141, 1991.

[78] M. Fujioka, P. Miskiewicz, L. Raj, A. A. Gulledge, M. Weir, and T. Goto. *Drosophila* Paired regulates late *even-skipped* expression through a composite binding site for the paired domain and the homeodomain. *Development*, 122:2697–2707, 1996.

[79] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

[80] G David Garson. A comparison of neural network and expert systems algorithms with common multivariate procedures for analysis of social science data. *Social Science Computer Review*, 9(3):399–434, 1991.

[81] G David Garson. Interpreting neural-network connection weights. *AI expert*, 6(4):46–51, 1991.

[82] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[83] Edward I George and Robert E McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[84] Edward I George and Robert E McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.

[85] Subhashis Ghosal, Jüri Lember, and Aad Van Der Vaart. Nonparametric bayesian model selection and averaging. *Electronic Journal of Statistics*, 2:63–89, 2008.

[86] Subhashis Ghosal and Aad Van Der Vaart. Convergence rates of posterior distributions for noniid observations. *The Annals of Statistics*, 35(1):192–223, 2007.

[87] Subhashis Ghosal and Aad Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

[88] Jayanta K Ghosh and Tapas Samanta. Nonsubjective bayes testing?an overview. *Journal of statistical planning and inference*, 103(1-2):205–223, 2002.

[89] Michael Goldstein. Subjective bayesian analysis: principles and practice. *Bayesian analysis*, 1(3):403–420, 2006.

[90] Isidore Jacob Good. Probability and the weighing of evidence. *Philosophy*, 26(97), 1950.

[91] R. Gramacy and H. Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103:1119–11303, 2008.

[92] S. Gray, P. Szymanski, and M. Levine. Short-range repression permits multiple enhancers to function autonomously within a complex promoter. *Genes and Development*, 8:1829–1838, 1994.

[93] Peyton Greenside, Tyler Shimko, Polly Fordyce, and Anshul Kundaje. Discovering epistatic feature interactions from neural network models of regulatory dna sequences. *Bioinformatics*, 34(17):i629–i637, 2018.

[94] Y. Guan and M. Stephens. Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *The Annals of Applied Statistics*, 5:1780–1815, 2011.

[95] S Gupta, G Joshi, and O Yagan. Correlated multi-armed bandits with a latent random source. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[96] E. E. Hare, B. K. Peterson, and M. B. Eisen. A careful look at binding site reorganization in the *even-skipped* enhancers of *Drosophila* and sepsids. *PLoS Genetics*, 4(11):e1000268, 2008. PMCID:PMC2582681.

[97] E. E. Hare, B. K. Peterson, V. N. Iyer, R. Meier, and M. B. Eisen. Sepsid *even-skipped* enhancers are functionally conserved in *Drosopila* despite lack of sequence conservation. *PLoS Genetics*, 4:e1000106, 2008. PMCID:PMC2430619.

[98] X. He, M. A. H. Samee, C. Blatti, and S. Sinha. Thermodynamics-based models of transcriptional regulation by enhancers: The roles of synergistic activation, cooperative binding and short-range repression. *PLoS Computational Biology*, 6:e1000935, 2010. PMCID:PMC2940721.

[99] G. F. Hewitt, B. Strunk, C. Margulies, T. Priputin, X. D. Wang, R. Amey, B. Pabst, D. Kosman, J. Reinitz, and D. N. Arnosti. Transcriptional repression by the *Drosophila* Giant protein: Cis element positioning provides an alternative means of interpreting an effector gradient. *Development*, 126:1201–1210, 1999.

[100] J. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20:217–240, 2011.

[101] J. Hill, A. Linero, and J. Murray. Bayesian additive regression trees: A review and look forward. *Annual Review of Statistics and Its Application*, 7:251–278, 2020.

[102] Giles Hooker. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732, 2007.

[103] Enguerrand Horel and Kay Giesecke. Towards explainable AI: Significance tests for neural networks. arXiv:1902.06021, 2019.

[104] Garth R. Ilsley, Jasmin Fisher, Rolf Apweiler, Angela H. Depace, and Nicholas M. Luscombe. Cellular resolution models for even skipped regulation in the entire Drosophila embryo. *Elife*, 2:e00522, 2013.

[105] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[106] Hemant Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

[107] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[108] J. Jaeger, S. Surkova, M. Blagov, H. Janssens, D. Kosman, K. N. Kozlov, Manu, E. Myasnikova, C. E. Vanario-Alonso, M. Samsonova, D. H. Sharp, and J. Reinitz. Dynamic control of positional information in the early *Drosophila* embryo. *Nature*, 430:368–371, 2004.

[109] H. Janssens, S. Hou, J. Jaeger, A. R. Kim, E. Myasnikova, D. Sharp, and J. Reinitz. Quantitative and predictive model of transcriptional control of the *Drosophila melanogaster even skipped* gene. *Nature Genetics*, 38:1159–1165, 2006.

[110] H. Janssens, D. Kosman, C. E. Vanario-Alonso, J. Jaeger, M. Samsonova, and J. Reinitz. A high-throughput method for quantifying gene expression data from early *Drosophila* embryos. *Development, Genes and Evolution*, 215:374–381, 2005.

[111] Adel Javanmard, Andrea Montanari, et al. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of Statistics*, 46(2):526–554, 2018.

[112] Emil Jeřábek. Dual weak pigeonhole principle, boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 129(1-3):1–37, 2004.

[113] B. Jiang, T. Wu, C. Zheng, and W. Wong. Learning summary statistic for approximate bayesian computation via deep neural network. *Statistica Sinica*, 27:1595–1618, 2017.

[114] Bai Jiang, Tung-yu Wu, Charles Zheng, and Wing H Wong. Learning summary statistic for approximate bayesian computation via deep neural network. *Statistica Sinica*, pages 1595–1618, 2017.

[115] Valen E Johnson and David Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.

[116] M. Kazemian, C. Blatti, A. Richards, M. McCutchan, N. Wakabayashi-Ito, A. S. Hammonds, S. E. Celniker, S. Kumar, S. A. Wolfe, M. H. Brodsky, and S. Sinha. Quantitative analysis of the *Drosophila* segmentation regulatory network using pattern generating potentials. *PLoS Biology*, 8:e1000456, 2010. PMCID:PMC2923081.

[117] Jalil Kazemitabar, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar. Variable importance using decision trees. In *Advances in Neural Information Processing Systems*, pages 425–434, 2017.

[118] A. R. Kim. *Breaking the genomic* cis*-regulatory code by an experimental and theoretical analysis of* eve *enhancer fusions*. PhD Thesis, Department of Biochemistry and Cell Biology, Stony Brook University, 2012.

[119] A. R. Kim, C. Martinez, J. Ionides, A. F. Ramos, M. Z. Ludwig, N. Ogawa, D. H. Sharp, and J. Reinitz. Rearrangements of 2.5 kilobases of noncoding DNA from the *Drosophila even-skipped* locus define predictive rules of genomic *cis*-regulatory logic. *PLoS Genetics*, 9:e1003243, 2013. PMCID:PMC3585115.

[120] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[121] M. Klingler, J. Soong, B. Butler, and J. P. Gergen. Disperse versus compact elements for the regulation of *runt* stripes in *Drosophila*. *Developmental Biology*, 177:73–84, 1996.

[122] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[123] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, 2015.

[124] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[125] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: fast combinatorial optimization with learning. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 420–429, 2014.

[126] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems*, 2015.

[127] John Lafferty and Larry Wasserman. Iterative Markov Chain Monte Carlo computation of reference priors and minimax risk. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 293–300. Morgan Kaufmann Publishers Inc., 2001.

[128] John Lafferty and Larry Wasserman. RODEO: sparse, greedy nonparametric regression. *The Annals of Statistics*, pages 28–63, 2008.

[129] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[130] D. Lebrecht, M. Foehr, E. Smith, F. J. P. Lopes, C. E. Vanario-Alonso, John Reinitz, D. S. Burz, and S. D. Hanes. Bicoid cooperative DNA binding is critical for embryonic patterning in *Drosophila*. *Proceedings of the National Academy of Sciences USA*, 102:13176–13181, 2005.

[131] Sunmin Lee, Jeong-Cheol Kim, Hyung-Sup Jung, Moung Jin Lee, and Saro Lee. Spatial prediction of flood susceptibility using random-forest and boosted-tree models in seoul metropolitan city, korea. *Geomatics, Natural Hazards and Risk*, 8(2):1185–1203, 2017.

[132] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[133] Jan Leike, Tor Lattimore, Laurent Orseau, and Marcus Hutter. Thompson sampling is asymptotically optimal in general environments. In *Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2016.

[134] Jüri Lember and Aad van der Vaart. On universal Bayesian adaptation. *Statistics & Decisions*, 25(2/2007):127–152, 2007.

[135] Fred B Lempers. Posterior probabilities of alternative linear models. Technical report, Rotterdam University, 1971.

[136] Wentao Li and Paul Fearnhead. Convergence of regression-adjusted approximate Bayesian computation. *Biometrika*, 105(2):301–318, 2018.

[137] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *FE@ NIPS*, pages 196–212, 2015.

[138] Faming Liang, Qizhai Li, and Lei Zhou. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association*, 113(523):955–972, 2018.

[139] Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.

[140] Maxwell W. Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16:321–332, 2015.

[141] Antonio Lijoi, Igor Prünster, and Stephen G Walker. On consistency of nonparametric normal mixtures for bayesian density estimation. *Journal of the American Statistical Association*, 100(472):1292–1296, 2005.

[142] Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

[143] Antonio R Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, pages 1–11, 2018.

[144] Antonio R Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636, 2018.

[145] Antonio R Linero and Yun Yang. Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1087–1110, 2018.

[146] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[147] Yi Liu, Kenneth Barr, and John Reinitz. Fully interpretable deep learning model of transcriptional control. *Bioinformatics*, 36(Supplement_1):i499–i507, 2020.

[148] Yi Liu, Veronika Rockova, and Yuexi Wang. ABC Variable Selection with Bayesian Forests. arXiv:1806.02304, 2018.

[149] Yi Liu, Veronika Ročková, and Yuexi Wang. Variable selection with abc bayesian forests. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(3):453–481, 2021.

[150] Yi Liu and Veronika Ročková. Variable selection via thompson sampling. *Journal of the American Statistical Association*, 0(0):1–18, 2021.

[151] Yuwen Liu, Shan Yu, Vineet K Dhiman, Tonya Brunetti, Heather Eckart, and Kevin P White. Functional assessment of human enhancer activities using whole-genome starr-sequencing. *Genome biology*, 18(1):219, 2017.

[152] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems*, 26:431–439, 2013.

[153] Yang Lu, Yingying Fan, Jinchi Lv, and William Stafford Noble. DeepPINK: reproducible feature selection in Deep Neural Networks. In *Advances in Neural Information Processing Systems*, 2018.

[154] X. Ma, D. Yuan, K. Diepold, T. Scarborough, and J. Ma. The *Drosophila* morphogenetic protein Bicoid binds DNA cooperatively. *Development*, 112:1195–1206, 1996.

[155] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9:2579–2605, 2008.

[156] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

[157] Manu, S. Surkova, A. V. Spirov, V. Gursky, H. Janssens, A. Kim, O. Radulescu, C. E. Vanario-Alonso, D. H. Sharp, M. Samsonova, and J. Reinitz. Canalization of gene expression in the *Drosophila* blastoderm by gap gene cross regulation. *PLoS Biology*, 7:e1000049, 2009. doi:10.371/journal.pbio.1000049 PMCID:PMC2653557.

[158] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[159] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[160] James S Martin, Ajay Jasra, Sumeetpal S Singh, Nick Whiteley, Pierre Del Moral, and Emma McCoy. Approximate Bayesian computation for smoothing. *Stochastic Analysis and Applications*, 32(3):397–420, 2014.

[161] Carlos Martinez, Ah-Ram Kim, Joshua S. Rest, Michael Ludwig, Martin Kreitman, Kevin White, and John Reinitz. Ancestral resurrection of the *Drosophila* S2E enhancer reveals accessible evolutionary paths through compensatory change. *Molecular Biology and Evolution*, 31:903–916, 2014. PMCID:PMC3969564.

[162] Masayoshi Mase, Art B Owen, and Benjamin Seiler. Explaining black box decisions by shapley cohort refinement. arXiv:1911.00467, 2019.

[163] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

[164] Robert McCulloch, Rodney Sparapani, Robert Gramacy, Charles Spanbauer, and Matthew Pratola. *BART: Bayesian Additive Regression Trees*, 2018. R package version 1.6.

[165] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

[166] Elías Moreno, Javier Girón, and George Casella. Posterior model consistency in variable selection as the model dimension grows. *Statistical Science*, 30(2):228–241, 2015.

[167] Rajiv Movva, Peyton Greenside, Georgi K Marinov, Surag Nair, Avanti Shrikumar, and Anshul Kundaje. Deciphering regulatory dna sequences and noncoding genetic variants using neural network models of massively parallel reporter assays. *PloS one*, 14(6), 2019.

[168] Surag Nair, Daniel S Kim, Jacob Perricone, and Anshul Kundaje. Integrating regulatory dna sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts. *bioRxiv*, page 605717, 2019.

[169] Naveen Naidu Narisetty and Xuming He. Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2):789–817, 2014.

[170] Michael A Newton and Adrian E Raftery. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–48, 1994.

[171] M. B. Noyes, X. Meng, A. Wakabayashi, S. Sinha, M. H. Brodsky, and S. A. Wolfe. A systematic characterization of factors that regulate *drosophila* segmentation via a bacterial one-hybrid system. *Nucleic Acids Research*, pages 1–14, 2008.

[172] Matthew A Nunes and David J Balding. On optimal selection of summary statistics for approximate bayesian computation. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

[173] Anthony O'Hagan. Fractional bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):99–118, 1995.

[174] Julian D Olden and Donald A Jackson. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154(1-2):135–150, 2002.

[175] Akio Onogi, Masanobu Nurimoto, and Mitsuo Morita. Characterization of a bayesian genetic clustering algorithm based on a dirichlet process prior and comparison among bayesian clustering methods. *BMC bioinformatics*, 12(1):1–16, 2011.

[176] N. Orgawa and M. D. Biggin. High-Throughput SELEX Determination of DNA Sequences Bound by Transcription Factors In Vitro. *Methods in Molecular Biology*, 786:51–63, 2012. PMID:21938619 doi:10.1007/978-1-61779-292-2_3.

[177] Art B Owen and Clémentine Prieur. On Shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002, 2017.

[178] Weishen Pan and Changshui Zhang. The definitions of interpretability and learning of interpretable models. *arXiv preprint arXiv:2105.14171*, 2021.

[179] Sandeep Pandey, Deepayan Chakrabarti, and Deepak Agarwal. Multi-armed bandit problems with dependent arms. In *International Conference on Machine learning*, 2007.

[180] Evan Patterson and Matteo Sesia. *Knockoff: The Knockoff Filter for Controlled Variable Selection*. Statistics Department, Stanford University, 2018. R package version 0.3.2.

[181] Rupali P Patwardhan, Choli Lee, Oren Litvin, David L Young, Dana Pe'er, and Jay Shendure. High-resolution analysis of dna regulatory elements by synthetic saturation mutagenesis. *Nature biotechnology*, 27(12):1173, 2009.

[182] José M Pérez and James O Berger. Expected-posterior prior distributions for model selection. *Biometrika*, 89(3):491–512, 2002.

[183] Virginia L Pimmett, Matthieu Dejean, Carola Fernandez, Antonio Trullo, Edouard Bertrand, Ovidiu Radulescu, and Mounia Lagha. Quantitative imaging of transcription in living drosophila embryos reveals the impact of core promoter motifs on promoter state dynamics. *Nature communications*, 12(1):1–16, 2021.

[184] Vincent Plagnol and Simon Tavaré. Approximate Bayesian Computation and MCMC. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 99–113. Springer, 2004.

[185] F. Pouladi, H. Salehinejad, and A. M. Gilani. Recurrent Neural Networks for Sequential Phenotype Prediction in Genomics. In *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, pages 225–230, 2015.

[186] Matthew T Pratola. Efficient Metropolis–Hastings proposal mechanisms for Bayesian regression tree models. *Bayesian Analysis*, 11(3):885–911, 2016.

[187] Pierre Pudlo, Jean-Michel Marin, Arnaud Estoup, Jean-Marie Cornuet, Mathieu Gautier, and Christian P Robert. Reliable ABC model choice via random forests. *Bioinformatics*, 32(6):859–866, 2015.

[188] Peter Radchenko and Gareth M James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105(492):1541–1553, 2010.

[189] Aaditya Ramdas, Fanny Yang, Martin J Wainwright, and Michael I Jordan. Online control of the false discovery rate with decaying memory. In *Advances In Neural Information Processing Systems*, pages 5650–5659, 2017.

[190] Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

[191] J. Reinitz, S. Hou, and D. H. Sharp. Transcriptional control in *Drosophila*. *ComPlexUs*, 1:54–64, 2003.

[192] J. Reinitz and D. H. Sharp. Mechanism of *eve* stripe formation. *Mechanisms of Development*, 49:133–158, 1995.

[193] Andrea Repele, Shawn Krueger, Tapas Bhattacharyya, and Michelle Y Tuineau. The regulatory control of cebpa enhancers and silencers in the myeloid and red-blood cell lineages. *PloS one*, 14(6):e0217580, 2019.

[194] Soo-Yon Rhee, W Jeffrey Fessel, Andrew R Zolopa, Leo Hurley, Tommy Liu, Jonathan Taylor, Dong Phuong Nguyen, Sally Slome, Daniel Klein, and Michael Horberg. Hiv-1 protease and reverse-transcriptase mutations: correlations with antiretroviral therapy in subtype b isolates and implications for drug-resistance surveillance. *The Journal of infectious diseases*, 192(3):456–465, 2005.

[195] Soo-Yon Rhee, Jonathan Taylor, Gauhar Wadhera, Asa Ben-Hur, Douglas L Brutlag, and Robert W Shafer. Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences*, 103(46):17355–17360, 2006.

[196] Christian P Robert, Jean-Marie Cornuet, Jean-Michel Marin, and Natesh S Pillai. Lack of confidence in approximate Bayesian computation model choice. *Proceedings of the National Academy of Sciences*, 108(37):15112–15117, 2011.

[197] Veronika Ročková. Particle em for variable selection. *Journal of the American Statistical Association*, 2018.

[198] Veronika Rockova and Edward I. George. EMVS: The EM Approach to Bayesian Variable Selection. *Journal of the American Statistical Association*, 109(506):828–846, 2014.

[199] Veronika Ročková and Edward I George. EMVS: The EM approach to Bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846, 2014.

[200] Veronika Rockova and Edward I. George. The spike-and-slab LASSO. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

[201] Veronika Ročková and Edward I George. The Spike-and-Slab LASSO. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

[202] Veronika Ročková and Stephanie van der Pas. Posterior concentration for Bayesian regression trees and their ensembles. *The Annals of Statistics (in revision)*, 2017.

[203] David Rossell and Donatello Telesca. Nonlocal priors for high-dimensional estimation. *Journal of the American Statistical Association*, 112(517):254–265, 2017.

[204] E. Roulet, S. Busso, A. A. Camargo, A. J. G. Simpson, N. Mermod, and P. Bucher. High-throughput SELEX SAGE method for quantitative modeling of transcription-factor binding sites. *Nature Biotechnology*, 20:831–835, 2002.

[205] V. Ročková. Particle EM for variable selection. *Journal of the American Statistical Association*, pages 1–30, 2017.

[206] V. Ročková and E. Saha. On theory for BART. In *Artificial Intelligence and Statistics*, 2019.

[207] Xu Rui, D.C. Wunsch, and R.L. Frank. Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4:681–692, 2007.

[208] Daniel Russo. Simple Bayesian algorithms for best arm identification. In *Conference on Learning Theory*, 2016.

[209] M. A. H. Samee and S. Sinha. Quantitative modeling of a gene's expression from its intergenic sequence. *PLoS Computational Biology*, 10:1–21, 2014.

[210] Terrance Savitsky, Marina Vannucci, and Naijun Sha. Variable selection for non-parametric Gaussian process priors: Models and computational strategies. *Statistical Science*, 26(1):130, 2011.

[211] R. Sayal, J. M. Dresch, I. Pushel, B. R. Taylor, and D. Arnosti. Quantitative perturbation-based analysis of gene expression predicts enhancer activity in early *Drosophila* embryo. *eLife*, 5:e08445, 2016. PMID:27152947 PMCID:PMC4859806 doi:10.7554/eLife.08445.

[212] Fabian Scheipl. spikeslabgam: Bayesian variable selection, model choice and regularization for generalized additive mixed models in r. *arXiv preprint arXiv:1105.5253*, 2011.

[213] James G Scott and James O Berger. Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, pages 2587–2619, 2010.

[214] E. Segal, T. Raveh-Sadka, M. Schroeder, U. Unnerstall, and U. Gaul. Predicting expression patterns from regulatory sequence in *Drosophila* segmentation. *Nature*, 451:535–540, 2008. PMID:18172436 doi:10.1038/nature06496.

[215] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[216] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[217] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[218] Jingxiang Shen, Mariela D. Petkova, Feng Liu, and Chao Tang. Toward deciphering developmental patterning with deep neural network. *bioRxiv*, page 374439, 2018.

[219] S. Small, D. N. Arnosti, and M. Levine. Spacing ensures autonomous expression of different stripe enhancers in the *even-skipped* promoter. *Development*, 119:767–772, 1993.

[220] S. Small, A. Blair, and M. Levine. Regulation of *even-skipped* stripe 2 in the *Drosophila* embryo. *The EMBO Journal*, 11:4047–4057, 1992.

[221] S. Small, A. Blair, and M. Levine. Regulation of two pair-rule stripes by a single enhancer in the *Drosophila* embryo. *Developmental Biology*, 175:314–324, 1996.

[222] R. P. Smith, L. Taher, R. P. Patwardhan, M. J. Kim, F. Inoue, J. Shendure, I. Ovcharenko, and N. Ahituv. Massively parallel decoding of mammalian regulatory sequences supports a flexible organizational model. *Nature Genetics*, 45:1021–1028, 2013. PMID:23892608 PMCID:PMC3775494 doi:10.1038/ng.2713.

[223] D. Stanojevic, S. Small, and M. Levine. Regulation of a segmentation stripe by overlapping activators and repressors in the *Drosophila* embryo. *Science*, 254:1385–1387, 1991.

[224] Mikael Sunnaaker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate bayesian computation. *PLoS computational biology*, 9(1):e1002803, 2013.

[225] S. Surkova, D. Kosman, K. Kozlov, Manu, E. Myasnikova, A. Samsonova, A. Spirov, C. E. Vanario-Alonso, M. Samsonova, and J. Reinitz. Characterization of the *Drosophila* segment determination morphome. *Developmental Biology*, 313(2):844–862, 2008. PMCID:PMC2254320.

[226] M. Taddy, R. Gramacy, and N. Polson. Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106:409–123, 2011.

[227] Matthew A Taddy, Robert B Gramacy, and Nicholas G Polson. Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106(493):109–123, 2011.

[228] Katjana Tantale, Encar Garcia-Oliver, Marie-Cécile Robert, Adèle L'hostis, Yueyuxiao Yang, Nikolay Tsanov, Rachel Topno, Thierry Gostan, Alja Kozulic-Pirher, Meenakshi Basu-Shrivastava, et al. Stochastic pausing at latent hiv-1 promoters generates transcriptional bursting. *Nature communications*, 12(1):1–20, 2021.

[229] Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997.

[230] Henry Teicher. Identifiability of finite mixtures. *The annals of Mathematical statistics*, pages 1265–1269, 1963.

[231] Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2018. R package version 4.1-13.

[232] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two sample. *Biometrika*, 25(3/4):285–294, 1933.

[233] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[234] Robert Tibshirani. Regression shrinkage and selection via the LASSO: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

[235] Berwin A Turlach. Discussion on least angle regression. *The Annals of Statistics*, 32(2):481–490, 2004.

[236] Jacob C Ulirsch, Satish K Nandakumar, Li Wang, Felix C Giani, Xiaolan Zhang, Peter Rogov, Alexandre Melnikov, Patrick McDonel, Ron Do, Tarjei S Mikkelsen, et al. Systematic functional dissection of common genetic variation affecting red blood cell traits. *Cell*, 165(6):1530–1545, 2016.

[237] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[238] Sara A Van De Geer and Peter Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

[239] Stephanie van der Pas, James Scott, Antik Chakraborty, and Anirban Bhattacharya. *horseshoe: Implementation of the Horseshoe Prior*, 2019. R package version 0.2.0.

[240] Stéphanie van der Pas, Botond Szabó, Aad van der Vaart, et al. Uncertainty quantification for the horseshoe (with discussion). *Bayesian Analysis*, 12(4):1221–1274, 2017.

[241] Marina Vannucci and Francesco C Stingo. Bayesian models for variable selection that incorporate biological information. *Bayesian Statistics*, 9:1–20, 2010.

[242] Jing Wang, Jie Shen, and Ping Li. Provable variable selection for streaming features. In *International Conference on Machine Learning*, 2018.

[243] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 5114–5122, 2018.

[244] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[245] Yun Yang and Debdeep Pati. Bayesian model selection consistency and oracle inequality with intractable marginal likelihood. *arXiv preprint arXiv:1701.00311*, 2017.

[246] Mao Ye and Yan Sun. Variable selection via penalized neural network: a drop-out-one loss approach. In *International Conference on Machine Learning*, 2018.

[247] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[248] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[249] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8827–8836, 2018.

[250] Tao Zhang, Shuzhi Sam Ge, and Chang Chieh Hang. Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. *Automatica*, 36(12):1835–1846, 2000.

[251] Peng Zhao and Bin Yu. On model selection consistency of LASSO. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

[252] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[253] Jing Zhou, Dean P Foster, Robert A Stine, and Lyle H Ungar. Streamwise feature selection. *Journal of Machine Learning Research*, 7:1861–1885, 2006.

[254] Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.

APPENDIX A

APPENDIX FOR ABC FOR NON-PARAMETRIC INFERENCE

A.1 Theory of ABC

A.1.1 Proof of Theorem 2.5.1

We first review some notation used throughout this section and adapted from [202]. Recall that $\Pi_{\mathcal{S}}(\cdot)$ denotes the conditional distribution given the model $\mathcal{S}$. Next, $\mathcal{F}_{\mathcal{S}}(K)$ denotes a set of all step functions $f_{\mathcal{T},\boldsymbol{\beta}}(\cdot)$ with K steps that split on covariates $\mathcal{S}$ and $\|f_{\mathcal{T},\boldsymbol{\beta}}\|_\infty \leq B$. A tree partition is called valid when each tree splits on observed values $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and has nonempty cells. We denote with $\mathcal{V}_{\mathcal{S}}^K$ all valid trees obtained by splitting $K-1$ times along coordinates inside $\mathcal{S}$. The number of such valid trees is denoted with $\Delta(\mathcal{V}_{\mathcal{S}}^K)$. For a valid tree partition $\mathcal{T} \in \mathcal{V}_{\mathcal{S}}^K$, we denote with $\mathcal{F}(\mathcal{T}) \subset \mathcal{F}_{\mathcal{S}}(K)$ all step functions supported on $\mathcal{T}$. We prove Theorem 2.5.1 by verifying conditions B1-B4 in Theorem 4 of [245] (further referred to as YP17). We build on tools developed in [202] (further referred to as RP17).

Prior Concentration Condition

The first condition pertains to prior concentration and consists of two parts: (a) the model prior mass condition and (b) the prior concentration condition in the parameter space under the true model. Namely, we want to show that

$$\pi(\mathcal{S}_0) \geq \mathbf{e}^{-n\,\varepsilon_{n,\mathcal{S}_0}^2} \tag{A.1}$$

and

$$\Pi_{\mathcal{S}_0}\left(f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}_0}(K) : \|f_{\mathcal{T},\boldsymbol{\beta}} - f_0\|_n \leq \varepsilon_{n,\mathcal{S}_0}\right) \geq \mathbf{e}^{-d\,n\,\varepsilon_{n,\mathcal{S}_0}^2} \tag{A.2}$$

for some $d > 2$. The prior concentration (A.1) follows directly from the definition of model weights (2.13) for $C \leq C_\varepsilon^2$ under our assumption $q_0 \log p < n^{q_0/(2\alpha+q_0)}$.

Regarding (A.2), a variant of this condition is verified in Section 8.2 of RP17 assuming that K is random with a prior. It follows from their proof, however, that (A.2) holds if we fix K at $K_{\mathcal{S}_0} = \lfloor C_K / C_\varepsilon^2 \, n \, \varepsilon_{n,\mathcal{S}_0}^2 / \log n \rfloor = 2^{q_0 s}$ for some $s \in \mathbb{N}$. The proof consists of (a) constructing a single approximating tree (i.e. the k-d tree with $s = (\log_2 K_{\mathcal{S}_0})/q_0$ cycles of splits on each coordinate in $\mathcal{S}_0$) and showing that it has enough prior support. This tree exists under the assumption that the design is $\mathcal{S}_0$-regular. From (8.5) of RP17, such tree approximates f_0 with an error bounded by a constant multiple of $\varepsilon_{n,\mathcal{S}_0}$. The verification of (A.2) then follows directly from RP17.

Entropy Condition

The second condition (B4 in the notation of YP17) entails controlling the complexity of over/underfitting models. In the sequel, we focus only on models with up to q_n covariates, where $q_n = C_q \lceil n \, \varepsilon_{n,\mathcal{S}_0}^2 / \log p \rceil$. This restriction is justified by the following lemma.

Lemma A.1.1. *Denote with $q_n = C_q \lceil n \, \varepsilon_{n,\mathcal{S}_0}^2 / \log p \rceil$. Under the assumptions of Theorem 2.5.1, we have*

$$\Pi(q \geq q_n \mid \boldsymbol{Y}^{(n)}) \to 0 \tag{A.3}$$

in $\mathbb{P}_{f_0}^{(n)}$-probability as $n \to \infty$.

Proof: First, we show that $\Pi(q \geq q_n) \mathbf{e}^{(d+2) n \, \varepsilon_{n,\mathcal{S}_0}^2} \to 0$, where $d > 2$ is as in (A.2). We can write

$$\begin{aligned}
\Pi(q > q_n) \mathbf{e}^{(d+2) n \varepsilon_{n,\mathcal{S}_0}^2} &\lesssim \mathbf{e}^{(d+2) n \, \varepsilon_{n,\mathcal{S}_0}^2} \sum_{k=q_n}^{p} \tbinom{p}{k} \mathbf{e}^{-C \times \max\{n^{k/(2\alpha+k)} \log n, k \log p\}} \\
&\leq \mathbf{e}^{(d+2) n \, \varepsilon_{n,\mathcal{S}_0}^2 - (C-2) \, q_n \log p} = \mathbf{e}^{-n \, \varepsilon_{n,\mathcal{S}_0}^2 [(C-2) C_q - (d+2)]}.
\end{aligned}$$

The right hand side above goes to zero when $(C-2)C_q - (d+2) > 0$. This can be satisfied with $C > 2$ and C_q large enough. This fact, together with prior mass conditions (A.2) and (A.1), yields (A.3) according to Lemma 1 of [86].

Lemma A.1.1 essentially states that the posterior will not reward models whose dimensionality is larger than (or equal to) q_n. In our following considerations, we thus condition only models with less than q_n variables.

We now verify that the complexity of overfitting models $\mathcal{S} \supset \mathcal{S}_0$ is not too large in the sense that their global metric entropy satisfies

$$\log N\left(\varepsilon_{n,\mathcal{S}}\,;\;\mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}})\,;\;\|\cdot\|_n\right) \leq n\,\varepsilon_{n,\mathcal{S}}^2. \tag{A.4}$$

First, we note that for two tree step functions $f_{\mathcal{T},\boldsymbol{\beta}_1} \in \mathcal{F}(\mathcal{T})$ and $f_{\mathcal{T},\boldsymbol{\beta}_2} \in \mathcal{F}(\mathcal{T})$ that have the same partition $\mathcal{T} \in \mathcal{V}_{\mathcal{S}}^{K_{\mathcal{S}}}$ and different step heights $\boldsymbol{\beta}_1 \in \mathbb{R}^{K_{\mathcal{S}}}$ and $\boldsymbol{\beta}_2 \in \mathbb{R}^{K_{\mathcal{S}}}$, we have $\{\|f_{\mathcal{T},\boldsymbol{\beta}_1} - f_{\mathcal{T},\boldsymbol{\beta}_2}\|_n \leq \varepsilon_{n,\mathcal{S}}\} \supset \{\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_2 \leq \varepsilon_{n,\mathcal{S}}\}$. Furthermore, noting that $\mathcal{F}(\mathcal{T}) = \{f_{\mathcal{T},\boldsymbol{\beta}} : \|f_{\mathcal{T},\boldsymbol{\beta}}\|_\infty \leq B\} \subset \{\boldsymbol{\beta} \in \mathbb{R}^{K_{\mathcal{S}}} : \|\boldsymbol{\beta}\|_2 \leq B\sqrt{n}\}$ we can write

$$N\left(\varepsilon_{n,\mathcal{S}}\,;\;\mathcal{F}(\mathcal{T})\,;\;\|\cdot\|_n\right) \leq \left(\frac{3\,B\sqrt{n}}{\varepsilon_{n,\mathcal{S}}}\right)^{K_{\mathcal{S}}} \leq \left(3\,B\,n^{3/2}/C_\varepsilon\right)^{K_{\mathcal{S}}},$$

where we used the standard $\varepsilon_{n,\mathcal{S}}$ covering number of a $K_{\mathcal{S}}$-Euclidean ball of a radius $B\sqrt{n}$ and the fact that $1/\varepsilon_{n,\mathcal{S}} \leq 1/C_\varepsilon \times n^{\alpha/(2\alpha+|\mathcal{S}|)} \leq 1/C_\varepsilon \times n$. Then we can write

$$N\left(\varepsilon_{n,\mathcal{S}}\,;\;\mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}})\,;\;\|\cdot\|_n\right) \leq \Delta(\mathcal{V}_{\mathcal{S}}^{K_{\mathcal{S}}})\left(3\,B\,n^{3/2}/C_\varepsilon\right)^{K_{\mathcal{S}}}.$$

Using Lemma 3.1 of Rockova and van der Pas (2017), we have $\Delta(\mathcal{V}_{\mathcal{S}}^{K_{\mathcal{S}}}) \leq (K_{\mathcal{S}}\,n \mid \mathcal{S} \mid)^{K_{\mathcal{S}}}$.

The overall log-covering number is then upper-bounded with (since $\mid \mathcal{S} \mid \leq q_n \leq n$)

$$K_{\mathcal{S}} \log\left(3\,B\,n^3\,n^{3/2}\right) \lesssim K_{\mathcal{S}} \log n \propto n\,\varepsilon_{n,\mathcal{S}}^2. \tag{A.5}$$

This verifies the model complexity condition for overfitting models. Next, we need to verify (A.4) with $\varepsilon_{n,\mathcal{S}}$ replaced by $\widetilde{\varepsilon}_n$ for "underfitting" models $\mathcal{S} \in \boldsymbol{\Gamma}_{\mathcal{S} \not\supset \mathcal{S}_0}$ where $\mid \mathcal{S} \mid \leq q_n$. This follows from the same arguments as above and the fact that $\varepsilon_{n,\mathcal{S}} \leq \widetilde{\varepsilon}_n$. Finally, the last

requirement in Assumption B4 of YP17 is verifying that

$$\sum_{\mathcal{S}\not\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \mathbf{e}^{-C_2\, n\, \widetilde{\varepsilon}_n^2} + \sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \mathbf{e}^{-C_2\, n\, \varepsilon_{n,\mathcal{S}}^2} \leq 1 \tag{A.6}$$

for some large constant $C_2 > 0$. Since $\widetilde{\varepsilon}_n \geq \varepsilon_{n,\mathcal{S}} > \varepsilon_{n,\mathcal{S}_0}$ for any $\mathcal{S} \supset \mathcal{S}_0$ such that $|\ \mathcal{S}\ | \leq q_n$, we can upper-bound the left-hand side above with

$$\sum_{q=0}^{q_n} \sum_{\mathcal{S}:|\mathcal{S}|=q} \mathbf{e}^{-C_2\, n\, \varepsilon_{n,\mathcal{S}_0}^2} \leq \mathbf{e}^{-C_2\, n\, \varepsilon_{n,\mathcal{S}_0}^2} \sum_{q=0}^{q_n} \tbinom{p}{q} \leq \left(\frac{2\,\mathbf{e}\,p}{q_n}\right)^{q_n+1} \mathbf{e}^{-C_2\, n\varepsilon_{n,\mathcal{S}_0}^2}$$

From our definition of q_n, we have $q_n \log p \asymp n\, \varepsilon_{n,\mathcal{S}_0}^2$ and (A.6) will be satisfied for a large enough C_2.

Prior Anticoncentration Condition

Lastly, as one of the sufficient conditions for model selection consistency, we need to verify

$$\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \pi(\mathcal{S})\, \Pi_{\mathcal{S}} \left(f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}}) : \|f_0 - f_{\mathcal{T},\boldsymbol{\beta}}\|_n \leq M\, \varepsilon_{n,\mathcal{S}}\right) \leq \mathbf{e}^{-H\, n\, \varepsilon_{n,\,\mathcal{S}_0}^2} \tag{A.7}$$

for some $H > 0$. Alternatively, YP17 introduce the so-called "anti-concentration condition" $\Pi_{\mathcal{S}} \left(f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}}) : \|f_0 - f_{\mathcal{T},\boldsymbol{\beta}}\|_n \leq M\, \varepsilon_{n,\mathcal{S}}\right) \leq \mathbf{e}^{-H\, n\, \varepsilon_{n,\,\mathcal{S}_0}^2}$ for overfitting models $\mathcal{S} \supset \mathcal{S}_0$ where $\varepsilon_{n,\mathcal{S}} \geq \varepsilon_{n,\mathcal{S}_0}$. This condition is needed to show that the posterior probability of more complex models that contain the truth goes to zero.

It turns out that this condition can be avoided with our choice of model weights $\pi(\mathcal{S})$ [85]. We can verify (A.7) directly (without the anticoncentration condition) by upper-bounding the left hand side of (A.7) with

$$\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \pi(\mathcal{S}) \leq \sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \mathbf{e}^{-C\, n\, \varepsilon_{n,\mathcal{S}}^2} \leq \mathbf{e}^{-C\, n\, \varepsilon_{n,\mathcal{S}_0}^2} \left(\frac{2\,\mathbf{e}\,p}{q_n}\right)^{q_n+1}. \tag{A.8}$$

Since $q_n \log p \asymp n\, \varepsilon^2_{n,\mathcal{S}_0}$, (A.7) holds for $H < C - 1$.

Identifiability

Under the identifiability and irrepresentability assumptions (2.5.1) and (2.5.2), it turns out that we cannot approximate f_0 well enough with models that miss at least one covariate. This property is summarized in the following Lemma, which is a variant of Proposition 1 of YP17.

Lemma A.1.2. *For $f_0 \in \mathcal{H}^\alpha_p \cap \mathcal{C}(\mathcal{S}_0)$, assume that $\mathcal{S}_0$ is (f_0, ε)-identifiable and that ε-irrepresentability holds. Then*

$$\inf_{\mathcal{S} \not\supset \mathcal{S}_0} \inf_{f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_\mathcal{S}} \|f_0 - f_{\mathcal{T},\boldsymbol{\beta}}\|_n > M\, \varepsilon.$$

Proof: We decompose $\mathcal{S} \not\supset \mathcal{S}_0$ into true positives and false positives, i.e. $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, where $\mathcal{S}_1 \subset \mathcal{S}_0$ and $\mathcal{S}_2 \cap \mathcal{S}_0 = \emptyset$. We denote with $\widehat{f}^{\mathcal{S}}$ the projection of f_0 onto $\mathcal{F}_\mathcal{S}$, omitting the subscripts $\widehat{\mathcal{T}}$ and $\widehat{\boldsymbol{\beta}}$. With a slight abuse of notation we denote $\mathbb{E}(f, g) = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i) g(\boldsymbol{x}_i)$. Then we can write

$$\|f_0 - \widehat{f}^{\mathcal{S}}\|_n^2 = \|f_0 - \widehat{f}^{\mathcal{S}_1} + \widehat{f}^{\mathcal{S}_1} - \widehat{f}^{\mathcal{S}}\|_n^2 > \|f_0 - \widehat{f}^{\mathcal{S}_1}\|_n^2 - 2 \mid \mathbb{E}[(f_0 - \widehat{f}^{\mathcal{S}_1})(\widehat{f}^{\mathcal{S}} - \widehat{f}^{\mathcal{S}_1})] \mid,$$

where $\mathbb{E}[(f_0 - \widehat{f}^{\mathcal{S}_1})(\widehat{f}^{\mathcal{S}} - \widehat{f}^{\mathcal{S}_1})]$ equals $\rho_n^{\mathcal{S}}$ defined in (2.18). We note that $\delta_n^{\mathcal{S}_1}$ is monotone increasing in the number of false non-discoveries $\mid \mathcal{S}_0 \backslash \mathcal{S}_1 \mid$. The statement of the Lemma then follows from the fact that $\|f_0 - \widehat{f}^{\mathcal{S}}\|_n^2 > \inf_{\mathcal{S}_1 \subset \mathcal{S}_0} \delta_n^{S_1} - 2 \sup_{\mathcal{S} \not\supset \mathcal{S}_0} \rho_n^S > \inf_{i \in \mathcal{S}_0} \delta_n^{S_0 \backslash i} - M\, \varepsilon > M\varepsilon$.

A.1.2 Proof of Theorem 2.5.2

We introduce some more notation. We denote with $\mathcal{F}_\mathcal{S} = \bigcup_{K=1}^n \mathcal{F}_\mathcal{S}(K)$ all valid trees that split on directions inside $\mathcal{S}$ and we write $\Pi_{K,\mathcal{S}}(\cdot)$ for the conditional prior, given K and $\mathcal{S}$.

Similarly as in Section A.1.1, we verify the three conditions (Prior Concentration, En-

tropy, Prior Anti-concentration). The prior model concentration condition is again satisfied automatically from the definition of model weights in (2.20) and $K_{\mathcal{S}_0} = \lfloor C_K / C_\varepsilon \, n \, \varepsilon^2_{n,\mathcal{S}_0} / \log n \rfloor$. Namely,

$$\pi(K_{\mathcal{S}_0}, \mathcal{S}_0) \propto \mathbf{e}^{-C \, \max\{C_K / C_\varepsilon n \, \varepsilon^2_{n,\mathcal{S}_0}, q_0 \log p\}} \geq \mathbf{e}^{-n \, \varepsilon^2_{n,\mathcal{S}_0}}, \tag{A.9}$$

for $C_K < C_\varepsilon / C$, where we used the assumption $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$. Next, the prior concentration in the parameter space associated with the true model

$$\Pi_{K_{\mathcal{S}_0}, \mathcal{S}_0} \left(f_{\mathcal{T}, \boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}_0}(K_{\mathcal{S}_0}) : \| f_{\mathcal{T}, \boldsymbol{\beta}} - f_0 \|_n \leq \varepsilon_{n, \mathcal{S}_0} \right) \geq \mathbf{e}^{-d \, n \, \varepsilon^2_{n, \mathcal{S}_0}}$$

follows again from Section 8.2 of RP17.

For the entropy considerations, we focus only on models with up to q_n covariates and up to K_n splits, where $q_n = \lceil C_q n \, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$ and $K_n = \lceil \bar{C} n \, \varepsilon^2_{n,\mathcal{S}_0} / \log n \rceil$ were defined in Theorem 2.5.2. This restriction is justified by the following Lemma.

Lemma A.1.3. *Denote with $q_n = \lceil C_q n \, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$ and $K_n = \lceil \bar{C} \, n \, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$. Under the assumptions of Theorem 2.5.1, we have*

$$\Pi(q \geq q_n \mid \boldsymbol{Y}^{(n)}) \to 0 \quad \textit{and} \quad \Pi(K \geq K_n \mid \boldsymbol{Y}^{(n)}) \to 0 \tag{A.10}$$

in $\mathbb{P}^{(n)}_{f_0}$-probability as $n \to \infty$.

Proof: It suffices to show that $\Pi(q > q_n) \mathbf{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \to 0$ and $\Pi(K \geq K_n) \mathbf{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \to 0$ for $d > 2$ from (A.2). We have $q_0 \leq q_n$ for n large enough, since $q_0 = \mathcal{O}(1)$ as $n \to \infty$, and thereby

$$\begin{aligned}
&\Pi(q \geq q_n) \mathbf{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \lesssim \mathbf{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=q_n}^{p} \tbinom{p}{q} \sum_{K=1}^{n} \mathbf{e}^{-C \, \max\{K \log n, q \log p\}} \\
&\leq \mathbf{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=q_n}^{p} \mathbf{e}^{\log n + q \log(p \, \mathbf{e}/q) - C \, q \log p} \leq \mathbf{e}^{\log p + \log n - (C-1) \, q_n \log p + (d+2) n \, \varepsilon^2_{n,\mathcal{S}_0}} \\
&\leq \mathbf{e}^{-(C-3) \, q_n \log p + (d+2) n \, \varepsilon^2_{n,\mathcal{S}_0}},
\end{aligned}$$

where we used the fact that for $q_0 \geq 2$ and $\alpha \in (0, 1]$, we have $\log n \leq n^{q_0/(2\alpha+q_0)}$. Since $q_n \log p \geq C_q n \varepsilon^2_{n,\mathcal{S}_0}$, the right hand side above goes to zero when $(C-3)C_q > d+2$. This will be guaranteed with $C > 3$ and C_q large enough. Similarly, we have

$$\begin{aligned}
\Pi(K \geq K_n)\mathbf{e}^{(d+2)n\,\varepsilon^2_{n,\mathcal{S}_0}} &\lesssim \mathbf{e}^{(d+2)n\,\varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=0}^{p} \tbinom{p}{q} \sum_{K=K_n}^{n} \mathbf{e}^{-C\,\max\{K\log n, q\log p\}} \\
&\leq \mathbf{e}^{(d+2)n\,\varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=0}^{p} \sum_{K=K_n}^{n} \mathbf{e}^{-(C-1)\max\{K\log n, q\log p\}} \\
&\leq \mathbf{e}^{\log(p+1)+\log n-(C-1)\,K_n\log n+(d+2)n\,\varepsilon^2_{n,\mathcal{S}_0}} \leq \mathbf{e}^{-(C-2)\,K_n\log n+(d+3)n\,\varepsilon^2_{n,\mathcal{S}_0}},
\end{aligned}$$

where we used our assumption $\log p \leq n^{q_0/(2\alpha+q_0)}$. Since $K_n \geq \bar{C} n\,\varepsilon^2_{n,\mathcal{S}_0}$, the right hand side above goes to zero when $(C-2)\bar{C} > d+3$. Together with the prior mass conditions (A.2) and (A.9), (A.10) follows from Lemma 1 of [86].

This Lemma essentially says that the posterior does not overfit in terms of both q and K, where the mass concentrates on models with $K < K_n$ splits. Note that K_n is of the same order as the optimal regularity $K_{\mathcal{S}_0}$. Now, we denote with $\mathbf{\Gamma}_n \subset \mathbf{\Gamma}$ a sieve consisting of all models with less than q_n variables and K_n splits. For the entropy bounds of overfitting and underfitting models (inside the sieve $\mathbf{\Gamma}_n$), we can use the same arguments as in Section A.1.1. Assume a model $(K, \mathcal{S}) \in \mathbf{\Gamma}_n$. Then it follows from (A.5) that

$$\log N\left(\varepsilon_{n,\mathcal{S}}\,;\ \mathcal{F}_{\mathcal{S}}(K)\,;\ \|\cdot\|_n\right) \leq K\log(3\,B\,n^3 n^{3/2}) \lesssim K_n\log n \lesssim n\,\varepsilon^2_{n,\mathcal{S}_0}.$$

For over-fitting models, this can be further upper-bounded with a multiple of $n\,\varepsilon^2_{n,\mathcal{S}}$, thus satisfying (A.4). The last requirement for the entropy condition is verifying the following variant of (A.6)

$$\sum_{(K,\mathcal{S})\in\mathbf{\Gamma}_n:\,\mathcal{S}\not\supset\mathcal{S}_0\cup K<K_{\mathcal{S}_0}} \mathbf{e}^{-C_2\,M^2\,n\,\varepsilon^2_{n,\mathcal{S}_0}} + \sum_{(K,\mathcal{S})\in\mathbf{\Gamma}_n:\,\mathcal{S}\supset\mathcal{S}_0\cap K\geq K_{\mathcal{S}_0}} \mathbf{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}}} \leq 1 \qquad \text{(A.11)}$$

for some suitable $C_2 > 0$. Since $n\,\varepsilon^2_{n,\mathcal{S}_0} \leq n\,\varepsilon^2_{n,\mathcal{S}}$ for $\mathcal{S} \supset \mathcal{S}_0$, we can upper-bound the left hand side with

$$\sum_{\mathcal{S}:|\mathcal{S}|<q_n} \sum_{K=1}^{K_n} \mathbf{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}_0}} \leq \mathbf{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}_0}} \left(\frac{2\,\mathbf{e}\,p}{q_n}\right)^{q_n+1} \mathbf{e}^{\log K_n} \leq \mathbf{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}_0}+(q_n+1)\log p+\log K_n}. \tag{A.12}$$

Since $q_n \log p \asymp n\,\varepsilon^2_{n,\mathcal{S}_0}$ and $\log K_n \lesssim n^{q_0/(2\alpha+q_0)} \lesssim n\varepsilon^2_{n,\mathcal{S}_0}$, the right-hand side of (A.12) converges to zero for some suitably large C_2 as $n \to \infty$, thus satisfying (A.11).

In place of the anti-concentration condition (similarly as in (A.8)), we need to verify that the prior probability of larger models (that contain the truth) is small in the sense that, for some $H > 0$,

$$\sum_{(K,\mathcal{S})\in\mathbf{\Gamma}_n:\{\mathcal{S}\supset\mathcal{S}_0\}\cap\{K\geq K_{\mathcal{S}_0}\}} \pi(\mathcal{S},K) \leq \mathbf{e}^{-H\,n\,\varepsilon^2_{n,\mathcal{S}_0}}. \tag{A.13}$$

We can write

$$\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|<q_n} \sum_{K=K_{\mathcal{S}_0}}^{K_n} \pi(\mathcal{S},K) \leq \sum_{q=0}^{q_n} \tbinom{p}{q} \sum_{K=K_{\mathcal{S}_0}}^{K_n} \mathbf{e}^{-C\,K_{\mathcal{S}_0}\log n} \tag{A.14}$$

$$\leq \left(\frac{2\,\mathbf{e}\,p}{q_n}\right)^{q_n+1} \mathbf{e}^{\log K_n} \mathbf{e}^{-C\,K_{\mathcal{S}_0}\log n}. \tag{A.15}$$

Because $q_n \log p \asymp n\,\varepsilon^2_{n,\mathcal{S}_0}$ and $\log K_n \lesssim n^{q_0/(2\alpha+q_0)} \lesssim n\varepsilon^2_{n,\mathcal{S}_0}$ the condition (A.13) is satisfied for some $H > 0$ when C and C_K are large enough.

A.1.3 Proof of Theorem 2.5.3

We modify the notation a bit. We adopt the definition of δ-valid ensembles from RP17 (Definition 5.3). With $\mathcal{F}_{\mathcal{S}}(\boldsymbol{K})$ we denote all δ-valid tree ensembles $f_{\mathcal{E},\boldsymbol{B}}$ that (a) are uniformly bounded (i.e. $\|f_{\mathcal{E},\boldsymbol{B}}\|_\infty \leq B$ for some $B > 0$), (b) consist of T trees with $\boldsymbol{K} = (K^1,\ldots,K^T)' \in \mathbb{N}^T$ leaves and (c) that split along directions $\mathcal{S}$.

We start by showing that the prior model concentration condition is satisfied. From our assumption $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$ and definition $K_{\mathcal{S}_0} < C_K/C_\varepsilon^2\, n\,\varepsilon^2_{n,\mathcal{S}_0}/\log n$ and using

(2.23) and (2.24), we obtain

$$\pi(\mathcal{S}_0, E(K_{\mathcal{S}_0})) \propto \sum_{T=1}^{K_{\mathcal{S}_0}} \mathbf{e}^{-C_T\, T} \sum_{\boldsymbol{K}\in\mathbb{N}^T:\sum_{t=1}^T K^t = K_{\mathcal{S}_0}} \mathbf{e}^{-C\, n^{q_0/(2\alpha+q_0)} \log n}$$
$$\geq \mathbf{e}^{-(C_T C_K/(C_\varepsilon\, \log n) + C/C_\varepsilon^2)\, n\, \varepsilon_{n,\mathcal{S}_0}^2}.$$

The right-hand side can be further lower-bounded with $\mathbf{e}^{-n\,\varepsilon_{n,\mathcal{S}_0}^2}$ for a large enough C_ε and n. Next, we need to show prior concentration in the parameter space under the true model equivalence class $(\mathcal{S}_0, E(K_{\mathcal{S}_0}))$. All that is needed is finding a single well-approximating forest supported on one partition ensemble characterized by $(T, \boldsymbol{K})$ from the equivalence class $E(K_{\mathcal{S}_0})$. Such an ensemble can be obtained by considering $T = 1$ and a single k-d tree with $K_{\mathcal{S}_0}$ leaves from Lemma 3.2 of RP17. The prior concentration condition then boils down to (A.2), which has already been verified in RP17.

Next, we show that for $K_n = \lceil \bar{C}\, n\, \varepsilon_{n,\mathcal{S}_0}^2 / \log n \rceil$ we have

$$\Pi\left((T, \boldsymbol{K}) : \sum_{t=1}^T K^t \geq K_n \mid \boldsymbol{Y}^{(n)}\right) \to 0.$$

We can write

$$\Pi\left((T, \boldsymbol{K}) : \sum_{t=1}^T K^t \geq K_n\right) \lesssim \sum_{T=1}^n \mathbf{e}^{-C_T\, T} \sum_{q=0}^p \tbinom{p}{q} \sum_{Z=K_n}^n \sum_{\boldsymbol{K}:\sum_{t=1}^T K^t = Z} \mathbf{e}^{-C \max\{Z\log n, q\log p\}}$$
$$\lesssim \mathbf{e}^{-(C-1)K_n \log n + \log p + 2\log n + \log p(n) - C_T},$$

where $p(n)$ is the partitioning number. According to Andrews (1976), we have

$$\log p(n) \sim \pi\sqrt{\frac{2\,n}{3}} \quad \text{as} \quad n \to \infty. \tag{A.16}$$

Under our assumptions $q_0 > 2$ and $\alpha \in (0, 1]$, we have $\sqrt{n} \leq n^{q_0/(2\alpha+q_0)}$ and $\log n \leq$

$n^{q_0/(2\alpha+q_0)}$. From $\log p \leq n^{q_0/(2\alpha+q_0)}$ and using the fact that $K_n \geq \bar{C}\, n\, \varepsilon_{n,\mathcal{S}_0}^2 / \log n$, we can then write

$$\Pi\left((T, \boldsymbol{K}) : \sum_{t=1}^{T} K^t \geq K_n\right) \mathbf{e}^{(d+2)n\,\varepsilon_{n,\mathcal{S}_0}^2} \lesssim \mathbf{e}^{-[(C-1)\bar{C} - D\,\pi\sqrt{2/3} - d - 5]n\,\varepsilon_{n,\mathcal{S}_0}^2}$$

for some $D > 0$. The right hand side goes to zero for $C > 1$ and $\bar{C}$ large enough. Similarly, we can show that $\Pi(q \geq q_n \mid \boldsymbol{Y}^{(n)}) \to 0$ as $n \to \infty$ for $q_n = \lceil C_q\, n\, \varepsilon_{n,\mathcal{S}_0}^2 / \log p \rceil$ by proceeding as in Lemma A.1.3 in Section A.1.2.

Based on the previous paragraph, we narrow down attention to a subset of model indices $\boldsymbol{\Gamma}_n \subset \boldsymbol{\Gamma}$, consisting of models $(\mathcal{S}, E(Z))$ such that $\mid \mathcal{S} \mid < q_n$ and $Z < K_n$. We now define a sieve $\mathcal{F}_n$ as follows

$$\mathcal{F}_n = \bigcup_{q=0}^{q_n} \bigcup_{T=1}^{K_n} \bigcup_{\sum_{t=1}^{T} K^t \leq K_n} \bigcup_{\mathcal{S}:|\mathcal{S}|=q} \mathcal{F}_{\mathcal{S}}(\boldsymbol{K}).$$

It follows from the previous paragraph that $\Pi(\mathcal{F}_n^c \mid \boldsymbol{Y}^{(n)}) \to 0$ as $n \to \infty$. For the entropy calculation we thus focus on the sieve $\mathcal{F}_n$.

We first note that the metric entropy $\log N\left(\varepsilon_{n,\mathcal{S}};\ \mathcal{F}(\mathcal{E});\ \|\cdot\|_n\right)$, where $\mathcal{F}(\mathcal{E})$ are all uniformly bounded forests supported on a δ-valid partition ensemble $\mathcal{E}$, can be upper-bounded with $\left(\sum_{t=1}^{T} K^t\right) \log(B/\varepsilon_{n,\mathcal{S}} C_1 \kappa(\mathcal{E})\sqrt{n})$ (follows from equation (9.3) of RP17), where $\kappa(\mathcal{E})$ is the condition number of a valid ensemble (defined in Section 9.1. of RP17). Next, we find an upper bound for the covering number of the tree ensembles that are attached to a model $(\mathcal{S}, E(Z))$, where $E(Z)$ is the equivalence class of $(T, \boldsymbol{K})$ defined in (2.22). From Section 9.1 of RP17, and using the fact that $\Delta(E(Z)) \leq Z! p(Z)$, it follows that

$$\begin{aligned}
&\log N\left(\varepsilon_{n,\mathcal{S}};\ \bigcup_{(T,\boldsymbol{K})\in E(Z)} \mathcal{F}_{\mathcal{S}}(\boldsymbol{K}) \cap \mathcal{F}_n;\ \|\cdot\|_n\right) \\
&\quad \leq \log \Delta(E(Z)) + \log \Delta(\mathcal{V}\mathcal{E}_{\mathcal{S}}^{\boldsymbol{K}}) + Z \log(B/\varepsilon_{n,\mathcal{S}} C_1 \kappa(\mathcal{E})\sqrt{n}) \\
&\quad \lesssim Z \log Z + \sqrt{Z} + Z \log(\mid \mathcal{S} \mid n^2) + Z \log\left(n^{2+\delta/2}\sqrt{Z}\right)
\end{aligned}$$

for some $C_1 > 0$, where $\Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}})$ is the cardinality of δ-valid ensembles $\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}$. Inside the sieve, we have $|\,\mathcal{S}\,| < q_n \leq n$ and $Z < K_n \asymp n\,\varepsilon_{n,\mathcal{S}_0}^2 / \log n$ and thereby we can upper bound the log entropy with a constant multiple of $n\,\varepsilon_{n,\mathcal{S}_0}^2$. For an overfitting model $(\mathcal{S}, E(Z))$ such that $Z \geq K(\mathcal{S}_0)$ and $\mathcal{S} \supset \mathcal{S}_0$, the log-covering number is further upper-bounded with $n\,\varepsilon_{n,\mathcal{S}}^2 \geq n\,\varepsilon_{n,\mathcal{S}_0}^2$. Next, we verify the following variant of condition (A.6)

$$\sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{Z < K_{\mathcal{S}_0}\}}} \mathbf{e}^{-C_2\, M^2\, n\, \varepsilon_{n,\mathcal{S}_0}^2} + \sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{Z \geq K_{\mathcal{S}_0}\}}} \mathbf{e}^{-C_2 n\, \varepsilon_{n,\mathcal{S}}^2} \leq 1 \tag{A.17}$$

for some $C_2 > 0$. Since $n\,\varepsilon_{n,\mathcal{S}}^2 > n\,\varepsilon_{n,\mathcal{S}_0}^2$ for $\mathcal{S} \supset \mathcal{S}_0$ and $M > 1$, we can upper-bound the left-hand-side with

$$\mathbf{e}^{-C_2\, n\, \varepsilon_{n,\mathcal{S}_0}^2} \sum_{q=0}^{q_n} \tbinom{p}{q} \sum_{Z=1}^{K_n} \Delta(E(Z)) \lesssim$$
$$\left(\frac{2\,\mathbf{e}\,p}{q_n}\right)^{q_n+1} \mathbf{e}^{-C_2\, n\, \varepsilon_{n,\mathcal{S}_0}^2 + \log q_n + \log K_n + K_n \log K_n + \pi\sqrt{2K_n/3}},$$

where we used the fact $\Delta(E(Z)) \leq Z! p(Z)$ and (A.16). Since $K_n \log K_n \lesssim n\,\varepsilon_{n,\mathcal{S}_0}^2$ and $q_n \log p \asymp n\varepsilon_{n,\mathcal{S}_0}^2$, the right hand side goes to zero for a large enough constant $C_2 > 0$.

Lastly, the anti-concentration condition is replaced with

$$\sum_{T=K_n}^{n} \pi(T) \sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{Z \geq K_{\mathcal{S}_0}\}}} \sum_{\boldsymbol{K} \in \mathbb{N}^T : \sum_{t=1}^T K^t = Z} \pi(\mathcal{S}, \boldsymbol{K} \mid T) \leq \mathbf{e}^{-H\, n\, \varepsilon_{n,\mathcal{S}_0}^2}$$

for some $H > 0$. Using the fact $\pi(\mathcal{S}, \boldsymbol{K} \mid T) \gtrsim \mathbf{e}^{-C \sum K^t \log n}$, we can upper-bound the left hand side above with

$$\sum_{T=1}^{K_n} \pi(T) \mathbf{e}^{-C\, K_{\mathcal{S}_0} \log n} \sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{Z \geq K_{\mathcal{S}_0}\}}} \Delta(E(Z))$$
$$\lesssim \mathbf{e}^{-C\, K_{\mathcal{S}_0} \log n} \left(\frac{2\,\mathbf{e}\,p}{q_n}\right)^{q_n+1} \mathbf{e}^{2\log K_n + K_n \log K_n + \pi\sqrt{2K_n/3} - C_T}$$

Using similar arguments as before, and because $K_{\mathcal{S}_0} \log n \geq C_K/C_\varepsilon \, n \, \varepsilon^2_{n,\mathcal{S}_0}$, the condition will be satisfied for large enough $C > 0$ and $C_K > 0$.

A.1.4 Theory for ABC

First, we show the following ABC posterior concentration result.

Theorem A.1.1. *Under the assumptions of Theorem 4.1 and assuming $\sigma^2 = 1/n$ in (1), the naive ABC posterior satisfies with $\mathbb{P}^{(n)}_{f_0}$ tending to one*

$$\Pi\left[\|f - f_0\|_n > \lambda_n \mid \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] \lesssim 1/M$$

for $\epsilon_n^T = \sqrt{2 \log n/n}$, $\lambda_n = 4\epsilon_n^T/3 + 1/\sqrt{n}$ and for any $M > 0$ large enough.

Proof: We will be working conditionally on the event $\mathcal{A} = \{\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)' : \max_{1 \leq i \leq n} \mid \varepsilon_i \mid \leq \sqrt{2 \log n/n}\}$ whose complement has a small probability, i.e. $\mathbb{P}^{(n)}_{f_0}[\mathcal{A}^c] \leq c_0/\sqrt{2 \log n}$ for some $c_0 > 0$ when $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, 1/n)$. On the event $\mathcal{A}$, we have

$$\|\boldsymbol{Y} - f_0\|_n = \sqrt{\frac{1}{n}\sum_{i=1}^n \varepsilon_i^2} \leq \sqrt{2 \log n/n} \equiv \epsilon_n^T.$$

We now define a joint event

$$\mathcal{A}(\epsilon_n^T, \lambda_n) \equiv \{(\boldsymbol{Y}^\star, f) : \|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n \leq \epsilon_n^T \quad \text{and} \quad \|f - f_0\|_n > \lambda_n\}.$$

For all $(\boldsymbol{Y}^\star, f) \in \mathcal{A}(\epsilon_n^T, \lambda_n)$ we have

$$\|f - f_0\|_n \leq \|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n + \|f - \boldsymbol{Y}^\star\|_n + \|f_0 - \boldsymbol{Y}\|_n \leq \frac{4}{3}\epsilon_n^T + \|f - \boldsymbol{Y}^\star\|_n.$$

This means that $(\boldsymbol{Y}^\star, f) \in \mathcal{A}(\epsilon_n^T, \lambda_n)$ implies $\|f - \boldsymbol{Y}^\star\|_n > \lambda_n - \frac{4}{3}\epsilon_n^T$ and choosing $\lambda_n \geq \frac{4}{3}\epsilon_n^T + t_\varepsilon$ leads to

$$\Pi[\mathcal{A}(\epsilon_n^T, \lambda_n)] \leq \int \mathbb{P}_f[\|f - \boldsymbol{Y}^\star\|_n > t_\varepsilon] d\Pi(f)$$

and

$$\Pi\left[\|f - f_0\|_n > \frac{4}{3}\epsilon_n^T + t_\varepsilon \;\middle|\; \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] \leq \frac{\int \mathbb{P}_f[\|\boldsymbol{Y}^\star - f\|_n > t_\varepsilon] d\Pi(f)}{\int \mathbb{P}_f[\|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n \leq \epsilon_n^T] d\Pi(f)}. \quad \text{(A.18)}$$

Now, we have for a random variable χ_n^2 with a chi-square distribution with n degrees of freedom

$$\mathbb{P}_f[\|\boldsymbol{Y}^\star - f\|_n > u] = \mathbb{P}_f\left[\frac{\chi_n^2}{n^2} > u^2\right] = \mathbb{P}_f\left[\mathbf{e}^{\chi_n^2/4} > \mathbf{e}^{u^2\, n^2/4}\right] \leq \frac{2^{n/2}}{\mathbf{e}^{u^2\, n^2/4}}.$$

Next, for n large enough we can write

$$\begin{aligned}
\int \mathbb{P}_f[\|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n \leq \epsilon_n^T] d\Pi(f) &\geq \int_{\|f - f_0\|_n \leq \epsilon_n^T/3} \mathbb{P}_f[\|\boldsymbol{Y}^\star - f\|_n \leq \epsilon_n^T/3] d\Pi(f) && \text{(A.19)} \\
&\geq \Pi[\|f - f_0\|_n \leq \epsilon_n^T/3] - \mathbf{e}^{n/2 \log 2 - n \log n/18} && \text{(A.20)} \\
&\geq \Pi[\|f - f_0\|_n \leq \epsilon_n^T/3]/2. && \text{(A.21)}
\end{aligned}$$

Next (under the assumption $q_0 \log p < n^{q_0/(2\alpha+q_0)}$, we have $\pi(\mathcal{S}_0) \geq \mathbf{e}^{-n\varepsilon_{n,\mathcal{S}_0}^2}$ and (assuming $K = K_{\mathcal{S}_0} \asymp n\varepsilon_{n,\mathcal{S}_0}^2 / \log n$ and denoting $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^K$ the steps of the $\|\cdot\|_n$ projection of f_0 onto trees with K leafs) for some $c > 0$

$$\begin{aligned}
\Pi[\|f - f_0\|_n \leq \epsilon_n^T/3] &> \mathbf{e}^{-n\,\varepsilon_{n,\mathcal{S}_0}^2} \Pi(\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|_2 \leq \epsilon_n^T/6) && \text{(A.22)} \\
&> \mathbf{e}^{-n\,\varepsilon_{n,\mathcal{S}_0}^2} \frac{\mathbf{e}^{-K\log 2 - \|\widehat{\boldsymbol{\beta}}\|_2^2 - (\varepsilon_n^T)^2/72 + K/2 \log[(\varepsilon_n^T)^2/36]}}{\Gamma(K/2) K/2} > \mathbf{e}^{-c\,n\,\varepsilon_{n,\mathcal{S}_0}^2}. && \text{(A.23)}
\end{aligned}$$

We can now upper-bound (A.18) with $2^{n/2}\mathbf{e}^{-t_\varepsilon^2 n^2/4 + cn\,\varepsilon_{n,\mathcal{S}_0}^2}$ which is smaller than an arbitrary constant $M > 0$ for n large enough if we choose $t_\varepsilon = 1/\sqrt{n}$.

Given this consistency result, we can immediately conclude (using the inequality in (21) in the paper) that the ABC posterior will not reward underfitting model as long as our identifiability and irrepresentability conditions are satisfied with $\varepsilon = \lambda_n$. In other words, under the assumptions of Theorem A.1.1 and assuming that $\mathcal{S}_0$ is (f_0, λ_n)-identifiable and that λ_n-irrepresentability holds we have, with $\mathbb{P}_{f_0}$ tending to one and for any $M > 0$,

$$\Pi\left[\mathcal{S} \not\supset \mathcal{S}_0 \mid \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] \lesssim 1/M.$$

Regarding over-fitting models, we first show the following ABC analogue of Lemma 8.1. We can write, on the event $\mathcal{A}$, and for $q_n = C_q \lceil n\,\varepsilon_{n,\mathcal{S}_0}^2 / \log p \rceil$ (as in Lemma 8.1)

$$\Pi_1 \equiv \Pi\left[q \geq q_n \mid \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] = \sum_{\mathcal{S}:|\mathcal{S}|\geq q_n} \pi(\mathcal{S}) \frac{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f \mid \mathcal{S})}{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f)}.$$

It turns out from the proof of Theorem A.1.1 that

$$\Pi_1 \leq \frac{\sum_{q \geq q_n} \sum_{\mathcal{S}:|\mathcal{S}|=q} \pi(\mathcal{S})}{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f)} \lesssim \mathbf{e}^{c\,n\varepsilon_{n,\mathcal{S}_0}^2} \Pi(q \geq q_n).$$

In the proof of Lemma 1.1 we have already showed (under the assumptions of Theorem 4.1) that $\Pi(q \geq q_n) \lesssim \mathbf{e}^{-n\,\varepsilon_{n,\mathcal{S}_0}^2 C}$ for some $C > 0$. Choosing C_q large enough, one concludes that $\Pi_1 \to 0$ as $n \to \infty$. This shows that the ABC posterior concentrates on the sieve of models $\mathcal{F}_n$ with up to q_n covariates. Using this result, we can focus on models of size up to q_n and show that the posterior probability of over-fitting models goes to zero. Indeed, on the event

$\mathcal{A}$ and on $\mathcal{F}_n$ we have (using an inequalities (4) and (6))

$$\Pi\left[\{\mathcal{S}\supset\mathcal{S}_0\}\cap\mathcal{F}_n \mid \|\boldsymbol{Y}-\boldsymbol{Y}^\star\|_n\leq\epsilon_n^T\right] \leq \frac{\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n}\pi(\mathcal{S})}{\int\mathbb{P}_f[\|\boldsymbol{Y}-\boldsymbol{Y}^\star\|_n\leq\epsilon_n^T]d\Pi(f)}$$
$$\lesssim \mathbf{e}^{(c-C)\,n\varepsilon^2_{n,\mathcal{S}_0}}\left(\frac{2\mathbf{e}p}{q_n}\right)^{q_n+1} \lesssim \mathbf{e}^{-H\,n\varepsilon^2_{n,\mathcal{S}_0}}$$

for some $H>0$ with $C>0$ is large enough. This concludes that the ABC posterior will lead to consistent variable selection as well.

We now discuss how the theory can be extended when data-splitting is deployed in ABC. First, we discuss the case when the split is done only once before applying ABC (not internally at each iteration). Denote with n_1 the training sample size and with n_2 the validation sample size. In order for the consistency result in Theorem A.1.1 to hold, we need to make sure that prior concentration holds in the sense that $\Pi[\|f-f_0\|_{n_2}\lesssim\epsilon_{n_2}^T]\geq\mathbf{e}^{-c\,n_2\epsilon^2_{n_2,\mathcal{S}_0}}$ for some $c>0$. Leaving n_1 data-points for training the prior, we know (from results in RP17 under fixed σ^2) that the posterior concentrates at the optimal rate (up to a log factor), i.e.

$$\Pi[|\ f-f_0 mid_{n_1}\lesssim\epsilon_{n_1,\mathcal{S}_0}\mid\boldsymbol{Y}_{\mathcal{I}}^{(n)},\mathcal{S}_0]\to 1\quad\text{as } n_1\to\infty.$$

Choosing n_1 and n_2 in such a way so that $\epsilon_{n_1,\mathcal{S}_0}\lesssim\epsilon_{n_2}^T\equiv\sqrt{2(\log n_2)/n_2}$ (and assuming that observed fixed covariates in the training and testing sets are close), the prior concentration condition will be satisfied and the ABC will be consistent and concentrate at the rate λ_{n_2}. This implies variable selection consistency of our ABC method under identifiability and irrepresentabilty conditions which depend on λ_{n_2}. A similar conclusion is obtained for the expected posterior prior (2.9) where

$$\Pi[\|f-f_0\|_{n_1}\lesssim\epsilon_{n_1,\mathcal{S}_0}]\geq\pi(\mathcal{S}_0)\frac{1}{L}\sum_l\Pi[\|f-f_0\|_{n_1}\lesssim\epsilon_{n_1,\mathcal{S}_0}\mid\boldsymbol{Y}_{\mathcal{I}_l}^{(n)},\mathcal{S}_0]\gtrsim\pi(\mathcal{S}_0).$$

A rigorous proof of ABC consistency for the expected posterior priors would require more

Table A.1: Computation time of 1 000 MCMC iterations of BART/DART in seconds (using the Friedman's datasets with $\sigma = 5$ and autocorrelation of 0.9).

		BART				DART			
		$T = 10$	$T = 20$	$T = 50$	$T = 200$	$T = 10$	$T = 20$	$T = 50$	$T = 200$
$n = 100$	$p = 100$	0.21	0.32	0.54	1.86	0.55	0.64	0.76	2.27
	$p = 1\,000$	0.53	0.67	1.57	5.48	0.96	0.98	1.91	5.79
	$p = 10\,000$	3.56	5.58	10.91	39.16	5.93	7.72	12.55	36.95
$n = 250$	$p = 100$	0.21	0.34	0.79	2.99	0.41	0.56	0.99	3.19
	$p = 1\,000$	0.50	0.82	1.81	6.51	0.87	1.14	1.83	6.58
	$p = 10\,000$	3.70	5.63	11.38	40.48	6.29	8.06	12.97	39.13
$n = 500$	$p = 100$	0.29	0.53	1.23	4.93	0.49	0.71	1.40	5.07
	$p = 1\,000$	0.63	1.11	2.36	8.65	1.01	1.30	2.29	7.89
	$p = 10\,000$	4.21	6.54	12.27	43.35	6.80	8.35	13.46	37.81
$n = 1\,000$	$p = 100$	0.53	0.97	2.22	8.86	0.71	1.13	2.30	8.95
	$p = 1\,000$	0.91	1.49	3.32	12.82	1.24	1.84	3.54	12.12
	$p = 10\,000$	5.41	8.18	14.23	48.92	7.61	9.06	14.47	41.90
$n = 10\,000$	$p = 100$	7.17	10.78	23.25	82.45	6.37	12.07	22.33	92.23
	$p = 1\,000$	13.00	22.12	34.67	125.98	12.76	16.95	40.25	102.72
	$p = 10\,000$	25.35	31.39	59.71	218.08	28.59	39.93	73.99	171.73

care and will be left for future investigation.

A.2 ABC Computational Feasibility

Regarding computational considerations, our sampling method deploys MCMC inside each ABC iteration but uses only on a subset of the original observations) (say $\frac{n}{2}$ observations) and a subset of $\mid \mathcal{S} \mid < p$ variables. In addition, we only need to collect one posterior sample after a burnin period B.

In order to understand how ABC scales with $|\mathcal{S}|$, p and s, we first assess the computing time of plain BART/DART. The timing comparisons are summarized in Table A.1. From these computations we can conclude, for example, that running $M = 1\,000$ BART iterations with $T = 200$ trees (the default) on a dataset with $p = 10\,000$ variables and $n = 500$ observations takes 43.35 seconds which roughly amounts to running $43.35 \times 5/0.5 = 433.5$ ABC iterations with $B = 200$ burnin MCMC iterations, $T = 10$ trees and with $s = n/2$, assuming that the sparsity prior is such that $\mid \mathcal{S} \mid \approx 1\,000$. Under the same settings but a stricter sparsity prior such that $\mid \mathcal{S} \mid \approx 100$, we obtain $43.35 \times 5/0.21 = 1032.14$ ABC iterations for the same time as 1 000 BART iterations. These computing times, however, do not take into account autocorrelation in BART samples, where $M = 1\,000$ BART MCMC

iterations do not necessarily yield 1 000 *effective* samples. One advantage of ABC sampling over MCMC is that it is embarrassingly parallel and that it does not incur correlation. This provides an opportunity for large speedups using parallel computing.

A.3 Spike-and-Forests: MCMC Variant

As a precursor to ABC Bayesian Forests, we first implemented an MCMC algorithm for joint sampling from a posterior $\Pi(\mathcal{S}, \mathcal{E} \mid \boldsymbol{Y}^{(n)})$ over the space of models and tree ensemble partitions. We refer to this algorithm as Spike-and-Forests. The sampling follows a Metropolis-Hasting scheme, exploiting the additive structure of forests by sampling each tree individually from conditionals in a Gibbs manner within each Metropolis step (Bayesian backfitting by [48]). The key is assigning a joint proposal distribution $pr(\mathcal{S}, \mathcal{E} \mid \mathcal{S}_m, \mathcal{E}_m) = pr(\mathcal{S} \mid \mathcal{S}_m) pr(\mathcal{E} \mid \mathcal{S}, \mathcal{E}_m)$ over variable subsets $\mathcal{S}$ and partition ensembles $\mathcal{E}$, where $\mathcal{S}_m$ and $\mathcal{E}_m$ are current MCMC states.

We explain the proposal mechanism using a single tree and write $\mathcal{T}$ instead of $\mathcal{E}$. First, a model proposal $\mathcal{S}^\star$ is sampled from $pr(\mathcal{S} \mid \mathcal{S}_m)$ which consists of the following three options: `add`, `delete` and `stay` for adding/deleting one (or none) of the variables. These three steps are chosen with probabilities $0.4, 0.4$ and 0.2, respectively. Candidate variables for deletion/addition are chosen from a uniform distribution. Given the newly suggested model $\mathcal{S}^\star$, the proposal distribution $pr(\mathcal{T} \mid \mathcal{S}^\star, \mathcal{T}_m)$ consists of various moves, described below, depending on the status of $\mathcal{S}^\star$.

If $\mathcal{S}^\star$ was obtained from $\mathcal{S}_m$ by `adding` a variable, the proposal $pr(\mathcal{T} \mid \mathcal{S}^\star = \texttt{add}, \mathcal{T}_m)$ consists of two steps: `birth` and `replace`. In the `birth` step, a bottom node is added to $\mathcal{T}_m$ and in the `replace` step one of the variables that occurs more than once inside $\mathcal{T}_m$ is replaced with the new variable. The birth step increases the size of the tree, while the replace

step does not. The two steps are chosen with probabilities

$$\pi_{\text{birth,add}} = 0.7 \min\left\{\frac{\pi(K+1)}{\pi(K)}, 1\right\}, \pi_{\text{birth, replace}} = 1 - \pi_{\text{birth,add}},$$

where K is the number of bottom nodes in $\mathcal{T}_m$ and $\pi(K)$ is a prior on the number of bottom nodes. If no variable appears more than once in the tree, then `replace` is invalid and $\pi_{\text{birth, replace}}$ is set to 0.

If $\mathcal{S}^\star$ is obtained from $\mathcal{S}_m$ by `deleting` a variable, the proposal $pr(\mathcal{T} \mid \mathcal{S}^\star = \texttt{delete}, \mathcal{T}_m)$ consists of two steps: `death` and `replace`. If the variable chosen for deletion occurs in a bottom node, it can be removed from a tree $\mathcal{T}_m$ with a `delete` step that erases the bottom node. If the variable occurs inside the tree, it can be deleted by replacing it with other variables in the `replace` step. If both of these moves are eligible, we pick one of them with probabilities

$$\pi_{\text{death,delete}} = 0.7 \min\left\{\frac{\pi(K-1)}{\pi(K)}, 1\right\}, \pi_{\text{death,replace}} = 1 - \pi_{\text{death,delete}}.$$

If the variable suggested for deletion is not in a bottom node, then $\pi_{\text{death,delete}} = 0$.

If the pool of variables stays the same, i.e. $\mathcal{S}^\star = \mathcal{S}_m$, the proposal $pr(\mathcal{T} \mid \mathcal{S}^\star = \texttt{stay}, \mathcal{T}_m)$ consists of 4 moves: `add, delete, replace` and `rule`. All proposal moves, and their probabilities, are adopted from Bayesian CART of [56]. These steps only modify the tree configuration without adding/deleting variables.

Regarding the prior distributions for our MCMC implementation, we assume the beta-binomial prior on the variable subsets. Namely, for binary indicators $\gamma_j \in \{0, 1\}$, for whether or not x_j is active, we assume $\mathbb{P}(\gamma_j = 1 \mid \theta) = \theta$ and $\theta \sim \mathcal{B}(a, b)$. The prior distribution on trees consists of (a) the truncated Poisson distribution on the number of bottom leaves, (b) uniform prior over trees with the same number of leaves and (c) standard Gaussian prior on the step sizes. This is the Bayesian CART prior proposed by [56] and analyzed theoretically by [202]. In the computation of MH acceptance ratios, we leverage the fact that the bottom

leave parameters can be integrated out to obtain a conditional marginal likelihood, given each partition.

The MCMC sampling routine can be extended to spike-and-forests, altering each tree inside the forests one by one through Bayesian backfitting [48]. One big advantage of the Bayesian forest representation is that it accelerates mixing since most trees are shallow and thereby more easily modified throughout MCMC (see [186]).

A.4 Sensitivity Analysis

Our sensitivity analysis focuses on two aspects. First, we want to assess how the choices of M (the number of ABC samples), T (the number of trees in each forest), B (the number of burn-in iterations inside each ABC iteration) and ϵ (tolerance for ABC acceptance) collaboratively impact ABC variable importance. Second, we want to investigate the impact of different data splitting strategies, including varying choices of s (proportion of data used in training) and pre-determined data splitting versus internal data splitting. There is an obvious tradeoff between s and M, where small s will yield fewer ABC pseudo-observations that are compatible with the observed data and M will thereby have to be larger. We have considered the following combinations

$$M \in \{1\,000, 10\,000\} \times T \in \{10, 25, 50\} \times B \in \{200, 1\,000\} \times \epsilon \in \{top\,1\%, 5\%, 10\%\}$$

These comparisons are conducted using the Friedman's simulation setup with $p \in \{100, 1\,000\}$, $\rho = 0.9$(autoregressive) and $\sigma = 5$, assuming $s = n/2$ and internal splitting for ABC. We also include various sample sizes $n \in \{100, 500, 1\,000\}$ for each p. For each setting, we show ABC inclusion probabilities (`ip`) for the first 30 variables of which only the first 5 are active (Figure A.1). We denote the parameters for each ABC setup by $T \star B$ where, for example, $20 \star 200$ means each forest consists of $T = 20$ trees and uses $B = 200$ MCMC iterations as a burnin.

Figure A.1: ABC inclusion probabilities of the first 30 variables over different ϵ. Each panel corresponds to a different combination of $p \in \{100, 1\,000\}$ and $M \in \{1\,000, 10\,000\}$. Each row indicates a different model averaging strategy based on a different ϵ value. Each column corresponds to a different sample size. The legend represents various combinations of $T \star B$. For example, $20 * 200$ means each forest consists of $T = 20$ trees and $B = 200$ MCMC iterations as burnin. Note that we use $s = n/2$ here.

From the figures we can see that ABC is more sensitive to the choice of T than to the choice of B. This is not entirely unexpected. As suggested in [48] and [21], a large value of T allows for increased flexibility in fitting the model while smaller T should be adopted for the purpose of variable selection. The variables must compete with each other to be included when T is small. In terms of a median probability model, the model tends to have more

Figure A.2: ABC inclusion probabilities of the first 30 variables over different s. Each panel corresponds to a different combination of $p \in \{100, 1\,000\}$ and $M \in \{1\,000, 10\,000\}$. Each row indicates a different model averaging strategy based on a different ratio of s over n. Each column corresponds to a different sample size. The legend represents various combinations of $T \star B$. For example, $20 * 200$ means each forest consists of $T = 20$ trees and $B = 200$ MCMC iterations as burnin. Note that we use ϵ={top 10%} here.

power and higher false discoveries when T is large, and less power and fewer false discoveries when T is small.

Regarding ϵ, although the trends are similar for top 1%, 5% and 10% selected model, higher variance is observed for smaller tolerance when M is not large enough, especially for $M = 1\,000$ with top 1% models accepted. This is, again, not entirely unexpected.

The comparisons in Figure A.1 were done assuming $s = n/2$. We now consider a similar simulation study, but for $\epsilon = \{\text{top } 10\%\}$ and various s by considering

$$M \in \{1\,000, 10\,000\} \times T \in \{10, 25, 50\} \times B \in \{200, 1\,000\} \times s \in \{n/5, n/2, 4n/5\}.$$

The results are displayed in Figure A.2. The posterior inclusion probabilities do not seem to vary much with respect to s. This suggests that even $s = 0.2n$ provides reasonable prior guesses for ABC regarding variable selection. Based on this sensitivity analysis, we choose $T = 20, B = 200, M = 1\,000, s = n/2, \epsilon = \{\text{top } 10\%\}$ as the default parameters for our ABC model.

The last part of the sensitivity analysis we want to investigate the differences between pre-determined data splitting and internal data splitting. Customarily [17], the subsample size s is chosen as the minimal number of samples needed to convert an improper prior into a proper one. Our situation, however, is different in at least three aspects: (a) we are converting a proper uninformative prior into an informative one, (b) our model is entirely non-parametric and (c) we aim to enhance ABC acceptance rate rather than using non-informative priors for model selection with Bayes factors. As pointed out in [17], defining any optimal training sample is very challenging and one needs to exercise statistical judgment to select from among various strategies. While [17] argue that: "Judgments involved in choosing good training samples will typically be much less than the judgments needed to implement an actual subjective Bayesian analysis", we argue that entertaining some reasonable form of the data splitting (even if not optimal) will provide better results than naive ABC strategy in our context. The following simulated example shows that the variable selection performance with internal splitting is at least as good as with pre-determined splitting. We still use the Friedman's dataset with $n = 500$, $p = 100$ and $p = 1000$, $\sigma = 5$ and autocorrelation 0.9. The ABC settings are $T = 20, \theta = 0.5, s = 0.5n, \epsilon = \text{top } 10\%$. The inclusion probabilities are averaged over 10 datasets and plotted in Figure A.3. From Figure A.3, we can see that

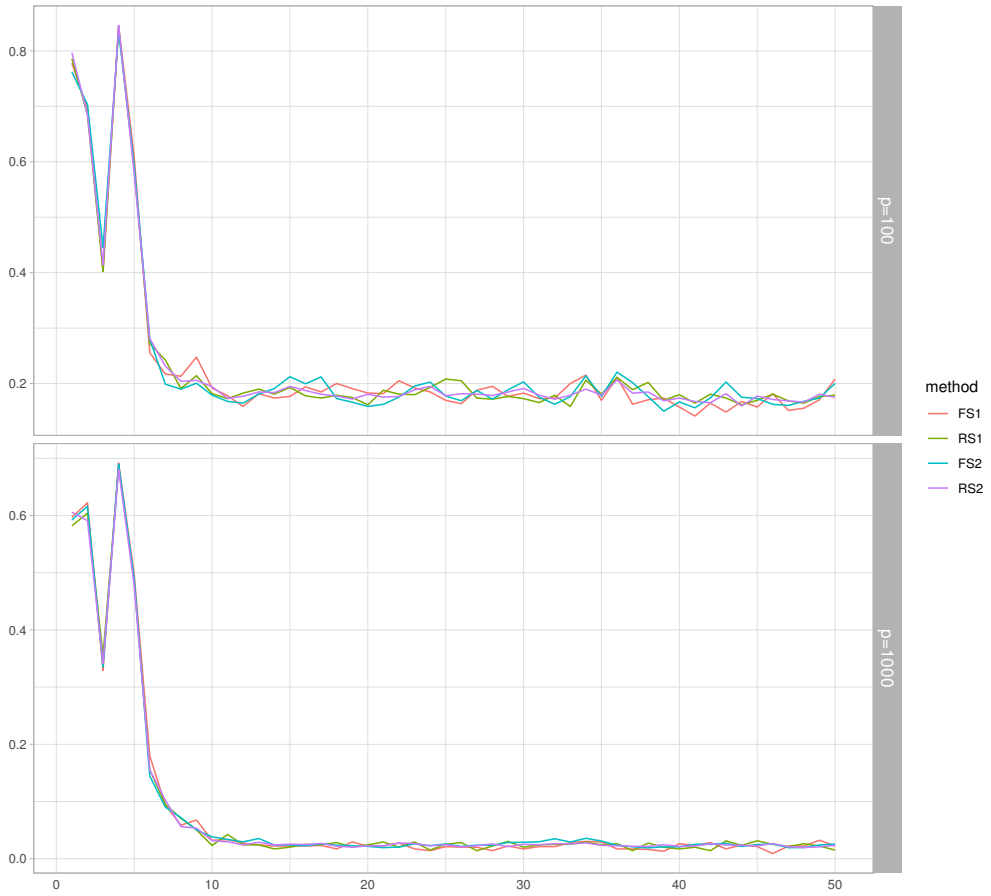


Figure A.3: Comparison of Inclusion Probabilities of Pre-determined Splitting (FS) and Internal Splitting (RS). The inclusion probabilities are averaged over 10 independent Friedman's datasets ($n = 500, \sigma = 5$, autocorrelation $= 0.9$). FS1/RS1 are built with $M = 1\,000$, and FS2/RS2 are built with $M = 10\,000$

the differences in inclusion probabilities of pre-determined splitting and internal splitting are small. This could be explained by the fact the data have been generated with Gaussian noise without any outliers which could potentially affect quality of splits.

Combining our findings from all of the sensitivity analyses above, we recommend the following default settings for the parameters: $s = 0.5n, T = 20, \text{burnin} = 200, \epsilon = \text{top } 10\%$, $M = 1,000$ and internal data splitting.

Table A.2: Basic summary statistics of the HIV dataset. DS refers to the decrease in susceptibility of the drug once the mutations has occurred.

HIV Virus Life Cycle	Drug Class	Mean Log DS	Number of Features	Number of Samples
PI	APV	0.75	201	767
	ATV	1.59	147	328
	IDV	1.33	206	825
	LPV	1.74	184	515
	NFV	2.00	207	842
	RTV	1.72	205	793
	SQV	1.22	206	824
NRTI	X3TC	3.10	283	629
	ABC	1.14	283	623
	AZT	1.55	283	626
	D4T	0.43	281	625
	DDI	0.43	283	628
	TDF	0.22	215	351
NNRTI	DLV	0.98	305	730
	EFV	1.08	312	732
	NVP	1.80	313	744

A.5 Full HIV Data Analysis

In this section, we provide a summary of our results on the entire dataset from [8]. The summary statistics of the data are reported in Table A.2. Comparisons are made between ABC Bayesian Forests, BART, DART and Random Forests. BART and DART are run with 50 trees for 20 000 MCMC iterations (taking the first 10 000 as a burn-in). Random Forests are implemented with the default number of 500 trees.

To summarize the results, we adopted 2 cutoff selection criteria. The first selection threshold is adaptive and is chosen as the maximum importance measure of a non-experimentally validated mutation. This cutoff point corresponds to zero false discoveries. Next, we use an automatic criterion for each method. For ABC Bayesian Forests (run with $T = 20$ trees and $M = 200$ burnin iterations, 10 000 ABC samples and top $100,500$ and 1 000 samples with the smallest discrepancy), we adopted the median probability model with the 0.5 cutoff. For DART and BART, we choose variables which have been split on at least once on average. For Random forest, the RFE approach (as described in [143]) is used to find the variables.

Similarly as in [8], we report the number true positions discovered and the number of false positions. To further study the separation power, we also report AUC of each method. The results are shown in Table A.3, A.4 and A.5.

Across all the drugs, we notice that ABC Bayesian Forest has a strong separation power, as is indicated by the performance of AUC scores. Random Forests with RFE tends to overfit by selecting too many mutations. BART and DART are performing well in this case but ABC is seen to have better AUC while being overall more conservative.

Table A.3: The table summarizes results for a drug class PI. There are three performance criteria. For the adaptive cutoff, we report the number of true discoveries since the number of false discoveries is 0. For the automatic cutoff, we report both the number of false and true discoveries. Finally, we report a cutoff-free metric AUC. The best performance in each row is in bold font.

APV							
Methods		ABC			BART	DART	Random Forest
		100	500	1000			
Adaptive cut-off	True Discoveries	17	**19**	**19**	14	15	15
Automatic cut-off	False Discoveries	**0**	**0**	**0**	**0**	7	31
	True Discoveries	13	11	11	14	20	**34**
AUC		0.69	0.75	**0.77**	0.65	0.65	0.61
ATV							
Adaptive cut-off	True Discoveries	**23**	**23**	**23**	19	19	13
Automatic cut-off	False Discoveries	**0**	**0**	**0**	**0**	3	**0**
	True Discoveries	16	15	15	18	**21**	19
AUC		0.77	0.78	**0.79**	0.62	0.65	0.71
IDV							
Adaptive cut-off	True Discoveries	8	9	9	6	11	**13**
Automatic cut-off	False Discoveries	**1**	**1**	**1**	2	5	32
	True Discoveries	14	14	14	18	18	**34**
AUC		0.73	**0.75**	**0.75**	0.65	0.63	0.62
LPV							
Adaptive cut-off	True Discoveries	14	14	14	**15**	13	9
Automatic cut-off	False Discoveries	**0**	**0**	**0**	**0**	7	31
	True Discoveries	13	13	13	14	17	**34**
AUC		0.72	0.74	**0.75**	0.56	0.57	0.62
NFV							
Adaptive cut-off	True Discoveries	8	10	10	11	**16**	15
Automatic cut-off	False Discoveries	**1**	**1**	**1**	**1**	5	32
	True Discoveries	15	15	14	17	20	**34**
AUC		0.73	**0.74**	**0.74**	0.65	0.64	0.65
RTV							
Adaptive cut-off	True Discoveries	10	10	9	**13**	11	11
Automatic cut-off	False Discoveries	2	**1**	**1**	3	4	31
	True Discoveries	13	11	11	14	20	**34**
AUC		0.72	0.74	**0.75**	0.62	0.60	0.67
SQV							
Adaptive cut-off	True Discoveries	15	15	15	3	**17**	10
Automatic cut-off	False Discoveries	**0**	**0**	**0**	3	6	31
	True Discoveries	15	15	14	16	17	**34**
AUC		0.74	0.77	**0.78**	0.64	0.62	0.57

Table A.4: The table summarizes results for a drug class NRTI. There are three performance criteria. For the adaptive cutoff, we report the number of true discoveries since the number of false discoveries is 0. For the automatic cutoff, we report both the number of false and true discoveries. Finally, we report a cutoff-free metric AUC. The best performance in each row is in bold font.

X3TC							
Methods		ABC			BART	DART	Random Forest
		100	500	1000			
Adaptive cut-off	True Discoveries	6	**9**	**9**	4	5	6
Automatic cut-off	False Discoveries	**0**	**0**	**0**	4	3	6
	True Discoveries	6	5	5	7	12	**15**
AUC		**0.70**	**0.70**	**0.70**	0.62	0.64	0.66
ABC							
Adaptive cut-off	True Discoveries	8	8	7	7	10	**12**
Automatic cut-off	False Discoveries	2	**1**	1	**1**	7	2
	True Discoveries	10	10	10	11	14	**16**
AUC		0.74	0.73	**0.76**	0.66	0.71	0.74
AZT							
Adaptive cut-off	True Discoveries	7	7	7	3	10	**13**
Automatic cut-off	False Discoveries	2	**1**	**1**	6	8	2
	True Discoveries	12	11	11	14	**16**	15
AUC		0.71	0.72	**0.73**	0.70	0.69	0.75
D4T							
Adaptive cut-off	True Discoveries	**9**	8	**9**	5	0	8
Automatic cut-off	False Discoveries	2	**1**	**1**	3	12	80
	True Discoveries	12	12	11	12	14	**24**
AUC		**0.75**	**0.75**	**0.75**	0.70	0.70	0.73
DDI							
Adaptive cut-off	True Discoveries	5	5	6	7	3	**10**
Automatic cut-off	False Discoveries	**1**	**1**	**1**	2	11	81
	True Discoveries	8	7	7	8	13	**24**
AUC		0.71	0.73	**0.74**	0.68	0.66	0.72
TDF							
Adaptive cut-off	True Discoveries	4	**9**	**9**	3	7	2
Automatic cut-off	False Discoveries	2	**1**	**1**	4	11	8
	True Discoveries	9	9	9	10	18	**15**
AUC		0.69	0.72	0.72	0.72	**0.75**	0.73

Table A.5: The table summarizes results for a drug class NNRTI. There are three performance criteria. For the adaptive cutoff, we report the number of true discoveries since the number of false discoveries is 0. For the automatic cutoff, we report both the number of false and true discoveries. Finally, we report a cutoff-free metric AUC. The best performance in each row is in bold font.

		DLV					
Methods		ABC			BART	DART	Random Forest
		100	500	1000			
Adaptive cut-off	True Discoveries	**4**	**4**	**4**	3	3	3
Automatic cut-off	False Discoveries	**3**	**3**	**3**	**3**	8	96
	True Discoveries	7	7	7	9	10	**14**
AUC		0.84	**0.87**	**0.87**	0.73	0.70	0.81
		EFV					
Adaptive cut-off	True Discoveries	**5**	**5**	**5**	**5**	4	4
Automatic cut-off	False Discoveries	5	**4**	**4**	5	6	9
	True Discoveries	8	7	6	9	9	**10**
AUC		0.80	0.83	**0.84**	0.74	0.73	0.78
		NVP					
Adaptive cut-off	True Discoveries	6	6	6	8	6	**14**
Automatic cut-off	False Discoveries	3	3	**2**	**2**	9	97
	True Discoveries	6	6	5	**7**	6	5
AUC		0.79	0.79	0.79	0.71	0.66	**0.82**

APPENDIX B

APPENDIX FOR COMPUTATIONAL SPEED-UPS USING BANDIT APPROACH

B.1 Proof of Theorem

We will denote with $\mathcal{S}^\star = \arg\max_{\mathcal{S}} r^C_{\boldsymbol{\theta}^\star}(\mathcal{S})$ the optimal model where

$$r^C_{\boldsymbol{\theta}^\star}(\mathcal{S}) = \sum_{i \in \mathcal{S}} \left[\theta^\star_i(\mathcal{S}) \log\left(\frac{C+1}{C}\right) - \log\left(\frac{1}{C}\right) \right].$$

First, we define the reward gap of a set of arms $\mathcal{S}_t$ as

$$\Delta_{\mathcal{S}_t} = \mathbb{E}[r^C_{\boldsymbol{\theta}^\star}(\mathcal{S}^\star) - r^C_{\boldsymbol{\theta}}(\mathcal{S}_t)].$$

and write the expected regret in (3.12) as $Reg(T) = \mathbb{E}\sum_{t=1}^T \Delta_{\mathcal{S}_t}$. Before proceeding, we need to introduce some notation. We denote with $N_i(t) = \sum_{k<t} \mathbb{I}(i \in \mathcal{S}_k)$ the number of times an arm i has been pulled up to time t. Next,

$$\hat{\mu}_i(t) = \frac{a_i(t) - 1}{N_i(t)} = \frac{1}{N_i(t)} \sum_{k<t} \mathbb{I}(i \in \mathcal{S}_k)\gamma^t_i \tag{B.1}$$

denotes the empirical mean of an arm i, i.e. the proportion of times an arm i has yielded a reward when pulled, i.e. $\gamma^t_i = 1$ when $i \in \mathcal{S}_t$. We will be using the following usual Chernoff-Hoeffding bounds which we state without a proof.

Lemma B.1.1 (Chernoff-Hoeffding Bound)**.** *Let $X_1, ..., X_n$ be independent Bernoulli random variables with $\mathbb{E}[X_i] = p_i$ and denote with $X = \frac{1}{n}\sum_{i=1}^n = X_i$ and $\mu = \frac{1}{n}\sum_{i=1}^n p_i$. Then, for any $0 < \lambda < 1 - \mu$, we have*

$$\mathbb{P}(X \geq \mu + \lambda) \leq \exp\{-nd(\mu + \lambda, \mu)\},$$

and, for any $0 < \lambda < \mu$,

$$\mathbb{P}(X \leq \mu - \lambda) \leq \exp\{-nd(\mu - \lambda, \mu)\},$$

where $d(a, b) = a \ln a/b + (1 - a) \ln(1 - a)/(1 - b)$.

Similarly as in other regret bound proofs [123, 243] we will bound the expected regret separately on the intersection of combinations of the following events:

$$\begin{aligned}
\mathcal{A}(t) &= \{\mathcal{S}_t \neq \mathcal{S}^*\}, \\
\mathcal{B}(t) &= \left\{\exists i \in \mathcal{S}^* \ s.t. \ \hat{\mu}_i(t) < 0.5 + \frac{\alpha}{2} \text{ or } \exists i \in \mathcal{S}_t \backslash \mathcal{S}^* \ s.t. \ \hat{\mu}_i(t) > 0.5 - \frac{\alpha}{2}\right\}, \\
\mathcal{C}(t) &= \left\{\exists i \in \mathcal{S}^* \ s.t. \ \hat{\mu}_i(t) - \theta_i(t) > \frac{\alpha}{2} \text{ or } \exists i \in \mathcal{S}_t \backslash \mathcal{S}^* \ s.t. \ \theta_i(t) - \hat{\mu}_i(t) > \frac{\alpha}{2}\right\}, \\
\mathcal{D}(t) &= \bigcap_{i \in \mathcal{S}_t} \left\{N_i(t) > \frac{8 \log T}{\alpha^2}\right\},
\end{aligned}$$

where α occurred in Assumption 3.5.1. First, we focus on the following term

$$Reg_1(T) = \sum_{t=1}^{T} \mathbb{E}[\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}(t))]. \tag{B.2}$$

The following Lemma B.1.2 finds an upper bound on $Reg_1(T)$:

Lemma B.1.2. *Under the Assumption 1, the TVS sampling policy in Table 2 with* $C = (\sqrt{5} - 1)/2$ *yields, for* $\Delta_{max} = \max_{\mathcal{S}} \Delta_{\mathcal{S}}$

$$\begin{aligned}
Reg_1(T) \leq & \Delta_{\max} \sum_{i \in \mathcal{S}^*} \mathbb{E}\left(\sum_{t=1}^{T} \mathbb{I}\left\{i \in \mathcal{S}_t, \hat{\mu}_i(t) < 0.5 + \frac{\alpha}{2}\right\}\right) && \text{(B.3)} \\
& + \Delta_{\max} \sum_{i \notin \mathcal{S}^*} \mathbb{E}\left(\sum_{t=1}^{T} \mathbb{I}\left\{i \in \mathcal{S}_t, \hat{\mu}_i(t) > 0.5 - \frac{\alpha}{2}\right\}\right) && \text{(B.4)} \\
\leq & \Delta_{\max} \times p\left(1 + \frac{4}{\alpha_{\max}^2}\right). && \text{(B.5)}
\end{aligned}$$

Proof: We will first prove that

$$\mathbb{E}\left(\sum_{t=1}^{T}\mathbb{I}\left\{i\in\mathcal{S}_t,\hat{\mu}_i(t)<0.5+\frac{\alpha}{2}\right\}\right)\leq 1+\frac{4}{\alpha_{\max}^2}. \tag{B.6}$$

The second inequality for the term in (B.22) can be obtained analogously. With $\theta_i(\mathcal{S}_t)$ as in (3.14) we denote with

$$\bar{\theta}_i(T)=\frac{1}{N_i(T)}\sum_{t=1}^{T}\mathbb{I}(i\in\mathcal{S}_t)\theta_i(\mathcal{S}_t)$$

and with $d(a,b)=a\log[a/b]+(1-a)\log[(1-a)/(1-b)]$. Note that for any $b\in(0,1)$ both functions $d(x,b)$ and $d(b,x)$ are monotone increasing on $[b,1]$ and, at the same time, monotone decreasing on $[0,b]$. Under the Assumption 1 and for $i\in\mathcal{S}^\star$ we have $\bar{\theta}_i(T)>0.5+\alpha$ and thereby

$$d(0.5+\alpha/2,\bar{\theta}_i(T))>d(0.5+\alpha/2,0.5+\alpha).$$

Similarly as in the proof of Lemma 3 in Wang and Chen (2018), we denote with $\tau_1,\tau_2,...$ the times such that $i\in\mathcal{S}_t$, define $\tau_0=0$ and write

$$\begin{aligned}
\mathbb{E}\left(\sum_{t=1}^{T}\mathbb{I}\{i\in\mathcal{S}_t,\hat{\mu}_i(t)<0.5+\alpha/2\}\right) &\leq 1+\sum_{w=0}^{T}\mathbb{P}(\hat{\mu}_i(\tau_w)<0.5+\alpha/2,N_i(t)=w)\\
&\leq 1+\sum_{w=0}^{T}\exp\left(-w\,d\left(0.5+\alpha/2,\bar{\theta}(T)\right)\right)\\
&\leq 1+\sum_{w=0}^{T}\exp\left(-w\,d(0.5+\alpha/2,0.5+\alpha)\right)\\
&\leq 1+\sum_{w=0}^{\infty}\exp\left(-w\,d(0.5+\alpha/2,0.5+\alpha)\right)\\
&\leq 1+\frac{\exp\left(-d(0.5+\alpha/2,0.5+\alpha)\right)}{1-\exp\left(-d(0.5+\alpha/2,0.5+\alpha)\right)}\\
&\leq 1+\frac{4}{\alpha^2}.
\end{aligned}$$

This concludes the proof of (B.6). The second term can be bounded analogously, which concludes the proof of the Lemma. □

Next, we focus on the following term

$$Reg_2(T) = \sum_{t=1}^{T} \mathbb{E}[\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t))]. \tag{B.7}$$

To bound this term, we will be using the following Lemma (Lemma 4 from [243]) which we, again, state without a proof.

Lemma B.1.3. *Denote with $\theta_i(t)$ the mean reward for an arm i sampled from $\mathcal{B}(a_i(t), b_i(t))$ during the step C1 in Table 2. Using the TVS algorithm from Table 2, we have the following two inequalities for any base arm i:*

$$\mathbb{P}\left[\theta_i(t) - \hat{\mu}_i(t) > \sqrt{\frac{2\log T}{N_i(t)}}\right] \leq \frac{1}{T},$$

$$\mathbb{P}\left[\hat{\mu}_i(t) - \theta_i(t) > \sqrt{\frac{2\log T}{N_i(t)}}\right] \leq \frac{1}{T}.$$

Proof: The proof relies on the observation that $\theta_i(t)$'s only depend on values $a_i(t)$ and $b_i(t)$. The proof is then the same as in Lemma 4 in [243].

The following lemma bounds the regret term (B.7).

Lemma B.1.4. *Using the TVS algorithm from Table 2, we have the following bound:*

$$Reg_2(T) = \sum_{t=1}^{T} \mathbb{E}[\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t))] \leq \Delta_{\max} \times p.$$

Proof: On the event $\mathcal{D}(t)$, we have $N_i(t) > \frac{8\log T}{\alpha^2}$ and thereby $\frac{\alpha}{2} > \sqrt{2\frac{\log T}{N_i(t)}}$. The set $\mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)$ is then subsumed within

$$\left\{\exists i \in \mathcal{S}_t \backslash \mathcal{S}^* : \theta_i(t) - \hat{\mu}_i(t) > \sqrt{\frac{2\log T}{N_i(t)}} \text{ or } \exists i \in \mathcal{S}_t \cap \mathcal{S}^* : \hat{\mu}_i(t) - \theta_i(t) > \sqrt{\frac{2\log T}{N_i(t)}}\right\}.$$

We can then directly apply Lemma B.1.3 to write

$$\sum_{t=1}^{T} \Delta_{\mathcal{S}_t} \mathbb{P}\left(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)\right) \leq \Delta_{\max} \sum_{t=1}^{T} \mathbb{P}\left(\mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)\right)$$
$$\leq \Delta_{\max} \sum_{t=1}^{T} p/T. \qquad \square$$

Finally, we focus on the following term

$$Reg_3 = \sum_{t=1}^{T} \mathbb{E}\left(\Delta_{\mathcal{S}_t} \times \mathbb{I}\left(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}^c(t)\right)\right). \tag{B.8}$$

Lemma B.1.5. *Let $i \in \mathcal{S}^*$ and let $f_i^*(j,s)$ be the probability that after j pulls of an arm i, s of those pulls result in a reward. If $s \leq \lfloor 0.5j \rfloor$, then*

$$f_i^*(j,s) \leq \binom{j}{s} (0.5+\alpha)^s (0.5-\alpha)^{j-s} \tag{B.9}$$

Proof: We denote with τ_j^i the j^{th} time such that the arm i has been pulled (i.e. $\theta_i(t) > 0.5$). We denote the probability of yielding a reward at time τ_j as $p_j = \mathbb{P}\left(\gamma_i^{\tau_j} = 1 \mid \mathcal{S}_{\tau_j}\right)$ and, for a given j and s write $f_{j,s}^*(p_1, \cdots, p_j) := f_i^*(j,s)$. Since we are studying one particular arm, we have dropped the subscript i without any loss of generality. Consider now a vector of binary indicators $\boldsymbol{B} = (B_1, B_2, B_3, \cdots, B_j)' \in \{0,1\}^j$ where $B_k = \gamma_i^{\tau_k} \in \{0,1\}$ for whether or not the k^{th} pull of an arm i has yielded a reward. Denoting $\mid \boldsymbol{B} \mid = \sum_{j=1}^{p} B_k$, we can write

$$f_{j,s}^*(p_1, \cdots, p_j) = \sum_{\boldsymbol{B}:|\boldsymbol{B}|=s} \prod_{l=1}^{j} p_l^{B_l} (1-p_l)^{1-B_l}.$$

We want to show that $\boldsymbol{p}^\star = (p_1^\star, \ldots, p_j^\star)' = \arg\max f_{j,s}^*(p_1, \cdots, p_j)$ when $p_i^\star = 0.5 + \alpha$ for all $1 \leq i \leq j$. First, we notice is that this is a multi-linear polynomial in the sense that $\frac{\partial f_{j,s}^*(p_1, \cdots, p_j)}{\partial p_k}$ is independent of p_k. Keeping every other coordinate constant, the value $p_k^\star$ that maximizes $f_{j,s}^*(\cdot)$ in the k^{th} direction has to be either $0.5 + \alpha$ or 1. The vector $(p_1^\star, \cdots, p_j^\star)'$ maximizing $f_{j,s}^*(p_1, \cdots, p_j)$ will thus have each coordinate p_k^* either equal to 1

or $0.5+\alpha$. Let $r \in \mathbb{N} \cup \{0\}$ be the number of coordinates k for which $p_k^\star = 1$ and $j-r$ be the number of coordinates k for which $p_k^\star = 0.5+\alpha$ (notice that $r \leq s$). Since $f_{j,s}^*(p_1, \cdots, p_j)$ is a symmetric polynomial (i.e. the value of the function is not affected by a permutation of its argument) we assume, without loss of generality, that $p_1^\star = p_2^\star = \cdots = p_r^\star = 1$ and $p_{r+1}^\star = p_{r+2}^\star = \cdots = p_j^\star = 0.5+\alpha$. In this case, we have the constraint on the binary indicators $\boldsymbol{B}$ where the first r indices have to be 1 and the remaining $s-r$ $1'$s can be anywhere between the index $r+1$ and j ($j-r$ indices). Therefore, we have

$$f_{j,s,r}^*(\boldsymbol{p}^\star) = \binom{j-r}{s-r}(0.5+\alpha)^{s-r}(0.5-\alpha)^{j-s}.$$

It is sufficient to prove that this function is maximized at $r=0$. We have

$$\begin{aligned}
\frac{f_{j,s,r+1}^*(\boldsymbol{p}^\star)}{f_{j,s,r}^*(\boldsymbol{p}^\star)} &= \frac{\binom{j-r-1}{s-r-1}(0.5+\alpha)^{s-r-1}(0.5-\alpha)^{j-s}}{\binom{j-r}{s-r}(0.5+\alpha)^{s-r}(0.5-\alpha)^{j-s}} = \frac{s-r}{(j-r)(1/2+\alpha)} \\
&\leq \frac{j/2-r}{(j-r)(1/2+\alpha)} \quad \text{(using the assumption } s \leq \lfloor j/2 \rfloor) \\
&= 1 - \frac{(1/2-\alpha)r + \alpha j}{(j-r)(1/2+\alpha)} < 1
\end{aligned}$$

since $1/2-\alpha \geq 0$ and $\alpha > 0$. Since this is true for all r, the function $\frac{f_{j,s,r+1}^*(p_1,\cdots,p_j)}{f_{j,s,r}^*(p_1,\cdots,p_j)}$ is maximized at $r=0$. This concludes the proof. □

Lemma B.1.6. *Let $i \in \mathcal{S}^*$ and let τ_j^i be the j^{th} time such that $\theta_i(t) > 0.5$. Suppose that Assumption 1 is true, then the TVS Algorithm 1 with $C = (\sqrt{5}-1)/2$ satisfies*

$$\mathbb{E}\left[\tau_{j+1} - \tau_j\right] \leq \begin{cases} 4 + \frac{1}{\alpha} & \text{when} \quad j \leq \frac{8}{\alpha} \\ 1 + \frac{1}{\mathbf{e}^{\alpha^2 j/4}-1} + \mathbf{e}^{-\alpha^2 j/2}\left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(j+1)}\right) & \text{when} \quad j > \frac{8}{\alpha}, \end{cases} \tag{B.10}$$

where constants $C_1, C_2 > 0$ are not related to Algorithm 1.

Proof: We denote with τ_j^i the j^{th} time such that the arm i has been pulled (i.e. $\theta_i(t) > 0.5$). First, we consider the time interval $[\tau_j^i, \tau_{j+1}^i)$. For any $t \in [\tau_j^i, \tau_{j+1}^i)$ we know that the arm

i has been played j times and, thereby, $\theta_i(t)$ comes from a beta distribution

$$\theta_i(t) \sim \mathcal{B}\left[a_i(t), b_i(t)\right],$$

where $j = a_i(t) + b_i(t) - 2$. The parameters of the beta distribution are only updated if the arm i is pulled and the distribution thus does not change until we reach the iteration τ_{j+1}. Therefore, given $\hat{\mu}_i(\tau_j^i)$ the expected difference between $\tau_{j+1}^i - \tau_j^i$ has a geometric distribution with an expectation

$$\mathbb{E}\left[\tau_{j+1} - \tau_j \mid \hat{\mu}_i(\tau_j^i)\right] = \frac{1}{\mathbb{P}(B_{ij} > 0.5)} = \frac{1}{p_{i,j}(0.5)}$$

where $B_{ij} \sim \mathcal{B}\left[a_i(\tau_{j+1}), b_i(\tau_{j+1})\right]$. We let $F_{n,p}(\cdot)$ and $f_{n,p}(\cdot)$ denote the cumulative distribution function (CDF) and the probability density function of a Binomial distribution with parameters (n, p). We now recall the following fact (see e.g. Fact 3 in [3]) about the CDF $F_{\alpha,\beta}^{\text{beta}}(x)$ of a beta distribution with parameters (α, β). We have the following identity which links the CDF of a beta distribution and a CDF of a binomial distribution:

$$F_{\alpha,\beta}^{\text{beta}}(y) = 1 - F_{\alpha+\beta-1,y}(\alpha - 1) \tag{B.11}$$

for all positive integers α, β. Let $f_i^*(j, s)$ be the probability that after j pulls of an arm i, s out of those j pulls result in a reward. Here, we have the following relationship $a_i(\tau_j) = s+1$ and $b_i(\tau_j) = j + 1 - s$. Using the identity (B.11) and given s successes among j pulls, we can write $p_{i,j}(0.5) = F_{s+1,j-s+1}^{\text{beta}}(0.5) = 1 - F_{j+1,0.5}(s)$ and thereby

$$\mathbb{E}\left[\frac{1}{p_{i,j}(0.5)}\right] = \sum_{s=0}^{j} \frac{f_i^*(j, s)}{F_{j+1,0.5}(s)}. \tag{B.12}$$

First, we consider the case when $j \leq \frac{8}{\alpha}$. In the following calculations, we will use the result from Lemma B.1.5. Let $p_{\max} = \alpha + 0.5$ and $R = \frac{p_{\max}}{1-p_{max}}$. Using the fact

$F_{j+1,0.5}(s) \geq 0.5F_{j,0.5}(s)$ and $F_{j,0.5}(s) \geq 1/2$ when $s \geq \lceil j/2 \rceil$ (since the median of Binomial distribution with parameters $(j, 1/2)$ is either $\lfloor j/2 \rfloor$ or $\lceil j/2 \rceil$) we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{p_{i,j}(0.5)}\right] \leq & 2\sum_{s=0}^{j} \frac{f_i^*(j,s)}{F_{j,0.5}(s)} && \text{(B.13)}\\
\leq & 2\sum_{s=0}^{\lfloor j/2 \rfloor} \frac{\binom{j}{s} p_{\max}^s (1-p_{\max})^{j-s}}{f_{j,0.5}(s)} + 4\sum_{s=\lceil j/2 \rceil}^{j} f_i^*(j,s) && \text{(B.14)}\\
\leq & 2\sum_{s=0}^{\lfloor j/2 \rfloor} \frac{p_{\max}^s (1-p_{\max})^{j-s}}{1/2^j} + 4 && \text{(B.15)}\\
\leq & 2\frac{(1-p_{\max})^j}{1/2^j}\sum_{s=0}^{\lfloor j/2 \rfloor} R^s + 4 && \text{(B.16)}\\
\leq & 2\left(\frac{R^{\lfloor j/2 \rfloor+1}-1}{R-1}\right)\frac{(1-p_{\max})^j}{1/2^j} + 4 && \text{(B.17)}\\
\leq & 2\left(\frac{R}{R-1}\right) R^{j/2}\frac{(1-p_{\max})^j}{1/2^j} + 4 && \text{(B.18)}\\
\leq & \frac{1}{\alpha}\mathbf{e}^{-j\, d(1/2,\, p_{\max})} + 4 && \text{(B.19)}\\
\leq & \frac{1}{\alpha} + 4, && \text{(B.20)}
\end{aligned}$$

where from (B.18) to (B.19) we have used the following two facts. First, using the definition of $d(\cdot,\cdot)$ in Lemma 4 and the fact that $d(p_1,p_2) > (p_1-p_2)^2$ we obtain

$$\begin{aligned}
\frac{(1-p_{\max})^j}{1/2^j} R^{\lfloor j/2 \rfloor} \leq \frac{(1-p_{\max})^j}{1/2^j} R^{j/2} &= \mathbf{e}^{j\log(1-p_{\max}) - j\log(1/2) + j/2\log(p_{\max}) - j/2\log(1-p_{\max})}\\
&= \mathbf{e}^{-j\left\{\frac{1}{2}\log(\frac{1}{2}/p_{\max}) + \frac{1}{2}\log[\frac{1}{2}/(1-p_{\max})]\right\}} = \mathbf{e}^{-j\, d(1/2,\, p_{\max})} \leq \mathbf{e}^{-\alpha^2 j}.
\end{aligned}$$

Second, since $p_{\max} = 0.5 + \alpha$, we have $\frac{R}{R-1} = \frac{p_{\max}}{2\alpha} \leq \frac{1}{2\alpha}$.

When $j > \frac{8}{\alpha}$, we will divide the sum $\Sigma(0,j) \equiv \sum_{s=0}^{j} \frac{f_i^*(j,s)}{F_{j+1,0.5}(s)}$ into 4 pieces and bound each

one of them

$$\Sigma\left(0, \lfloor j/2 \rfloor - 1\right) \leq c_2 \left[\mathbf{e}^{-\alpha^2 j} \frac{1-2\alpha}{4\alpha^2(j+1)}\right] + c_3\, \mathbf{e}^{-2\alpha^2 j} \tag{B.21}$$

$$\Sigma\left(\lfloor j/2 \rfloor, \lfloor j/2 \rfloor\right) \leq 3\, \mathbf{e}^{-\alpha^2 j} \tag{B.22}$$

$$\Sigma\left(\lceil j/2 \rceil, \lfloor (1/2+\alpha/2)j \rfloor\right) \leq c_3 \left(\mathbf{e}^{-\alpha^2 j/2}\right) \tag{B.23}$$

$$\Sigma\left(\lceil (1/2+\alpha/2)j \rceil, j\right) \leq 1 + \frac{1}{\mathbf{e}^{\alpha^2 j/4} - 1}, \tag{B.24}$$

where $c_2 > 0$ and $c_3 > 0$ are constants unrelated to Algorithm 2. This will complete the proof. We now prove the bounds in the last display. We start with the first inequality in (B.21). When $s \leq (j+1)/2 - \sqrt{(j+1)/4}$, we use the following bound for the Binomial CDF ([112])

$$F_{j+1,0.5}(s) \geq \frac{1}{c_2}\left[\frac{j+1-s}{j+1-2s}\begin{pmatrix} j+1 \\ s \end{pmatrix}\frac{1}{2^{j+1}}\right],$$

for some $c_2 > 0$. When $s \geq (j+1)/2 - \sqrt{(j+1)/4}$ we use that fact that, for some $c_3 > 0$,

$$F_{j+1,0.5}(s) \geq \frac{1}{c_3} > 0.$$

Altogether, we arrive at the following bound (using again Lemma B.1.5 and denoting $p_{\max} =$

$1/2+\alpha$ and $R=\frac{p_{\max}}{1-p_{max}}$)

$$\Sigma\left(0,\lfloor j/2\rfloor-1\right) \leq c_2 \sum_{s=0}^{\lceil (j+1)/2-\sqrt{(j+1)/4}\rceil} \frac{f_i^*(j,s)}{\frac{j+1-s}{j+1-2s}\begin{pmatrix} j+1 \\ s \end{pmatrix}\frac{1}{2^{j+1}}} \tag{B.25}$$

$$+ c_3 \sum_{s=\lceil (j+1)/2-\sqrt{(j+1)/4}\rceil+1}^{\lfloor j/2\rfloor-1} f_i^*(j,s)$$

$$\leq c_2 \sum_{s=0}^{\lfloor j/2\rfloor-1} \frac{f_i^*(j,s)}{\frac{j+1-s}{j+1-2s}\begin{pmatrix} j+1 \\ s \end{pmatrix}\frac{1}{2^{j+1}}} + c_3 \sum_{s=0}^{\lfloor j/2\rfloor-1} f_i^*(j,s)$$

$$\leq c_2 \left(\frac{(1-p_{\max})^j}{1/2^{j+1}} \sum_{s=0}^{\lfloor j/2\rfloor-1} \left(1-\frac{2s}{j+1}\right) R^s \right) + c_3 \sum_{s=0}^{\lfloor j/2\rfloor-1} f_i^*(j,s). \tag{B.26}$$

Now we bound the first term in (B.26) to obtain

$$\frac{(1-p_{\max})^j}{1/2^{j+1}} \sum_{s=0}^{\lfloor j/2\rfloor-1} \left(1-\frac{2s}{j+1}\right) R^s$$

$$= \frac{(1-p_{\max})^j}{1/2^{j+1}} \left\{ \frac{R^{\lfloor j/2\rfloor}-1}{R-1} - \frac{2}{j+1}\left[\frac{\left(\lfloor j/2\rfloor-1\right)R^{\lfloor j/2\rfloor}}{R-1} - \frac{R^{\lfloor j/2\rfloor}-R}{(R-1)^2} \right] \right\}$$

$$\leq \frac{(1-p_{\max})^j}{1/2^{j+1}} \left\{ \frac{R^{\lfloor j/2\rfloor}}{R-1} - \frac{2}{j+1}\left[\frac{\left(\lfloor j/2\rfloor-1\right)R^{\lfloor j/2\rfloor}}{R-1} - \frac{R^{\lfloor j/2\rfloor}}{(R-1)^2} \right] \right\}$$

$$\leq \frac{(1-p_{\max})^j}{1/2^{j+1}} \left[\frac{2}{j+1}\frac{R^{\lfloor j/2\rfloor}}{(R-1)^2} + \frac{2[(j+1)/2-\lfloor j/2\rfloor+1]}{j+1}\frac{R^{\lfloor j/2\rfloor}}{R-1} \right]$$

$$\leq \frac{(1-p_{\max})^j}{1/2^{j+1}} \frac{6}{j+1}\frac{R^{\lfloor j/2\rfloor+1}}{(R-1)^2}$$

$$\leq \mathbf{e}^{-\alpha^2 j}\frac{12}{j+1}\frac{R}{(R-1)^2},$$

where we have used the following facts. First, for any $x > 1$ we have

$$\sum_{s=0}^{n} s\, x^s = \frac{nx^{n+2} - (n+1)x^{n+1} + x}{(1-x)^2} = \frac{nx^{n+1}}{x-1} - \frac{x^{n+1} - x}{(x-1)^2}.$$

Second, $j/2 + 1/2 - \lfloor j/2 \rfloor + 1 < 3$ and (similarly as before)

$$\frac{(1-p_{\max})^j}{1/2^j} R^{\lfloor j/2 \rfloor} \leq \mathbf{e}^{-j\, d(1/2,\, p_{\max})} \leq \mathbf{e}^{-\alpha^2 j}.$$

Finally, since $R/(R-1) \leq \frac{1}{2\alpha}$ and $1/(R-1) = \frac{1-2\alpha}{4\alpha}$, we have

$$\frac{1}{(j+1)/4} \frac{R}{(R-1)^2} \leq \frac{1-2\alpha}{4\alpha^2(j+1)}.$$

For the second term in (B.26), we notice that $\sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j,s)$ is equal to the probability that the total number of successes is less than $\lfloor j/2 \rfloor - 1$. Here, we invoke Lemma B.1.1 and note that the success probability of each pull is always greater than $1/2 + \alpha$ and the difference between the average success probability over the j pulls and $1/2$ is thereby greater than α. Hence, $\sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j,s) \leq \mathbf{e}^{-2\alpha^2 j}$. We put the two terms together to finally obtain

$$\Sigma(0, \lfloor j/2 \rfloor - 1) \leq c_2 \left[\mathbf{e}^{-\alpha^2 j} \frac{1-2\alpha}{4\alpha^2(j+1)} \right] + c_3 \sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j,s) \tag{B.27}$$

$$\leq c_2 \left[\mathbf{e}^{-\alpha^2 j} \frac{1-2\alpha}{4\alpha^2(j+1)} \right] + c_3\, \mathbf{e}^{-2\alpha^2 j}. \tag{B.28}$$

Next, to bound the term $\Sigma(\lfloor j/2 \rfloor, \lfloor j/2 \rfloor)$ in (B.22), we use Lemma B.1.5 and the fact

that $p_{max} > 1/2$ to find that for $s = \lfloor j/2 \rfloor$

$$\begin{aligned}\Sigma(\lfloor j/2 \rfloor, \lfloor j/2 \rfloor) = \frac{f_i^*(j,s)}{F_{j+1,0.5}(s)} \leq \frac{f_i^*(j,s)}{f_{j+1,0.5}(s)} &\leq 2\left(1 - \frac{s}{j+1}\right) R^s \left(\frac{1-p_{\max}}{1/2}\right)^j \\ &\leq \frac{2}{j+1}\left(\frac{j}{2}+2\right) R^{j/2} \left(\frac{1-p_{\max}}{1/2}\right)^j \\ &\leq \left(1 + \frac{3}{j+1}\right) \mathbf{e}^{-\alpha^2 j} \leq 2\,\mathbf{e}^{-\alpha^2 j},\end{aligned}$$

where we used the assumption $j \geq 1/\alpha > 2$.

In order to bound the third term $\Sigma(\lceil j/2 \rceil, \lfloor (\alpha+1)j/2 \rfloor)$ in (B.23), we first note that if $j \geq \frac{8}{\alpha} > \frac{1}{\alpha} \geq 2$ (our assumption above), we have $\sqrt{(j+1)/4} > \sqrt{3/4} \geq \sqrt{1/2} > 1/2$ and thereby $(j+1)/2 - \sqrt{(j+1)/4} < j/2 \leq \lceil j/2 \rceil \leq s$. This implies that the condition in [112] is satisfied and we can apply the bound $F_{j+1,0.5}(s) \geq \frac{1}{c_3}$. Then we have

$$\begin{aligned}\Sigma(\lceil j/2 \rceil, \lfloor (\alpha+1)j/2 \rfloor) &\leq c_3 \left(\sum_{s=\lceil j/2 \rceil}^{\lfloor (\alpha+1)j/2 \rfloor} f_i^*(j,s)\right) \leq c_3 \left(\sum_{s=0}^{\lfloor (\alpha+1)j/2 \rfloor} f_i^*(j,s)\right) \\ &\leq c_3 \left(\mathbf{e}^{-\alpha^2 j/2}\right),\end{aligned}$$

where the last inequality stems from the Chernoff-Hoeffding inequality in Lemma B.1.1 and Assumption 1 which guarantees that the success probability of each pull is greater than $1/2 + \alpha$. This implies that the difference between the average probability of success over all the j pulls and $1/2 + \alpha/2$ is greater than $\alpha/2$.

Finally, to bound the term $\Sigma(\lceil (\alpha+1)j/2 \rceil, j)$ in (B.24) we use the Hoeffding inequality in Lemma B.1.1 with $\lambda = (\alpha+1)j/[2(j+1)] - 1/2 \leq s/(j+1) - 1/2$ to find (for a r.v. $X \sim Bin(j+1, 1/2)$) that

$$\begin{aligned}F_{j+1,0.5}(s) &\geq 1 - \mathbb{P}\left(\frac{X}{j+1} - \frac{1}{2} > \lambda\right) \geq 1 - \mathbf{e}^{-2(j+1)\lambda^2} = 1 - \mathbf{e}^{(-\frac{j\alpha^2}{2} + \frac{j\alpha^2}{2(j+1)} + \alpha\frac{j}{j+1} - \frac{1}{2(j+1)})} \\ &\geq 1 - \mathbf{e}^{-\alpha^2 j/2},\end{aligned}$$

where we used the fact that $1/j \leq \alpha/8$ and thereby $2\alpha \geq \frac{j\alpha^2}{2(j+1)} + \alpha\frac{j}{j+1} - \frac{1}{2(j+1)}$. Finally, we write

$$\Sigma(\lceil(\alpha+1)j/2\rceil, j) \leq \sum_{s=\lceil(\alpha+1)j/2\rceil}^{j} \frac{f_i^*(j,s)}{F_{j+1,0.5}(s)} \leq \frac{1}{1-\mathbf{e}^{-\alpha^2 j/4}} = 1 + \frac{1}{\mathbf{e}^{\alpha^2 j/4}-1}.$$

Now, denoting $C_1 = 2+2\,c_3$ and $C_2 = c_2$ we get the statement in the Lemma. This concludes our proof. □

Using Lemma B.1.6, we can achieve a similar bound in Lemma 6 of [243],

Lemma B.1.7. *Under Assumption 1, the TVS Algorithm 2 with $C = (\sqrt{5}-1)/2$ satisfies the following property. For any signal arm $i \in \mathcal{S}^*$, the expected number of total pulls before the given arm i is pulled $\frac{8\log(T)}{\alpha^2}$ times is bounded by*

$$\left\lceil\frac{8\log(T)}{\alpha^2}\right\rceil + \left\lceil\frac{8}{\alpha}\right\rceil\left(3+\frac{1}{\alpha}\right) + \widetilde{C}\,\frac{\mathbf{e}^{-4\alpha}}{1-\mathbf{e}^{-\alpha^2/2}} + \frac{8}{\alpha^2}\frac{1}{\mathbf{e}^{2\alpha}-1} + \frac{\mathbf{e}^{-1}}{1-\mathbf{e}^{-\alpha/8}},$$

where $\widetilde{C} = C_1 + C_2\frac{1-2\alpha}{32\,\alpha}$ for some $C_1 > 0$ and $C_2 > 0$ not related to the Algorithm 2.

Proof: We use the notation from Lemma B.1.6, where τ_j^i is the time when the arm i has been pulled for the j^{th} time. Denoting with $\widetilde{T} = \lfloor\frac{8\log(T)}{\alpha^2}\rfloor$, we want to find an upper bound for $\mathbb{E}[\tau^i_{\widetilde{T}}]$. We can write

$$\mathbb{E}\left[\tau^i_{\widetilde{T}}\right] = \sum_{j=0}^{\widetilde{T}} \mathbb{E}\left[\tau^i_{j+1} - \tau^i_j\right] = \sum_{j=0}^{\widetilde{T}} \mathbb{E}\left(\frac{1}{p_{i,j}(0.5)}\right)$$

and using Lemma B.1.6 we obtain

$$\mathbb{E}\left[\tau^i_{\widetilde{T}}\right] \leq \sum_{j=0}^{\lfloor\frac{8}{\alpha}\rfloor}\left(4+\frac{1}{\alpha}\right) + \sum_{j=\lfloor\frac{8}{\alpha}\rfloor+1}^{\widetilde{T}}\left[1 + \frac{1}{\mathbf{e}^{\alpha^2 j/4}-1} + \mathbf{e}^{-\alpha^2 j/2}\left(C_1 + C_2\frac{1-2\alpha}{4\alpha^2(j+1)}\right)\right]. \tag{B.29}$$

First, we note that $\sum_{j=0}^{\lfloor \frac{8}{\alpha} \rfloor} \left(4 + \frac{1}{\alpha}\right) + \sum_{j=\lfloor \frac{8}{\alpha} \rfloor + 1}^{\widetilde{T}} 1 \leq \lceil \frac{8 \log(T)}{\alpha^2} \rceil + \lceil \frac{8}{\alpha} \rceil (3 + \frac{1}{\alpha})$. Next, we write

$$\sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\widetilde{T}} \mathbf{e}^{-\frac{j\alpha^2}{2}} \leq \frac{\mathbf{e}^{-4\alpha}}{1 - \mathbf{e}^{-\alpha^2/2}}$$

and

$$\sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\widetilde{T}} \mathbf{e}^{-\frac{j\alpha^2}{2}} \left(C_1 + C_2 \frac{1 - 2\alpha}{4\alpha^2 (j+1)}\right) < \sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\widetilde{T}} \mathbf{e}^{-\frac{j\alpha^2}{2}} \left(C_1 + C_2 \frac{1 - 2\alpha}{32\,\alpha}\right) \leq \widetilde{C} \frac{\mathbf{e}^{-4\alpha}}{1 - \mathbf{e}^{-\alpha^2/2}}.$$

Finally, we use the fact that $\frac{1}{\mathbf{e}^x - 1} \leq \mathbf{e}^{-x/2}$ for $x \geq 1$ to obtain

$$\begin{aligned}
\sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\widetilde{T}} \frac{1}{\mathbf{e}^{\frac{j\alpha^2}{4}} - 1} &\leq \sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\lfloor \frac{8}{\alpha^2} \rfloor} \frac{1}{\mathbf{e}^{\frac{j\alpha^2}{4}} - 1} + \sum_{j=\lceil \frac{8}{\alpha^2} \rceil}^{\lfloor \frac{8 \log(T)}{\alpha^2} \rfloor} \frac{1}{\mathbf{e}^{\frac{j\alpha^2}{4}} - 1} \\
&\leq \frac{8}{\alpha^2} \frac{1}{\mathbf{e}^{2\alpha} - 1} + \sum_{j=\lceil \frac{8}{\alpha^2} \rceil}^{\lfloor \frac{8 \log(T)}{\alpha^2} \rfloor} \mathbf{e}^{-\alpha^2 j/8} \leq \frac{8}{\alpha^2} \frac{1}{\mathbf{e}^{2\alpha} - 1} + \frac{\mathbf{e}^{-1}}{1 - \mathbf{e}^{-\alpha^2/8}}
\end{aligned}$$

□

Using these lemmas we can prove the following lemma about $Reg_3(T)$.

Lemma B.1.8. *We denote with $Reg_3(T)$ the regret term in* (B.8)*. Under Assumption 1, the TVS Algorithm 2 with $C = (\sqrt{5} - 1)/2$ yields*

$$Reg_3(T) \leq \Delta_{\max} \left\{ \frac{8p \log(T)}{\alpha^2} + \widetilde{C}\, q^* \frac{\mathbf{e}^{-4\alpha}}{1 - \mathbf{e}^{-\alpha^2/2}} + q^* \left[\frac{8}{\alpha^2} \frac{1}{\mathbf{e}^{2\alpha} - 1} + \frac{\mathbf{e}^{-1}}{1 - \mathbf{e}^{-\alpha/8}} + \left\lceil \frac{8}{\alpha} \right\rceil \left(3 + \frac{1}{\alpha}\right) \right] \right\},$$

where $\widetilde{C} = C_1 + C_2 \frac{1-2\alpha}{32\,\alpha}$ for some $C_1 > 0$ and $C_2 > 0$ not related to the Algorithm 2.

Proof: We start with the following facts $Reg_3(T) \leq \sum_{t=1}^{T} \Delta_{\mathcal{S}_t} \mathbb{E}\left[\mathbb{I}\left(\mathcal{A}(t) \cap \mathcal{D}^c(t)\right)\right]$ and

$$Reg_3(T) \leq \sum_{t=1}^{T} \sum_{i \in \mathcal{S}_t} \Delta_{\mathcal{S}_t} \mathbb{E}\left[\mathbb{I}\left(\mathcal{A}(t) \cap \left\{N_i(t) \leq \frac{8 \log(T)}{\alpha^2}\right\}\right)\right],$$

where we used the fact that on the event $\mathcal{D}^c(t)$, there exists at least one arm $i \in \mathcal{S}_t$ such that $N_i(t) \leq 8\frac{\log(T)}{\alpha^2}$. We now decompose the sum above into signal arms and noise arms

$$Reg_3(T) \leq \sum_{t=1}^{T} \left[\sum_{i \in \mathcal{S}_t \cap \mathcal{S}^\star} \Delta_{\mathcal{S}_t} \mathbb{E}\left[\mathbb{I}\left(\mathcal{A}(t) \cap \left\{ N_i(t) \leq \frac{8\log(T)}{\alpha^2} \right\} \right) \right] \right] \tag{B.30}$$

$$+ \sum_{t=1}^{T} \left[\sum_{i \in \mathcal{S}_t \setminus \mathcal{S}^\star} \Delta_{\mathcal{S}_t} \mathbb{E}\left[\mathbb{I}\left(\mathcal{A}(t) \cap \left\{ N_i(t) \leq \frac{8\log(T)}{\alpha^2} \right\} \right) \right] \right]. \tag{B.31}$$

If $i \in \mathcal{S}_t \setminus \mathcal{S}^\star$, then $\mathcal{S}_t$ contributes to the regret but this can *only* happen $\frac{8\log(T)}{\alpha^2}$ times so the total regret contribution of pulling a subset $\mathcal{S}_t$ including an arm i before $N_i(t) > \frac{8\log(T)}{\alpha^2}$ is bounded by $\max_{\mathcal{S}: i \in \mathcal{S}} \frac{8\Delta_{\mathcal{S}}\log(T)}{\alpha^2}$. There are $p - q^*$ noise arms $i \notin \mathcal{S}^*$ and the term (B.31) can be thus bounded by $(p-q^*)\Delta_{max}\frac{8\log(T)}{\alpha^2}$.

If $i \in \mathcal{S}^\star \cap \mathcal{S}_t$, then the arm i contributes to the regret when $\mathcal{S}_t \neq \mathcal{S}^\star$. However, by Lemma B.1.6 and Lemma B.1.7 we can bound the expected number of pulls of an arm i before $N_i(t)$ reaches $\frac{8\log(T)}{\alpha^2}$. This means that the contribution to the regret when $i \in \mathcal{S}^\star \cap \mathcal{S}_t$ is bounded by $\Delta_{\max}\left(\left\lceil \frac{8\log(T)}{\alpha^2} \right\rceil + \left\lceil \frac{8}{\alpha} \right\rceil \left(3 + \frac{1}{\alpha}\right) + \widetilde{C}\,\frac{\mathbf{e}^{-4\alpha}}{1-\mathbf{e}^{-\alpha^2/2}} + \frac{8}{\alpha^2}\frac{1}{\mathbf{e}^{2\alpha}-1} + \frac{\mathbf{e}^{-1}}{1-\mathbf{e}^{-\alpha/8}} \right)$. Because there are $q^\star$ signal arms, we can combine (B.30) and (B.31) to arrive at the bound in the statement of this lemma. □

We now put the various pieces together to finally prove Theorem 3.5.1.

Proof: We start by noticing that

$$\begin{aligned}\mathbb{I}\left[\mathcal{A}(t)\right] &= \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \mathbb{I}\left[\mathcal{A}(t) \cap \mathcal{B}^c(t)\right] \\ &= \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \mathbb{I}\left[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t)\right] + \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}^c(t)].\end{aligned}$$

Now we note that $\mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}^c(t)] = 0$ because

$$\mathcal{B}^c(t) \cap \mathcal{C}^c(t) = \{\forall i \in \mathcal{S}^* \text{ we have } \theta_i(t) > 0.5 \quad \text{and} \quad \forall i \in \mathcal{S}_t \setminus \mathcal{S}^* \text{ we have } \theta_i(t) < 0.5\} = \mathcal{A}^c(t).$$

Thereby we can write

$$\mathbb{I}\left[\mathcal{A}(t)\right] = \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \mathbb{I}\left[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}^c(t)\right] + \mathbb{I}\left[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)\right]$$

which leads to the following decomposition

$$\begin{aligned}
Reg(T) &= \sum_{t=1}^{T} \Delta_{\mathcal{S}_t} \mathbb{E}\left[\mathbb{I}\left(\mathcal{A}(t)\right)\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\Bigg[\Delta_{\mathcal{S}_t} \times \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \Delta_{\mathcal{S}_t} \times \mathbb{I}\left[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)\right] \\
&\quad + \Delta_{\mathcal{S}_t} \times \mathbb{I}\left[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}^c(t)\right]\Bigg] \\
&= Reg_1(T) + Reg_2(T) + Reg_3(T).
\end{aligned}$$

Now, we bound $Reg_1(T)$, $Reg_2(T)$ and $Reg_3(T)$ with Lemma B.1.2,Lemma B.1.4, and Lemma B.1.8. This gives us Theorem 3.5.1.

B.1.1 Proof of Theorem 3.5.2

Proof:

Let's define the event $\mathcal{A}(t) = \{\mathcal{S}_t \neq \mathcal{S}^*\}$. It is sufficient to show that the probability of infinitely many events $\mathcal{A}(t)$ occurring is zero, i.e. $\mathbb{P}\left(\limsup_{t\to\infty} \mathcal{A}(t)\right) = 0$. This implies that $\mathbb{P}\left(\liminf_{t\to\infty} \{\mathcal{S}_t = \mathcal{S}^*\}\right) = 1$. By the Borel-Cantelli Lemma, it suffices to show that

$$\sum_{t=1}^{\infty} \mathbb{P}(\mathcal{A}(t)) < \infty. \tag{B.32}$$

For each $i \in \{1, \ldots, p\}$ we define an event

$$\mathcal{A}_i(t) = \begin{cases} \{\theta_i(t) > 0.5\} & \text{when} \quad i \in \mathcal{S} \backslash \mathcal{S}^* \\ \{\theta_i(t) < 0.5\} & \text{when} \quad i \in \mathcal{S}^* \end{cases} .$$

Using the fact that

$$\mathcal{A}(t) \subseteq \bigcup_{i=1}^{p} \mathcal{A}_i(t) \tag{B.33}$$

one has

$$\left(\sum_{t=1}^{\infty} \mathbb{P}(\mathcal{A}(t)) \right) \leq \left(\sum_{t=1}^{\infty} \sum_{i=1}^{p} \mathbb{P}(\mathcal{A}_i(t)) \right) = \sum_{i=1}^{p} \sum_{t=1}^{\infty} \mathbb{P}\left(\mathcal{A}_i(t)\right).$$

We now show that the sum on the right hand side is finite, splitting the arms into the true arms $i \in \mathcal{S}^*$ and the fake arms $i \in \mathcal{S} \backslash \mathcal{S}^*$.

Lemma B.1.9. *Under assumptions of Theorem 3.5.2 we have, for each $i \in \mathcal{S} \backslash \mathcal{S}^*$,*

$$\sum_{t=1}^{\infty} \mathbb{P}(\mathcal{A}_i(t)) < 1 + \frac{4}{\alpha^2} + \frac{1}{1 - \exp\left(-\alpha^2/8\right)}.$$

Proof:

We can write

$$\begin{aligned} \mathcal{A}_i(t) &= \{\theta_i(t) > 0.5\} \\ &= \{\hat{\mu}_i(t) \geq 0.5 - \alpha/2, \theta_i(t) > 0.5\} \cup \{\hat{\mu}_i(t) \leq 0.5 - \alpha/2, \theta_i(t) > 0.5\} \\ &= \{\hat{\mu}_i(t) \geq 0.5 - \alpha/2, \theta_i(t) > 0.5\} \cup \{\hat{\mu}_i(t) \leq 0.5 - \alpha/2, \theta_i(t) - \hat{\mu}_i(t) > \alpha/2, \theta_i(t) > 0.5\} \\ &\subseteq \{\hat{\mu}_i(t) \geq 0.5 - \alpha/2, \theta_i(t) > 0.5\} \cup \{\theta_i(t) - \hat{\mu}_i(t) > \alpha/2, \theta_i(t) > 0.5\}. \end{aligned}$$

Since $\{\theta_i(t) > 0.5\} = \{i \in \mathcal{S}_t\}$, we first show that

$$\sum_{t=1}^{\infty} \mathbb{P}\left(i \in \mathcal{S}_t, \hat{\mu}_i(t) \geq 0.5 - \alpha/2\right) \leq 1 + \frac{4}{\alpha^2} \tag{B.34}$$

We can write, using our Lemma 2 in the Supplement,

$$\sum_{t=1}^{\infty} \mathbb{P}\left(i \in \mathcal{S}_t, \hat{\mu}_i(t) \geq 0.5 - \alpha/2\right) = \lim_{T\to\infty} \mathbb{E} \sum_{t=1}^{T} \mathbb{I}\left(i \in \mathcal{S}_t, \hat{\mu}_i(t) \geq 0.5 - \alpha/2\right) \leq 1 + \frac{4}{\alpha^2}.$$

Recall the definition of $N_i(t) = \sum_{k<t} \mathbb{I}(i \in \mathcal{S}_k) = a_i(t) + b_i(t) - 2$ and $\hat{\mu}_i(t) = \frac{a_i(t)-1}{N_i(t)}$.

Now, we want to show that for any fixed $a_i(t) \geq 1$ and $b_i(t) \geq 1$, we have

$$\begin{aligned}
\mathbb{P}\left(\theta_i(t) > \hat{\mu}_i(t) + \alpha/2\right) &= 1 - F^{\texttt{beta}}_{(a_i(t),b_i(t))}(\hat{\mu}_i(t) + \alpha/2) \\
&= 1 - \left(1 - F_{a_i(t)+b_i(t)-1,\hat{\mu}_i(t)+\alpha/2}(a_i(t) - 1)\right) \\
&\text{(Using equation (11) in the Appendix)} \\
&= F_{a_i(t)+b_i(t)-1,\hat{\mu}_i(t)+\alpha/2}(\hat{\mu}_i(t)(a_i(t) + b_i(t) - 2)) \\
&\text{(Using the Chernoff-Hoeffding Bound)} \\
&\leq \exp(-(a_i(t) + b_i(t) - 2)d(\hat{\mu}_i(t) + \alpha/2, \hat{\mu}_i(t))) \\
&\leq \exp(-N_i(t)\alpha^2/8)
\end{aligned}$$

Then

$$\begin{aligned}
\sum_{t=1}^{\infty} \mathbb{E}\left(\mathbb{I}\left(\{\theta_i(t) - \hat{\mu}_i(t) > \alpha/2, i \in \mathcal{S}_t\}\right)\right) &\leq \sum_{N_i=0}^{\infty} \exp(-N_i\alpha^2/8) \\
&\leq \frac{1}{1 - \exp\left(-\alpha^2/8\right)}.
\end{aligned}$$

Combining the two cases, we finally obtain $\sum_{t=1}^{\infty} \mathbb{E}\left(\mathbb{I}(\mathcal{A}_i(t))\right) < 1 + \frac{4}{\alpha^2} + \frac{1}{1-\exp(-\alpha^2/8)}$. ■

Lemma B.1.10. *Under the Assumption 1, the TVS sampling policy in Table 2 with* $C = (\sqrt{5} - 1)/2$ *satisfies, for each* $i \in \mathcal{S}^*$ *and for some suitable* $C(\alpha) > 0$,

$$\sum_{t=1}^{\infty} \mathbb{P}\left(\mathcal{A}_i(t)\right) < C(\alpha). \tag{B.35}$$

Proof: For a signal arm $i \in \mathcal{S}^*$ we have $\mathcal{A}_i(t) = \{\theta_i(t) < 0.5\}$. We recall the arguments from Lemma B.1.6 in the Supplement. There we denoted with τ_j^i the j^{th} time the arm i has been pulled (i.e. $\theta_i(t) > 0.5$). Now, we note that in between $[\tau_j^i + 1, \tau_{j+1}^i - 1]$ we have $\{\theta_i(t) < 0.5\}$ since the arm i is not being pulled. One can deduce that

$$\sum_{t=0}^{\infty} \mathbb{E}\left(\mathbb{I}(\mathcal{A}_i(t))\right) = \sum_{j=0}^{\infty} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right). \tag{B.36}$$

Then note that using the result from Lemma B.1.6.

$$\mathbb{E}\left[\tau_{j+1} - \tau_j - 1\right] \leq \begin{cases} 3 + \frac{1}{\alpha} \quad \text{when} \quad j \leq \frac{8}{\alpha} \\ \frac{1}{\mathbf{e}^{\alpha^2 j/4} - 1} + \mathbf{e}^{-\alpha^2 j/2}\left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(j+1)}\right) \quad \text{when} \quad j > \frac{8}{\alpha}. \end{cases} \tag{B.37}$$

From this expression we can immediately see that (B.36) is summable. Below, we provide more details on the upper bound $C(\alpha)$ in (B.35).

$$\sum_{j=0}^{\infty} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right) \leq \sum_{j=0}^{\lceil \frac{8}{\alpha} \rceil} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right) + \sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\lceil \frac{4}{\alpha^2} \rceil} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right) + \sum_{j=\lceil \frac{4}{\alpha^2} \rceil + 1}^{\infty} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right). \tag{B.38}$$

Then we can bound each term separately as: $\sum_{j=0}^{\lceil \frac{8}{\alpha} \rceil} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right) \leq \lceil \frac{8}{\alpha} \rceil (3 + \frac{1}{\alpha})$,

$$\sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\lceil \frac{4}{\alpha^2} \rceil} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right) \leq \lceil \frac{4}{\alpha^2} \rceil \left(\frac{1}{\mathbf{e}^{\alpha^2(\lceil \frac{8}{\alpha} \rceil)/4} - 1} + \mathbf{e}^{-\alpha^2(\lceil \frac{8}{\alpha} \rceil)/2} \left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(\lceil \frac{8}{\alpha} \rceil + 1)} \right) \right),$$

$$\sum_{j=\lceil \frac{4}{\alpha^2} \rceil + 1}^{\infty} \mathbb{E}\left(\tau_{j+1}^i - \tau_j^i - 1\right) \leq \sum_{j=\lceil \frac{4}{\alpha^2} \rceil + 1}^{\infty} \left[\frac{1}{\mathbf{e}^{\alpha^2 j/4} - 1} + \mathbf{e}^{-\alpha^2 j/2} \left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(j+1)} \right) \right]$$

Using the fact that $\exp(-x/2) \geq 1/(\exp(x) - 1)$ for $x \geq 1$

$$\leq \sum_{j=\lceil \frac{4}{\alpha^2} \rceil + 1}^{\infty} \left[\mathbf{e}^{-\alpha^2 j/8} + \mathbf{e}^{-\alpha^2 j/2} \left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(\lceil \frac{4}{\alpha^2} \rceil + 1)} \right) \right]$$

$$\leq \frac{1}{1 - \mathbf{e}^{-\alpha^2/8}} + \frac{1}{1 - \mathbf{e}^{-\alpha^2/2}} \left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(\lceil \frac{4}{\alpha^2} \rceil + 1)} \right).$$

B.2 Additional Simulation Results

B.2.1 Offline Cases

The Figure B.1 below shows simulation results for $p = 1\,000$ under the same settings as Figure 6. In addition, we report convergence diagnostics (number of iterations until convergence) for the simulation study from Section 6.1, using the convergence criterion "$\widehat{S}_t$ stays the same for 100 consecutive TVS iterations", are included in Table B.1.

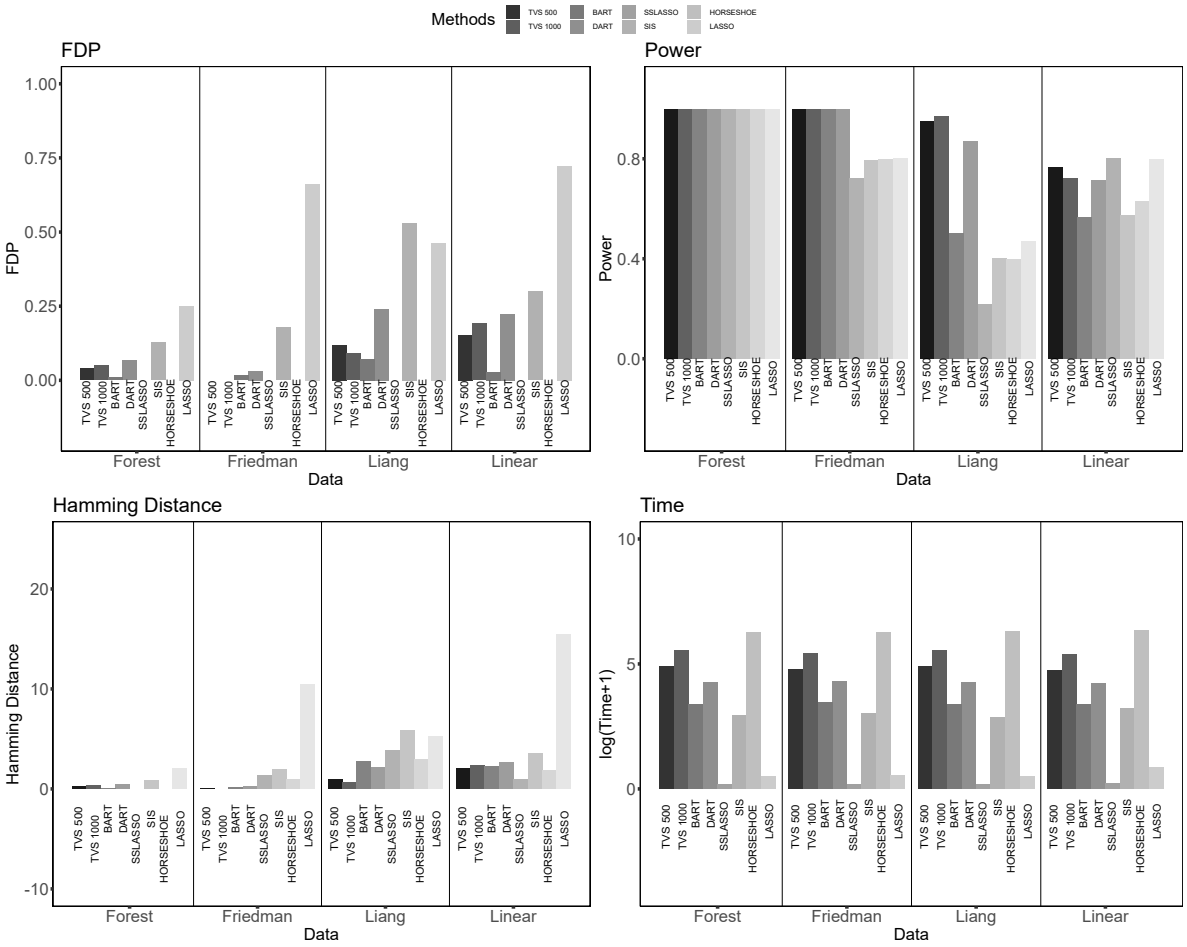


Figure B.1: Graphs denoting FDP , Power , Hamming Distance, and Time for the 4 choices of f_0 assuming $p = 1\,000$ and $n = 300$. The x-axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1\,000$.

In addition to the convergence criterion, we tried different number of trees (D) for DART and BART. We also considered a different variable selection rule, i.e. the Median Probability

p	M	Friedman		Linear		Forest		Liang	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
1000	500	216.40	112.69	216.50	112.68	126.16	31.11	283.58	162.44
	1000	196.34	94.79	197.48	94.20	115.30	11.02	232.06	130.52
10000	500	298.04	100.51	299.58	101.01	230.14	39.74	247.20	156.05
	1000	280.46	82.96	280.28	82.98	205.10	31.86	235.74	109.29

Table B.1: The table records the number of iterations needed for TVS to converge in the simulation study in Section 3.7.1.

Model using the inclusion probability of BART and DART (as mentioned in Linero (2018)). Due to space constraints, we showed only the best settings for BART and DART in Section 3.7.1. The following tables present the entire simulation study and show that TVS yields better Hamming distance and computational speed gains. Tables 5-8 present the offline simulation study, one table for each data setup.

Table B.2: Linear Setup: BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1)

$p = 1000$										
Method	Convergence Criteria	M	Mean Time	SD Time	Mean FDP	SD FDP	Mean Power	SD Power	Mean Ham	SD Ham
TVS	YES	100	30.27	11.14	0.16	0.15	0.76	0.10	2.10	1.04
	NO	100	37.67	5.35	0.22	0.14	0.82	0.09	2.20	1.11
	YES	500	84.02	39.67	0.03	0.10	0.71	0.14	1.64	1.01
	NO	500	115.41	8.44	0.15	0.16	0.77	0.12	2.04	1.18
	YES	1000	147.72	71.70	0.07	0.12	0.66	0.16	2.04	0.90
	NO	1000	213.62	14.82	0.19	0.16	0.72	0.15	2.38	1.23
	Avg Split/ Prob									
DART	Avg Split		38.25	2.17	0.07	0.15	0.52	0.15	2.62	0.97
	Prob		38.25	2.17	0.16	0.17	0.57	0.16	2.80	1.09
	Avg Split		83.47	4.68	0.23	0.19	0.72	0.16	2.78	1.56
	Prob		83.47	4.68	0.32	0.20	0.76	0.16	3.38	2.01
	Avg Split		281.94	34.48	0.42	0.19	0.66	0.17	4.46	1.96
	Prob		281.94	34.48	0.48	0.20	0.69	0.19	5.44	2.59
BART	Avg Split		34.93	3.13	0.03	0.10	0.57	0.15	2.24	0.87
	Prob		34.93	3.13	0.17	0.17	0.70	0.12	2.34	1.22
	Avg Split		83.39	6.19	0.03	0.08	0.66	0.15	1.82	0.83
	Prob		83.39	6.19	0.24	0.15	0.76	0.11	2.60	1.14
	Avg Split		321.77	21.53	0.06	0.11	0.72	0.11	1.68	0.77
	Prob		321.77	21.53	0.39	0.17	0.81	0.07	3.98	1.83
SSLASSO			0.40	0.06	0.00	0.00	0.80	0.10	0.98	0.51
p=10000										
TVS	YES	100	43.18	13.83	0.08	0.16	0.21	0.20	4.22	0.84
	NO	100	54.28	7.37	0.27	0.20	0.49	0.13	3.56	1.15
	YES	500	113.64	33.34	0.15	0.22	0.48	0.17	3.14	1.31
	NO	500	146.72	18.23	0.13	0.20	0.57	0.14	2.70	1.25
	YES	1000	184.65	52.60	0.07	0.15	0.50	0.12	2.74	0.90
	NO	1000	262.24	30.22	0.07	0.15	0.59	0.15	2.38	1.09
	Avg Split/ Prob									
DART	Avg Split		251.30	30.96	0.17	0.23	0.44	0.14	3.30	1.23
	Prob		251.30	30.96	0.24	0.23	0.48	0.16	3.48	1.42
	Avg Split		544.92	79.06	0.23	0.20	0.66	0.15	2.84	1.60
	Prob		544.92	79.06	0.44	0.19	0.70	0.17	4.84	2.48
	Avg Split		1690.91	303.85	0.74	0.09	0.68	0.19	11.74	3.46
	Prob		1690.91	303.85	0.84	0.05	0.70	0.19	21.52	5.68
BART	Avg Split		238.08	54.78	0.00	0.00	0.25	0.15	3.76	0.74
	Prob		238.08	54.78	0.29	0.24	0.48	0.18	3.92	1.41
	Avg Split		542.44	104.17	0.06	0.16	0.37	0.15	3.26	0.90
	Prob		542.44	104.17	0.25	0.21	0.58	0.15	3.32	1.42
	Avg Split		2057.58	368.99	0.01	0.05	0.42	0.15	2.92	0.78
	Prob		2057.58	368.99	0.15	0.18	0.57	0.12	2.82	1.06
SSLASSO			9.59	0.61	0.00	0.00	0.45	0.09	2.74	0.44

Table B.3: Liang Setup: BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1)

					p = 1000					
Method	Convergence Criteria	M	Mean Time	SD Time	Mean FDP	SD FDP	Mean Power	SD Power	Mean Ham	SD Ham
TVS	YES	100	36.24	14.22	0.15	0.15	0.82	0.18	1.80	1.23
	NO	100	41.61	9.80	0.20	0.13	0.87	0.15	1.78	1.23
	YES	500	97.16	44.16	0.08	0.13	0.86	0.17	1.20	1.14
	NO	500	129.34	18.88	0.14	0.12	0.93	0.13	1.18	1.16
	YES	1000	143.95	81.27	0.06	0.14	0.87	0.16	1.04	1.35
	NO	1000	239.32	32.67	0.11	0.15	0.94	0.13	0.96	1.47
	Avg Split/ Prob									
DART	Avg Split		42.08	6.82	0.14	0.18	0.56	0.16	2.74	1.23
	Prob		42.08	6.82	0.26	0.21	0.59	0.16	3.24	1.61
	Avg Split		91.97	14.56	0.24	0.17	0.87	0.15	2.22	1.72
	Prob		91.97	14.56	0.41	0.20	0.90	0.14	4.36	2.83
	Avg Split		285.21	46.55	0.33	0.21	0.91	0.14	3.18	2.69
	Prob		285.21	46.55	0.45	0.21	0.92	0.13	5.22	3.74
BART	Avg Split		37.46	6.25	0.06	0.13	0.50	0.16	2.70	0.99
	Prob		37.46	6.25	0.43	0.18	0.69	0.17	4.68	1.96
	Avg Split		87.44	13.23	0.09	0.15	0.62	0.17	2.28	1.13
	Prob		87.44	13.23	0.44	0.22	0.76	0.15	5.10	2.94
	Avg Split		334.66	50.66	0.01	0.06	0.58	0.16	2.16	0.87
	Prob		334.66	50.66	0.39	0.16	0.79	0.17	3.98	2.06
SSLASSO			0.35	0.05	0.00	0.00	0.22	0.08	3.90	0.42
					p=10000					
TVS	YES	100	36.28	18.21	0.01	0.07	0.12	0.13	4.44	0.67
	NO	100	58.38	7.85	0.02	0.12	0.37	0.09	3.22	0.65
	YES	500	129.30	49.27	0.02	0.07	0.33	0.17	3.42	0.73
	NO	500	160.50	28.76	0.13	0.18	0.52	0.13	2.88	1.08
	YES	1000	215.42	77.68	0.02	0.08	0.41	0.16	3.02	0.80
	NO	1000	281.03	33.14	0.17	0.20	0.62	0.17	2.64	1.55
	Avg Split/ Prob									
DART	Avg Split		264.51	45.73	0.12	0.21	0.33	0.14	3.64	1.01
	Prob		264.51	45.73	0.50	0.24	0.37	0.15	5.34	1.78
	Avg Split		568.41	105.58	0.41	0.24	0.58	0.18	4.44	2.36
	Prob		568.41	105.58	0.65	0.17	0.60	0.19	8.62	3.72
	Avg Split		1713.13	388.03	0.67	0.16	0.68	0.19	9.94	4.49
	Prob		1713.13	388.03	0.81	0.11	0.70	0.19	19.02	7.74
BART	Avg Split		235.79	48.24	0.02	0.10	0.21	0.05	3.98	0.32
	Prob		235.79	48.24	0.59	0.19	0.32	0.12	6.06	1.65
	Avg Split		538.84	103.64	0.05	0.17	0.25	0.09	3.90	0.65
	Prob		538.84	103.64	0.59	0.21	0.38	0.09	6.82	2.67
	Avg Split		2053.60	373.80	0.05	0.13	0.29	0.10	3.66	0.56
	Prob		2053.60	373.80	0.16	0.21	0.40	0.09	3.56	0.91
SSLASSO			2.45	0.22	0.00	0.00	0.20	0.00	4.00	0.00

Table B.4: Friedman Setup: BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1)

					$p = 1000$					
Method	Convergence Criteria	M	Mean Time	SD Time	Mean FDP	SD FDP	Mean Power	SD Power	Mean Ham	SD Ham
TVS	YES	100	31.06	12.07	0.17	0.14	0.77	0.10	2.08	1.03
	NO	100	38.42	6.13	0.23	0.14	0.82	0.10	2.26	1.14
	YES	500	86.28	41.23	0.04	0.11	0.70	0.15	1.72	1.05
	NO	500	119.10	12.98	0.16	0.16	0.77	0.13	2.08	1.21
	YES	1000	150.88	73.61	0.08	0.13	0.65	0.16	2.12	0.92
	NO	1000	221.15	24.51	0.20	0.16	0.71	0.16	2.48	1.22
	Avg Split/ Prob									
DART	Avg Split		41.52	5.41	0.07	0.15	0.52	0.15	2.62	0.97
	Prob		41.52	5.41	0.17	0.17	0.57	0.16	2.80	1.09
	Avg Split		92.19	12.73	0.24	0.18	0.72	0.17	2.82	1.56
	Prob		92.19	12.73	0.33	0.19	0.76	0.17	3.46	1.99
	Avg Split		309.60	48.65	0.43	0.18	0.67	0.17	4.52	1.96
	Prob		309.60	48.65	0.49	0.19	0.69	0.19	5.52	2.56
BART	Avg Split		38.17	5.52	0.03	0.10	0.56	0.15	2.26	0.90
	Prob		38.17	5.52	0.17	0.17	0.70	0.12	2.38	1.28
	Avg Split		91.71	13.07	0.03	0.09	0.66	0.15	1.84	0.84
	Prob		91.71	13.07	0.25	0.15	0.75	0.11	2.68	1.17
	Avg Split		356.32	52.27	0.07	0.11	0.72	0.11	1.72	0.78
	Prob		356.32	52.27	0.39	0.18	0.82	0.08	4.02	1.86
SSLASSO			0.39	0.08	0.00	0.00	0.80	0.10	1.00	0.49
					p=10000					
TVS	YES	100	42.50	13.82	0.08	0.16	0.20	0.20	4.24	0.85
	NO	100	53.41	6.40	0.26	0.20	0.50	0.12	3.50	1.15
	YES	500	113.50	35.77	0.15	0.22	0.48	0.17	3.14	1.31
	NO	500	145.87	15.29	0.13	0.20	0.57	0.14	2.70	1.25
	YES	1000	183.96	56.91	0.08	0.16	0.50	0.12	2.76	0.92
	NO	1000	259.48	26.15	0.08	0.16	0.58	0.16	2.42	1.11
	Avg Split/ Prob									
DART	Avg Split		269.16	39.30	0.17	0.23	0.44	0.14	3.30	1.23
	Prob		269.16	39.30	0.24	0.23	0.48	0.16	3.48	1.42
	Avg Split		590.78	96.73	0.24	0.19	0.66	0.15	2.90	1.58
	Prob		590.78	96.73	0.45	0.19	0.70	0.17	4.92	2.45
	Avg Split		1839.84	382.64	0.74	0.09	0.69	0.19	11.78	3.45
	Prob		1839.84	382.64	0.84	0.05	0.71	0.19	21.58	5.70
BART	Avg Split		250.06	46.46	0.00	0.00	0.24	0.15	3.78	0.76
	Prob		250.06	46.46	0.29	0.24	0.47	0.18	3.94	1.42
	Avg Split		573.10	104.74	0.06	0.16	0.36	0.14	3.30	0.89
	Prob		573.10	104.74	0.26	0.21	0.58	0.15	3.34	1.41
	Avg Split		2211.11	394.91	0.01	0.05	0.42	0.15	2.92	0.78
	Prob		2211.11	394.91	0.14	0.18	0.56	0.12	2.82	1.06
SSLASSO			9.38	0.64	0.00	0.00	0.45	0.09	2.76	0.43

Table B.5: Forest Setup: BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1)

Method	Convergence Criteria	M	Mean Time	SD Time	Mean FDP	SD FDP	Mean Power	SD Power	Mean Ham	SD Ham
					$p = 1000$					
TVS	YES	100	18.34	12.51	0.00	0.02	1.00	0.00	0.02	0.14
	NO	100	39.30	3.53	0.05	0.08	1.00	0.00	0.32	0.51
	YES	500	45.56	34.00	0.01	0.04	1.00	0.00	0.06	0.24
	NO	500	126.55	10.66	0.06	0.09	1.00	0.00	0.36	0.56
	YES	1000	84.82	67.87	0.00	0.02	1.00	0.00	0.02	0.14
	NO	1000	235.24	19.79	0.08	0.10	1.00	0.00	0.50	0.61
	Avg Split/ Prob									
DART	Avg Split		42.52	5.71	0.00	0.00	0.99	0.04	0.04	0.20
	Prob		42.52	5.71	0.00	0.02	1.00	0.00	0.02	0.14
	Avg Split		91.69	12.63	0.07	0.09	1.00	0.00	0.44	0.61
	Prob		91.69	12.63	0.17	0.15	1.00	0.00	1.24	1.20
	Avg Split		302.68	51.01	0.33	0.14	1.00	0.00	2.78	1.59
	Prob		302.68	51.01	0.42	0.14	1.00	0.00	4.14	2.00
BART	Avg Split		38.90	6.03	0.01	0.05	1.00	0.00	0.06	0.31
	Prob		38.90	6.03	0.24	0.15	1.00	0.00	1.88	1.52
	Avg Split		92.01	13.74	0.02	0.06	1.00	0.00	0.14	0.35
	Prob		92.01	13.74	0.39	0.17	1.00	0.00	3.94	2.82
	Avg Split		355.26	55.65	0.03	0.06	1.00	0.00	0.18	0.39
	Prob		355.26	55.65	0.52	0.11	1.00	0.00	5.86	2.33
SSLASSO			0.36	0.08	0.00	0.00	1.00	0.00	0.00	0.00
					$p = 10\,000$					
TVS	YES	100	38.14	10.73	0.00	0.00	0.78	0.34	1.12	1.72
	NO	100	57.82	7.10	0.00	0.00	0.98	0.07	0.12	0.33
	YES	500	85.29	16.72	0.00	0.00	0.96	0.09	0.18	0.44
	NO	500	158.70	17.03	0.00	0.02	0.99	0.04	0.06	0.24
	YES	1000	143.99	41.27	0.00	0.00	0.97	0.08	0.14	0.40
	NO	1000	282.44	30.03	0.01	0.05	1.00	0.03	0.10	0.36
	Avg Split/ Prob									
DART	Avg Split		262.89	38.21	0.00	0.00	0.94	0.10	0.28	0.50
	Prob		262.89	38.21	0.02	0.06	0.98	0.06	0.22	0.46
	Avg Split		585.06	100.00	0.15	0.14	1.00	0.00	1.10	1.15
	Prob		585.06	100.00	0.41	0.13	1.00	0.00	3.82	1.79
	Avg Split		1868.72	308.73	0.69	0.09	1.00	0.00	12.00	3.81
	Prob		1868.72	308.73	0.81	0.05	1.00	0.00	22.84	6.44
BART	Avg Split		251.37	53.89	0.00	0.00	0.76	0.17	1.20	0.86
	Prob		251.37	53.89	0.28	0.17	0.99	0.05	2.34	1.85
	Avg Split		571.27	115.64	0.01	0.05	0.95	0.12	0.32	0.62
	Prob		571.27	115.64	0.38	0.15	1.00	0.00	3.54	2.06
	Avg Split		2206.52	426.63	0.00	0.00	1.00	0.03	0.02	0.14
	Prob		2206.52	426.63	0.09	0.12	1.00	0.00	0.62	0.85
SSLASSO			2.99	0.72	0.00	0.00	0.90	0.15	0.52	0.74

B.2.2 Online Cases

In Table B.6, we report convergence diagnostics of the simulation from Section 6.2, where the convergence criterion is chosen as "$\widehat{S}_t$ stays the same for 100 consecutive TVS iterations". Table 10,12,14 and 16 report the results with 10 000, comparing TVS with BART. These results show that TVS at the very least highly competitive with DART in terms of Hamming distance but offers vast computational benefits compared to BART and DART. Tables 11, 13,15 and 17 report TVS results with $n = 50\,000$ and $n = 100\,000$. We could not run BART on these large datasets and thereby we report only on TVS.

n	M	s	Friedman		Forest		Linear		Liang	
			Mean	SD	Mean	SD	Mean	SD	Mean	SD
10 000	500	500	9.40	2.74	6.56	1.16	9.40	2.74	10.00	1.60
	1000	500	8.32	1.32	6.06	0.24	8.32	1.32	8.54	1.31
	500	1000	23.46	7.99	13.64	2.94	23.46	7.99	18.14	3.83
	1000	100	21.14	7.15	12.44	2.15	21.14	7.15	15.58	2.92
50 000	500	500	2.42	0.76	2.00	0.00	2.42	0.76	2.28	0.45
	1000	500	2.14	0.35	2.00	0.00	2.14	0.35	2.12	0.33
	500	1000	4.94	1.75	3.10	0.36	4.94	1.75	4.04	0.83
	1000	100	4.10	1.25	3.08	0.27	4.10	1.25	3.48	0.71
100 000	500	500	1.20	0.45	1.00	0.00	1.22	0.46	1.38	0.49
	1000	500	1.18	0.39	1.00	0.00	1.18	0.39	1.10	0.30
	500	1000	2.72	0.73	2.04	0.20	2.74	0.72	2.24	0.48
	1000	100	2.66	0.87	2.04	0.20	2.64	0.88	2.14	0.35

Table B.6: The table records the number of rounds needed for TVS to converge in the simulation study in Section 3.7.2.

In addition to the results shown in Section 6.2, we tried a different number of trees D for DART and BART. We also considered a different variable selection rule, i.e. the Median

Probability Model using the inclusion probability of BART and DART (as mentioned in [144]). Due to space constraints, we showed only the best settings for BART and DART in Section 3.7.2. Now we present additional simulation results for $n = 50\,000$ and $n = 100\,000$. Since BART and DART cannot be run with such large n, we only show the results for TVS.

Table B.7: Linear Setup with $n = 10\,000$, BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1); s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	52.80	12.24	0.00	0.02	1.00	0.00	0.02	0.14
	500	500	1	8.33	1.21	0.00	0.00	0.96	0.09	0.18	0.44
	500	500	5	40.29	5.31	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	79.77	10.26	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	88.68	11.37	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	14.79	1.95	0.00	0.00	0.98	0.05	0.08	0.27
	1000	500	5	73.30	9.15	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	145.98	18.12	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	91.11	23.10	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	6.42	1.01	0.03	0.09	0.69	0.19	1.68	1.00
	500	1000	5	33.02	5.05	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	10	66.82	9.69	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	Till Converge	152.94	33.50	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	1	11.15	1.61	0.02	0.05	0.83	0.15	0.94	0.77
	1000	1000	5	60.73	8.66	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	10	122.70	16.98	0.00	0.02	1.00	0.00	0.02	0.14
	D	Avg Split /Prob									
DART	10	Avg Split		67.86	11.51	0.00	0.02	0.98	0.06	0.12	0.33
	10	Prob		196.43	43.41	0.00	0.00	1.00	0.00	0.00	0.00
	50	Avg Split		12.77	2.04	0.15	0.25	0.45	0.18	3.30	1.49
	50	Prob		196.80	43.34	0.00	0.00	1.00	0.00	0.00	0.00
	200	Avg Split		61.14	9.85	0.00	0.00	0.94	0.10	0.30	0.51
	200	Prob		196.89	43.85	0.00	0.00	1.00	0.00	0.00	0.00
BART	10	Avg Split		125.31	19.75	0.00	0.02	0.98	0.05	0.10	0.30
	10	Prob		949.38	128.93	0.63	0.06	1.00	0.00	9.00	2.18
	50	Avg Split		1715.54	218.52	0.37	0.13	1.00	0.00	3.32	1.66
	50	Prob		1715.54	218.52	0.66	0.06	1.00	0.00	10.26	2.95
	200	Avg Split		5717.13	529.43	0.29	0.14	1.00	0.00	2.36	1.43
	200	Prob		5717.13	529.43	0.73	0.06	1.00	0.00	14.38	3.76
SSLASSO				9.40	0.93	0.00	0.02	1.00	0.00	0.02	0.14
Alpha				4.98	0.17	0.04	0.09	1.00	0.00	0.28	0.61

Table B.8: Linear Setup with $n = 50\,000$ and $n = 100\,000$; s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

						$n = 50\,000$					
Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	89.20	31.75	0.00	0.00	0.82	0.07	0.92	0.34
	500	500	1	37.34	5.51	0.04	0.08	0.76	0.11	1.38	0.64
	500	500	5	181.80	26.01	0.00	0.00	0.88	0.10	0.60	0.49
	500	500	10	364.25	51.93	0.00	0.00	0.96	0.08	0.18	0.39
	1000	500	Till Converge	140.49	31.04	0.00	0.00	0.80	0.03	0.98	0.14
	1000	500	1	66.15	9.06	0.00	0.00	0.76	0.08	1.22	0.42
	1000	500	5	326.66	43.50	0.00	0.00	0.88	0.10	0.60	0.49
	1000	500	10	656.69	85.25	0.00	0.00	0.97	0.07	0.16	0.37
	500	1000	Till Converge	150.23	63.39	0.09	0.08	0.96	0.08	0.70	0.46
	500	1000	1	29.93	4.74	0.08	0.11	0.76	0.13	1.56	0.76
	500	1000	5	151.28	21.95	0.09	0.08	0.98	0.05	0.60	0.49
	500	1000	10	304.49	43.17	0.14	0.07	1.00	0.03	0.84	0.42
	1000	1000	Till Converge	220.03	65.18	0.02	0.05	0.96	0.08	0.30	0.46
	1000	1000	1	53.29	7.90	0.04	0.08	0.81	0.14	1.14	0.67
	1000	1000	5	270.67	37.17	0.02	0.05	0.99	0.04	0.14	0.35
	1000	1000	10	545.72	72.54	0.02	0.06	1.00	0.00	0.14	0.35
						$n = 100\,000$					
TVS	500	500	Till Converge	93.14	39.67	0.00	0.02	0.82	0.07	0.92	0.34
	500	500	1	76.60	12.54	0.00	0.02	0.81	0.06	0.96	0.28
	500	500	5	379.48	58.92	0.00	0.00	0.99	0.05	0.06	0.24
	500	500	10	764.01	116.22	0.00	0.00	1.00	0.03	0.02	0.14
	1000	500	Till Converge	159.10	55.08	0.00	0.00	0.80	0.06	1.00	0.29
	1000	500	1	135.46	19.98	0.00	0.00	0.79	0.04	1.04	0.20
	1000	500	5	677.18	98.13	0.00	0.00	0.95	0.09	0.26	0.44
	1000	500	10	1363.99	200.85	0.00	0.00	0.99	0.05	0.06	0.24
	500	1000	Till Converge	173.36	60.13	0.07	0.08	0.97	0.07	0.58	0.57
	500	1000	1	62.23	10.77	0.06	0.09	0.84	0.13	1.12	0.77
	500	1000	5	315.53	51.68	0.13	0.07	1.00	0.00	0.78	0.42
	500	1000	10	635.09	102.95	0.15	0.05	1.00	0.00	0.92	0.27
	1000	1000	Till Converge	293.65	96.59	0.01	0.04	0.98	0.05	0.14	0.35
	1000	1000	1	111.14	18.43	0.01	0.05	0.90	0.11	0.60	0.61
	1000	1000	5	566.08	85.57	0.03	0.06	1.00	0.00	0.18	0.39
	1000	1000	10	1140.65	166.47	0.09	0.08	1.00	0.00	0.52	0.50

Table B.9: Liang Setup with $n = 10\,000$, BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1); s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	80.68	18.66	0.00	0.02	0.99	0.04	0.06	0.24
	500	500	1	9.16	1.55	0.02	0.08	0.38	0.14	3.12	0.75
	500	500	5	39.81	6.38	0.00	0.00	0.92	0.11	0.40	0.57
	500	500	10	79.97	12.62	0.00	0.00	0.99	0.05	0.06	0.24
	1000	500	Till Converge	125.43	30.63	0.00	0.00	0.99	0.05	0.06	0.24
	1000	500	1	15.31	2.49	0.01	0.07	0.53	0.18	2.40	0.93
	1000	500	5	72.52	10.90	0.00	0.00	0.97	0.07	0.14	0.35
	1000	500	10	146.75	21.39	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	124.65	37.15	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	7.53	1.33	0.23	0.25	0.34	0.15	4.04	1.14
	500	1000	5	33.59	5.93	0.00	0.02	0.84	0.14	0.80	0.70
	500	1000	10	67.86	11.51	0.00	0.02	0.98	0.06	0.12	0.33
	1000	1000	Till Converge	196.43	43.41	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	1	12.77	2.04	0.15	0.25	0.45	0.18	3.30	1.49
	1000	1000	5	61.14	9.85	0.00	0.00	0.94	0.10	0.30	0.51
	1000	1000	10	125.31	19.75	0.00	0.02	0.98	0.05	0.10	0.30
	D	Avg Split /Prob									
DART	10	Avg Split		965.45	151.88	0.85	0.05	0.58	0.18	19.10	3.31
	10	Prob		965.45	151.88	0.87	0.04	0.58	0.18	22.90	4.07
	50	Avg Split		1647.39	110.13	0.55	0.18	0.98	0.05	7.76	4.63
	50	Prob		1647.39	110.13	0.67	0.14	0.98	0.05	11.86	5.27
	200	Avg Split		4695.72	626.39	0.27	0.19	1.00	0.00	2.48	2.60
	200	Prob		4695.72	626.39	0.42	0.20	1.00	0.00	5.18	5.29
BART	10	Avg Split		1525.58	221.31	0.71	0.07	1.00	0.00	13.22	3.77
	10	Prob		1525.58	221.31	0.91	0.01	1.00	0.00	48.66	6.26
	50	Avg Split		2315.53	294.10	0.71	0.07	1.00	0.00	13.34	4.35
	50	Prob		2315.53	294.10	0.91	0.01	1.00	0.00	52.74	9.04
	200	Avg Split		5801.02	605.46	0.74	0.05	1.00	0.00	15.34	3.92
	200	Prob		5801.02	605.46	0.90	0.01	1.00	0.00	46.92	6.89
SSLASSO				16.14	1.64	0.00	0.00	0.60	0.00	2.00	0.00
Alpha				4.91	0.66	0.00	0.00	0.62	0.05	1.92	0.27

Table B.10: Liang Setup with $n = 50\,000$ and $n = 100\,000$; s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

n = 50000											
Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	92.32	20.87	0.00	0.00	0.99	0.05	0.06	0.24
	500	500	1	40.68	6.15	0.00	0.00	0.90	0.13	0.52	0.65
	500	500	5	203.82	29.86	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	410.74	59.11	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	157.52	33.75	0.00	0.00	0.98	0.07	0.08	0.34
	1000	500	1	73.48	10.50	0.00	0.00	0.96	0.10	0.20	0.49
	1000	500	5	376.54	51.66	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	759.76	103.03	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	139.35	33.68	0.00	0.00	0.99	0.05	0.06	0.24
	500	1000	1	33.99	5.70	0.00	0.00	0.82	0.16	0.88	0.82
	500	1000	5	172.82	26.65	0.00	0.02	1.00	0.03	0.04	0.20
	500	1000	10	348.24	53.31	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	Till Converge	221.64	57.18	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	1	61.49	10.03	0.00	0.00	0.96	0.08	0.20	0.40
	1000	1000	5	320.88	47.84	0.01	0.03	1.00	0.00	0.04	0.20
	1000	1000	10	649.88	92.55	0.00	0.02	1.00	0.00	0.02	0.14
n = 100000											
TVS	500	500	Till Converge	120.82	49.24	0.00	0.00	0.99	0.04	0.04	0.20
	500	500	1	86.77	13.57	0.00	0.00	0.99	0.05	0.06	0.24
	500	500	5	438.57	66.16	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	883.47	129.33	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	175.25	59.12	0.00	0.00	0.99	0.04	0.04	0.20
	1000	500	1	158.25	22.10	0.00	0.00	0.99	0.04	0.04	0.20
	1000	500	5	807.36	109.70	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	1625.51	219.19	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	167.40	46.18	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	74.04	11.86	0.01	0.03	0.98	0.06	0.14	0.35
	500	1000	5	374.54	56.43	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	10	752.86	112.17	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	Till Converge	293.39	68.52	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	1	134.11	20.44	0.00	0.00	0.98	0.05	0.08	0.27
	1000	1000	5	690.95	96.24	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	10	1380.91	192.17	0.00	0.00	1.00	0.00	0.00	0.00

Table B.11: Friedman Setup with $n = 10\,000$, BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1); s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	46.26	6.71	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	1	7.42	0.50	0.00	0.00	0.94	0.10	0.32	0.51
	500	500	5	35.04	2.46	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	69.44	5.05	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	80.81	6.85	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	13.08	1.01	0.00	0.00	0.98	0.06	0.10	0.30
	1000	500	5	65.29	4.65	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	129.90	9.27	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	110.23	41.32	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	6.46	0.50	0.07	0.12	0.86	0.15	1.08	1.16
	500	1000	5	32.55	2.33	0.01	0.04	1.00	0.00	0.06	0.24
	500	1000	10	65.30	4.27	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	Till Converge	181.50	50.55	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	1	11.51	0.87	0.04	0.08	0.94	0.11	0.56	0.70
	1000	1000	5	60.74	4.40	0.00	0.02	1.00	0.03	0.04	0.20
	1000	1000	10	122.46	8.51	0.00	0.00	1.00	0.00	0.00	0.00
	D	Avg Split /Prob									
DART	10	Avg Split		1022.94	194.68	0.72	0.16	0.89	0.12	15.62	7.70
	10	Prob		1022.94	194.68	0.80	0.08	0.90	0.12	21.00	7.48
	50	Avg Split		1442.99	136.58	0.02	0.05	1.00	0.00	0.12	0.33
	50	Prob		1442.99	136.58	0.08	0.14	1.00	0.00	0.66	1.61
	200	Avg Split		4939.07	788.75	0.00	0.00	1.00	0.00	0.00	0.00
	200	Prob		4939.07	788.75	0.01	0.05	1.00	0.00	0.06	0.31
BART	10	Avg Split		1751.38	274.31	0.80	0.06	1.00	0.00	21.70	6.53
	10	Prob		1751.38	274.31	0.89	0.02	1.00	0.00	43.30	9.50
	50	Avg Split		1861.83	229.61	0.24	0.13	1.00	0.00	1.80	1.20
	50	Prob		1861.83	229.61	0.57	0.11	1.00	0.00	7.44	3.47
	200	Avg Split		6009.07	892.87	0.21	0.14	1.00	0.00	1.56	1.26
	200	Prob		6009.07	892.87	0.70	0.08	1.00	0.00	12.54	4.08
SSLASSO				16.75	2.62	0.00	0.00	0.80	0.00	1.00	0.00
Alpha				4.96	0.91	0.03	0.08	0.81	0.05	1.10	0.51

Table B.12: Friedman Setup with $n = 50\,000$ and $n = 100\,000$; s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

						n = 50000					
Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	71.12	8.71	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	1	35.49	3.52	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	5	174.43	16.98	0.00	0.02	1.00	0.00	0.02	0.14
	500	500	10	347.30	35.24	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	130.57	13.44	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	65.60	6.82	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	5	324.83	33.21	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	647.04	65.02	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	122.22	45.41	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	32.26	3.27	0.01	0.03	1.00	0.00	0.04	0.20
	500	1000	5	162.98	16.85	0.00	0.02	1.00	0.00	0.02	0.14
	500	1000	10	325.19	32.44	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	Till Converge	227.68	73.92	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	1	60.99	7.37	0.00	0.02	1.00	0.03	0.04	0.20
	1000	1000	5	307.25	31.56	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	10	612.30	60.46	0.00	0.02	1.00	0.00	0.02	0.14
						n = 100000					
TVS	500	500	Till Converge	70.27	6.16	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	1	70.27	6.16	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	5	344.26	28.89	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	680.12	48.41	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	127.92	9.35	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	127.92	9.35	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	5	635.11	46.23	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	1266.57	94.32	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	160.84	45.59	0.01	0.05	1.00	0.00	0.08	0.27
	500	1000	1	63.31	4.11	0.01	0.03	1.00	0.00	0.04	0.20
	500	1000	5	316.52	20.79	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	10	633.90	37.48	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	Till Converge	272.04	68.80	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	1	119.85	7.15	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	5	599.73	35.98	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	10	1192.89	68.52	0.00	0.02	1.00	0.00	0.02	0.14

Table B.13: Forest Setup with $n = 10\,000$, BART and DART are implemented using `Prob` (median probability model) rule or `Avg Split` (truncating the importance measure at 1); s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	52.80	12.24	0.00	0.02	1.00	0.00	0.02	0.14
	500	500	1	8.33	1.21	0.00	0.00	0.96	0.09	0.18	0.44
	500	500	5	40.29	5.31	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	79.77	10.26	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	88.68	11.37	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	14.79	1.95	0.00	0.00	0.98	0.05	0.08	0.27
	1000	500	5	73.30	9.15	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	145.98	18.12	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	91.11	23.10	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	6.42	1.01	0.03	0.09	0.69	0.19	1.68	1.00
	500	1000	5	33.02	5.05	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	10	66.82	9.69	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	Till Converge	152.94	33.50	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	1	11.15	1.61	0.02	0.05	0.83	0.15	0.94	0.77
	1000	1000	5	60.73	8.66	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	10	122.70	16.98	0.00	0.02	1.00	0.00	0.02	0.14
	D	Avg Split /Prob									
DART	10	Avg Split		509.13	65.73	0.13	0.13	0.93	0.10	1.14	1.14
	10	Prob		509.13	65.73	0.14	0.14	0.93	0.10	1.22	1.17
	50	Avg Split		1132.98	190.56	0.16	0.16	1.00	0.00	1.18	1.34
	50	Prob		1132.98	190.56	0.32	0.16	1.00	0.00	2.78	1.80
	200	Avg Split		4028.27	931.10	0.59	0.09	1.00	0.00	7.76	2.92
	200	Prob		4028.27	931.10	0.68	0.07	1.00	0.00	11.18	3.35
BART	10	Avg Split		1099.59	270.16	0.34	0.16	1.00	0.00	3.08	1.91
	10	Prob		1099.59	270.16	0.68	0.05	1.00	0.00	11.14	2.60
	50	Avg Split		1857.85	280.43	0.47	0.15	1.00	0.00	5.08	2.30
	50	Prob		1857.85	280.43	0.82	0.03	1.00	0.00	23.26	3.86
	200	Avg Split		5620.33	847.79	0.83	0.03	1.00	0.00	25.50	4.81
	200	Prob		5620.33	847.79	0.91	0.01	1.00	0.00	53.72	6.32
SSLASSO				1.26	0.16	0.00	0.00	1.00	0.00	0.00	0.00
Alpha				5.03	1.08	0.11	0.13	1.00	0.00	0.78	1.02

Table B.14: Forest Setup with $n = 50\,000$ and $n = 100\,000$; s is the batch size, r is the number of rounds and M is the number of internal MCMC iterations in TVS.

						$n = 50\,000$					
Method	M	s	r	Mean Time	SD Time	Mean FDR	SD FDR	Mean Power	SD Power	Mean Ham	SD Ham
TVS	500	500	Till Converge	90.28	13.28	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	1	45.60	6.80	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	5	224.06	32.44	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	446.61	63.34	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	163.70	22.66	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	82.49	11.65	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	5	408.11	55.52	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	812.39	111.83	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	117.99	23.19	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	37.73	6.62	0.00	0.02	0.99	0.04	0.06	0.24
	500	1000	5	189.66	30.05	0.00	0.02	1.00	0.00	0.02	0.14
	500	1000	10	377.06	60.06	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	Till Converge	211.24	31.96	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	1	68.27	11.64	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	5	341.68	54.73	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	10	676.40	105.33	0.00	0.02	1.00	0.00	0.02	0.14
						$n = 100\,000$					
TVS	500	500	Till Converge	81.54	11.43	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	1	81.54	11.43	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	5	398.59	49.39	0.00	0.00	1.00	0.00	0.00	0.00
	500	500	10	796.94	98.24	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	Till Converge	146.05	17.39	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	1	146.05	17.39	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	5	718.56	79.50	0.00	0.00	1.00	0.00	0.00	0.00
	1000	500	10	1457.97	130.28	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	Till Converge	142.28	20.80	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	1	69.61	8.27	0.00	0.02	1.00	0.00	0.02	0.14
	500	1000	5	348.10	37.10	0.00	0.00	1.00	0.00	0.00	0.00
	500	1000	10	692.50	72.61	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	Till Converge	256.28	40.84	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	1	125.82	14.09	0.00	0.02	1.00	0.00	0.02	0.14
	1000	1000	5	622.97	59.21	0.00	0.00	1.00	0.00	0.00	0.00
	1000	1000	10	1251.07	133.65	0.00	0.00	1.00	0.00	0.00	0.00

B.3 Additional Results for the HIV Dataset

In this section, we show additional results on the analysis of the HIV dataset. First, we present some basic statistics about the data. The entire data comes from Stanford HIV Drug Resistance Database. The raw data can be downloaded from published cleaning codes which we adopt. We provide a basic overview of the dataset in Table A.2.

HIV Virus Life Cycle	Drug Class	Mean Log DS	Number of Mutations	Number of Samples
PI	APV	0.75	201	767
	ATV	1.59	147	328
	IDV	1.33	206	825
	LPV	1.74	184	515
	NFV	2.00	207	842
	RTV	1.72	205	793
	SQV	1.22	206	824
NRTI	X3TC	3.10	283	629
	ABC	1.14	283	623
	AZT	1.55	283	626
	D4T	0.43	281	625
	DDI	0.43	283	628
	TDF	0.22	215	351
NNRTI	DLV	0.98	305	730
	EFV	1.08	312	732
	NVP	1.80	313	744

Table B.15: Basic summary statistics of the HIV dataset. DS refers to the decrease in susceptibility of the drug once the mutations has occurred.

In Section 3.8.1, we illustrated TVS on only the drug LPV. Here, we present the rest of the results. As is done in [8], we record both the number of verified positions discovered (True Discoveries) and the number of discovered unverified positions (False Discoveries) for each of the five methods.

Table B.16: PI Drugs

Methods	Knockoff	LASSO	DART	SSLASSO	TVS
		APV			
True Discoveries	19	20	16	8	18
False Discoveries	3	3	0	0	1
		ATV			
True Discoveries	22	29	19	6	20
False Discoveries	8	20	0	0	0
		IDV			
True Discoveries	19	28	18	5	17
False Discoveries	12	24	3	0	4
		LPV			
True Discoveries	16	30	15	4	17
False Discoveries	1	22	1	0	1
		NFV			
True Discoveries	21	23	19	6	20
False Discoveries	2	2	2	0	3
		RTV			
True Discoveries	19	34	16	4	18
False Discoveries	8	29	4	0	3
		SQV			
True Discoveries	17	22	17	5	17
False Discoveries	4	8	1	0	2

Table B.17: NRTI Drugs

Methods	Knockoff	LASSO	DART	SSLASSO	TVS
		ABC			
True Discoveries	10	15	10	11	14
False Discoveries	1	7	4	3	6
		AZT			
True Discoveries	16	23	15	5	18
False Discoveries	4	51	5	0	8
		D4T			
True Discoveries	6	13	13	12	13
False Discoveries	1	2	5	4	6
		DDI			
True Discoveries	0	23	11	17	12
False Discoveries	0	34	6	7	6
		TDF			
True Discoveries	0	13	14	7	12
False Discoveries	0	9	7	1	6
		X3TC			
True Discoveries	0	14	11	1	11
False Discoveries	0	8	4	0	6

Table B.18: NNRTI Drugs

Methods	Knockoff	LASSO	DART	SSLASSO	TVS
		DLV			
True Discoveries	10	11	9	3	9
False Discoveries	14	31	6	1	9
		EFV			
True Discoveries	11	14	10	5	11
False Discoveries	11	83	5	0	6
		NVP			
True Discoveries	7	14	8	6	9
False Discoveries	10	87	6	0	11

B.4 Details ont the Marketing Data

Summary of Predictor Variables

Name	Indicator	Mean	Sd	Max	Min	Skewness
mail indicator	1	0.50	NA	NA	NA	NA
largest sale amount	0	169.02	286.77	7999.99	-3999.99	4.63
count of product categories that make up 20% or more of total sales	0	2.15	0.37	4.00	2.00	2.10
number of months since first esp or first esp return	0	30.14	16.05	60.00	2.00	0.17
number of months since most recent esp purchase or return	0	23.00	14.22	60.00	2.00	0.60
day of first purchase: weekend day	1	0.37	NA	NA	NA	NA
day of first purchase: weekday day	1	0.63	NA	NA	NA	NA
total number of sales in previous 12 months	0	5.09	4.76	189.00	2.00	5.79
total number of sales in previous 24 months	0	5.95	6.16	270.00	2.00	6.23
total number of sales in previous 36 months	0	6.53	7.10	344.00	2.00	6.44
total number of large ticket items in previous 12 months	0	2.37	0.94	21.00	2.00	6.95
total number of large ticket items in previous 24 months	0	2.50	1.08	32.00	2.00	5.79
total number of large ticket items in previous 36 months	0	2.58	1.24	46.00	2.00	6.94
total number of large ticket items in previous 60 months	0	2.80	1.47	50.00	-1.00	5.37
total number of medium ticket items in previous 12 months	0	3.06	1.92	46.00	2.00	4.62
total number of medium ticket items in previous 24 months	0	3.43	2.49	81.00	2.00	5.10
total number of medium ticket items in previous 36 months	0	3.71	2.90	124.00	2.00	5.48
total number of medium ticket items in previous 60 months	0	4.36	3.68	198.00	2.00	6.39
total number of small ticket items in previous 12 months	0	4.49	7.83	507.00	2.00	34.98
total number of small ticket items in previous 24 months	0	5.14	10.18	667.00	2.00	29.59
total number of small ticket items in previous 36 months	0	5.54	10.73	667.00	2.00	25.12
total number of small ticket items in previous 60 months	0	6.55	12.73	673.00	2.00	20.36
total sales amount in previous 12 months	0	524.90	766.12	18380.40	-437.80	4.31
total sales amount in previous 24 months	0	650.24	935.70	33664.26	-250.01	4.36
count of unique categories in previous 12 months	0	2.94	1.28	12.00	2.00	1.78
count of unique categories in previous 24 months	0	3.15	1.47	12.00	2.00	1.62
count of unique class numbers in previous 12 months	0	4.13	2.84	55.00	2.00	3.00
count of unique class numbers in previous 24 months	0	4.74	3.58	60.00	2.00	3.00
percent gift cards category sales of total sales	0	0.09	0.13	1.03	0.00	3.22
percent home ins category sales of total sales	0	0.02	0.05	0.42	0.00	4.48
percent imaging category sales of total sales	0	0.29	0.28	1.00	-0.20	1.08
percent mobile category sales of total sales	0	0.35	0.26	1.00	-0.03	0.60
percent music category sales of total sales	0	0.18	0.21	1.00	0.00	1.72
percent other category sales of total sales	0	0.36	0.33	0.99	0.01	0.63
percent pc hardware category sales of total sales	0	0.50	0.30	1.49	-2.88	-0.02
percent pst category sales of total sales	0	0.17	0.19	1.11	-0.49	1.89
percent tv category sales of total sales	0	0.40	0.29	1.00	-0.03	0.38
percent vcr category sales of total sales	0	0.29	0.25	3.13	-0.16	1.04
percent wireless category sales of total sales	0	0.24	0.24	1.45	-0.06	1.26
percent audio category sales of total sales	0	0.24	0.24	1.12	-0.08	1.17
percent dss category sales of total sales	0	0.32	0.28	1.00	0.00	0.94
largest return amount	0	176.57	273.43	4999.99	-2699.99	4.14
number of months since oldest return	0	26.20	15.33	56.00	2.00	0.28
number of months since most recent return	0	20.61	13.93	56.00	2.00	0.67
number of distinct merchandise classes returned	0	2.35	0.78	11.00	2.00	3.61
total return amount in previous 12 months	0	301.28	530.07	24926.85	0.01	10.03
total return amount in previous 24 months	0	326.14	584.04	31826.61	0.01	12.74
total number of items returned in previous 12 months	0	3.16	2.87	100.00	2.00	17.05
total number of items returned in previous 24 months	0	3.40	5.50	527.00	2.00	63.78
number of months shopped once in previous 12 months	0	2.59	1.07	12.00	2.00	2.89
number of months shopped once in previous 24 months	0	3.01	1.67	24.00	2.00	3.19
count of unique purchase trips in previous 12 months	0	2.95	1.97	81.00	2.00	8.23
count of unique purchase trips in previous 24 months	0	3.41	2.74	126.00	2.00	8.76
total number of items purchased in previous 12 months	0	5.23	7.24	513.00	2.00	28.99
total number of items purchased in previous 24 months	0	6.17	9.55	672.00	2.00	24.33
total number of weekday items in previous 12 months	0	4.65	6.68	506.00	2.00	35.47

Name	Indicator	Mean	Sd	Max	Min	Skewness
total number of weekend items in previous 12 months	0	4.15	5.18	403.00	2.00	40.58
total number of weekend items in previous 24 months	0	4.62	6.76	504.00	2.00	34.99
total christmas sales amount in previous 12 months	0	379.85	560.04	9877.93	-660.00	4.33
total christmas sales amount in previous 24 months	0	430.10	611.41	10500.09	-660.00	4.05
total christmas items in previous 12 months	0	4.13	6.63	506.00	2.00	44.49
total christmas items in previous 24 months	0	4.46	7.03	506.00	2.00	41.25
total amount of back to school sales in previous 12 months	0	396.37	619.21	10159.37	-150.00	4.23
total amount of back to school sales in previous 24 months	0	416.13	623.76	10159.37	-485.02	3.93
total amount of graduation sales in previous 12 months	0	376.62	564.66	13670.70	-200.00	5.34
total amount of graduation sales in previous 24 months	0	405.84	585.14	13678.65	-252.00	4.33
total spring sales amount in previous 12 months	0	381.35	567.06	13190.30	-308.90	4.63
total spring sales amount in previous 24 months	0	421.00	612.91	13190.30	-340.00	4.31
total summer sales amount in previous 12 months	0	407.91	623.00	13205.62	-300.00	4.51
total summer sales amount in previous 24 months	0	439.29	643.85	17671.37	-300.00	4.23
total autumn sales amount in previous 12 months	0	401.56	616.92	10650.45	-437.80	4.12
total autumn sales amount in previous 24 months	0	447.43	663.93	11819.31	-372.00	4.05
total winter sales amount in previous 24 months	0	453.60	645.40	13921.45	-189.99	4.02
total spring items in previous 12 months	0	4.02	4.72	325.00	2.00	30.25
total spring items in previous 24 months	0	4.35	5.95	400.00	2.00	28.17
total summer items in previous 12 months	0	4.44	4.79	362.00	2.00	32.61
total summer items in previous 24 months	0	4.69	7.63	504.00	2.00	38.59
total autumn items in previous 12 months	0	4.20	7.47	501.00	2.00	41.44
total autumn items in previous 24 months	0	4.46	7.51	501.00	2.00	42.06
total winter items in previous 12 months	0	4.12	5.76	506.00	2.00	48.15
total winter items in previous 24 months	0	4.54	6.97	506.00	2.00	38.14
total count of back to school items in previous 12 months	0	4.29	5.38	362.00	2.00	38.67
total count of back to school items in previous 24 months	0	4.41	7.12	502.00	2.00	36.16
total count of graduation items in previous 12 months	0	4.15	3.83	190.00	2.00	14.61
total count of graduation items in previous 24 months	0	4.41	6.34	504.00	2.00	37.55
total number of net instore esps in previous 12 months	0	2.60	1.17	17.00	2.00	3.74
total number of net instore esps in previous 24 months	0	2.81	1.45	35.00	2.00	4.48
total number of net instore esps lifetime	0	3.14	1.98	43.00	2.00	4.09
avg term of all esps in previous 12 months	0	25.32	14.48	120.00	0.48	0.89
total number of returned instore esps in previous 12 months	0	2.55	1.19	14.00	2.00	4.00
total number of returned instore esps in previous 24 months	0	2.61	1.38	23.00	2.00	5.36
total number of returned instore esps lifetime	0	2.71	1.52	25.00	2.00	5.19
total items purchased during back to school gift guide 2002 promotion	0	4.26	9.39	170.00	-1.00	15.76
total items purchased during bond 2002 promotion	0	3.97	9.15	307.00	-2.00	29.86
total items purchased during expo 2001 promotion	0	3.68	3.95	76.00	-2.00	11.07
total items purchased during holiday gift guide 2001 promotion	0	2.94	1.97	20.00	-2.00	3.43
total items purchased during holiday gift guide 2002 promotion	0	3.31	2.42	26.00	-1.00	3.36
total items purchased during holiday mailer 2001 promotion	0	3.59	2.22	18.00	-3.00	2.09
total items purchased during holiday mailer 2002 promotion	0	3.81	2.84	44.00	-3.00	4.20
promo_nov period: total sales	0	333.94	674.07	11633.21	-111.09	5.97
total $ spent during bond 2002 promotion	0	338.26	526.43	6821.51	-134.00	4.29
total $ spent during expo 2001 promotion	0	366.73	570.37	6491.80	-372.00	3.94
total $ spent during holiday mailer 2002 promotion	0	344.76	493.71	5639.97	-165.00	3.85
total $ spent during holiday mailer promotions	0	366.64	498.45	5818.81	-165.00	3.52
mailed in holiday 2001 mailer	1	0.18	NA	NA	NA	NA
mailed in holiday 2002 mailer	1	0.20	NA	NA	NA	NA
indicator of holiday gift guide 2002 promotion response	1	0.01	NA	NA	NA	NA
indicator of back to school gift guide 2002 promotion response	1	0.01	NA	NA	NA	NA
indicator of bond 2002 promotion response	1	0.01	NA	NA	NA	NA
indicator of expo 2001 promotion response	1	0.01	NA	NA	NA	NA
indicator of holiday gift guide 2001 promotion response	1	0.00	NA	NA	NA	NA
indicator of holiday mailer 2001 promotion response	1	0.01	NA	NA	NA	NA
indicator of holiday mailer 2002 promotion response	1	0.01	NA	NA	NA	NA

APPENDIX C

APPENDIX FOR APPLICATION OF NON-PARAMETRIC BAYESIAN INFERENCE TO TRANSCRIPTIONAL BURSTING

C.1 Proof of theorem 5.3.1

Here we make use of an extension of the extended Schwartz theorem [87]. Theorem 5.3.1 is a special case. In theorem C.1.1, $p_{F,\varphi}(x) = \int \psi(x;\ \lambda, \varphi) dF(\lambda)$ is a mixture distribution parameterized by λ. The theorem states that if $\psi(x;\ \lambda, \varphi)$, in our case the exponential density $\lambda \exp(-\lambda x)$, satisfies condition 1, 2, and 3, and the prior, in our case the Dirichlet process prior with central distribution Gamma(a, b), satisfies condition 4, then we have consistency.

Theorem C.1.1. *[87, Theorem 7.15] Let $p_{F,\varphi}(x) = \int \psi(x;\ \lambda, \varphi) dF(\lambda)$ for a given family of probability densities $x \to \psi(x;\ \lambda, \varphi)$ index by Euclidean parameters $\lambda \in \Lambda \subset \mathbb{R}$ and $\varphi \in \Phi \subset \mathbb{R}^l$. Equip F with the Dirichlet process prior and φ by some other prior π.*

Then if for any given $\epsilon > 0$ and any $m \geq 2$, there exist subsets $\Lambda_m \subset \mathbb{R}$ and $\Phi_m \subset \mathbb{R}^l$ and constants $a_m, A_m, b_m, B_m > 0$ such that

1. *$\|\psi(\cdot;\ \lambda, \varphi) - \psi(\cdot;\ \lambda', \varphi')\|_1 \leq a_m \|\lambda - \lambda'\| + b_m \|\varphi - \varphi'\|$. for all $\lambda, \lambda' \in \Lambda_m$ and $\varphi, \varphi' \in \Phi_m$.*

2. *$\operatorname{diam}(\Lambda_m) \leq A_m$ and $\operatorname{diam}(\Phi_m) \leq B_m$.*

3. *$\log(a_m A_m) \leq C \log(m)$ for some $C > 0$, and $\log(b_m B_m) \leq m\epsilon^2/8l$.*

4. *$\max(\tilde{A}(\Lambda_m^c), \pi(\Phi_m^c)) \leq e^{-C'm}$, for some $C' > 0$.*

Then the posterior distribution $\Pi_m(\cdot \mid X_1, \cdots X_m)$ for $p_{F,\varphi}$ in the model $X_1, \cdots X_m \mid (F, \varphi) \sim p_{F,\varphi}$, for $(F, \varphi) \sim DP(A)$, is strongly consistent relative to the total variational norm at every p_0 in the Kullback-Leibler support of the prior $p_{F,\varphi}$.

In the above, $\tilde{A}(\cdot)$ is the probability measure of the Dirichlet process for a given central measure G. In our case, it is the probability measure on λ_j from a Dirichlet process with central distribution Gamma. diam denotes maximum Euclidean distance between the any two points in the set. In the present application $\varphi = 0$, and the prior π assigned φ is the unit mass assigned at 0. Λ_m^c is the set complement of Λ_m and Φ_m^c is the set complement of Φ_m.

To prove theorem 5.3.1, we have $\psi(t, \lambda, 0) = \lambda \exp(-\lambda t)$, $\Lambda_m = \left(\frac{1}{m+M^*}, m + M^*\right)$ for a constant M^* to be determined. $\Phi_m = \{0\}$ and π is the unit mass on 0. We let $b_m = 1$, $B_m = 1$. Because of the construction of the prior, we have that $\pi(\Phi_m^c) = 0$. This means that we can further reduce the 4 conditions in theorem C.1.1 to the following 4 conditions.

There exists $a_m, A_m \in \mathbb{R}^+$ such that

1. $\|\lambda \exp(-\lambda t) - \lambda' \exp(-\lambda' t)\|_1 \leq a_m \|\lambda - \lambda'\|$. for all $\lambda, \lambda' \in \Lambda_m$
2. $\text{diam}(\Lambda_m) \leq A_m$.
3. $\log(a_m A_m) \leq C \log(m)$ for some $C > 0$
4. $\tilde{A}(\Lambda_m^c) \leq e^{-C'm}$, for some $C' > 0$

We now show that conditions 1-3 apply in lemmas C.1.1 to C.1.3 respectively. Lemmas C.1.4 to C.1.6 show that condition 4 applies. We begin with lemma C.1.1 (condition 1), where we let $a_m = m + M^*$ for a constant M^* to be determined.

Lemma C.1.1. *Let* $\psi(t, \lambda, 0) = \lambda \exp(-\lambda t)$ *and let* $\Lambda_m \subset \mathbb{R}$ *be such that* $\Lambda_m = \left(\frac{1}{m+M^*}, m + M^*\right)$, *then for* $\lambda, \lambda' \in \Lambda_m$,

$$\|\psi(t, \lambda, 1) - \psi(t, \lambda', 1)\|_1 = \int \mid \lambda \exp(-\lambda t) - \lambda' \exp(-\lambda' t) \mid dt \leq 2(m + M^*) \mid \lambda - \lambda' \mid$$

Proof: Without loss of generality, $\lambda' < \lambda$. Note that there is one intersection point of $\lambda \exp(-\lambda t)$ and $\lambda' \exp(-\lambda' t)$ at $t = \log\left(\frac{\lambda}{\lambda'}\right) \frac{1}{\lambda - \lambda'}$. Then,

$$\begin{aligned}\int \mid \lambda \exp(-\lambda t) - \lambda' \exp(-\lambda' t) \mid dt &= \int_0^{\log\left(\frac{\lambda}{\lambda'}\right)\frac{1}{\lambda-\lambda'}} \lambda \exp(-\lambda t) - \lambda' \exp(-\lambda' t) dt \\ &+ \int_{\log\left(\frac{\lambda}{\lambda'}\right)\frac{1}{\lambda-\lambda'}}^{\infty} \lambda' \exp(-\lambda' t) - \lambda \exp(-\lambda t) dt \\ &= 2\left(\frac{\lambda'}{\lambda}\right)^{\frac{\lambda'}{\lambda-\lambda'}} \frac{1}{\lambda}\left(\lambda - \lambda'\right).\end{aligned}$$

Since $\frac{\lambda'}{\lambda} < 1$ and $\lambda > \frac{1}{m+M^*}$, we can now write

$$\begin{aligned}\int \mid \lambda \exp(-\lambda t) - \lambda' \exp(-\lambda' t) \mid dt &\leq 2(m + M^*)\left(\lambda - \lambda'\right) \\ &= 2(m + M^*) \mid \lambda - \lambda' \mid .\end{aligned}$$

We now consider lemma C.1.2 (condition 2), taking $\Lambda_m = (\frac{1}{m+M^*}, m + M^*)$ and $A_m = m + M^*$.

Lemma C.1.2. *Let* $\Lambda_m = (\frac{1}{m+M^*}, m + M^*)$, $\mathrm{diam}(\Lambda_m) \leq m + M^*$.

Proof: $m + M^* - \frac{1}{m+M^*} \leq m + M^*$.

We now consider lemma C.1.3 (condition 3).

Lemma C.1.3. *Let* $a_m = 2(m + M^*)$ *and* $A_m = m + M^*$, *then for* $m \geq 2$, *there exists an* M' *such that for* $M^* \geq M'$,

$$\log(a_m A_m) = \log(2(m + M^*)^2) \leq 2(M^* + 1)\log(m)$$

Proof:

$$\log(a_m A_m) = \log(2(m + M^*)^2)$$

$$\begin{aligned}\log(2(m+M^*)^2) &\leq 2\log(2(m+M^*)) \\ &= 2(M^*+1)\log(m) + 2\log(2(m+M^*)) - 2(M^*+1)\log(m) \\ &= 2(M^*+1)\log(m) + 2\log\left(\frac{2(m+M^*)}{m^{(M^*+1)}}\right)\end{aligned}$$

There is an M' such that for $M^* \geq M'$, and $m \geq 2$, $\frac{2(m+M^*)}{m^{(M^*+1)}} \leq 1$. Then,

$$\log(2(m+M^*)^2) \leq 2(M^*+1)\log(m).$$

Now, take C to be $2(M^*+1)$ where $M^* \geq M'$: this gives us condition 3.

Condition 4 requires a few calculations to be done. The first calculation is to show the concentration rate of Gamma distribution..

Lemma C.1.4. *Let $X \sim$ Gamma(a,b) where $a \geq 2$ and $b \geq 2$, then for m big enough, the following two inequalities hold:*

1. For any $0 \leq v \leq b$, $\quad \mathbb{P}(X \geq m) \leq \exp(-mv - a\log(1-v/b))$.
2. $\mathbb{P}\left(X \leq \frac{1}{m}\right) \leq \frac{b^a}{\Gamma(a)}\frac{1}{m^a}\exp\left(-\frac{b}{m}\right)$.

Proof:

1.

$$\begin{aligned}\mathbb{P}(X \geq m) &= \mathbb{P}(\exp(vX) \geq \exp(vm)) \\ &\leq \frac{\mathbb{E}(\exp(vX))}{\exp(vm)} \\ &\text{by the Markov inequality.} \\ &\leq \exp\left(-mv - a\log(1-v/b)\right)\end{aligned}$$

2. Note that since $a \geq 2$ and $b \geq 2$ there is an m such that $x < \frac{1}{m}$, $\frac{b^a}{\Gamma(a)}x^{a-1}\exp(-bx)$ is

an increasing function in the neighbourhood of x. We can bound the integral below by the rectangle with height $\frac{b^a}{\Gamma(a)}(1/m)^{a-1}\exp(-b/m)$ and width $1/m$.

$$\begin{aligned}\mathbb{P}\left(X \leq \frac{1}{m}\right) &= \int_0^{1/m} \frac{b^a}{\Gamma(a)} x^{a-1}\exp(-bx)\,dx \\ &\leq \frac{1}{m}\frac{b^a}{\Gamma(a)}\frac{1}{m^{a-1}}\exp\left(-\frac{b}{m}\right) \\ &= \frac{b^a}{\Gamma(a)}\frac{1}{m^a}\exp\left(-\frac{b}{m}\right).\end{aligned}$$

We have concentration of gamma distribution. Now we use the following result to solve for $\tilde{A}(\Lambda_m^c)$.

Lemma C.1.5. *[87, Theorem 4.22] If F is the cumulative distribution function of the DP(KG)-process, (Dirichlet Process) where G is a center measure, then*

$$\lim_{G(x)\to 0} \inf F(x)\exp\left(\frac{r\log\|\log KG(x)\|}{KG(x)}\right) = \begin{cases} 0 & \text{if } r<1 \quad a.s. \\ \infty & \text{if } r>1 \quad a.s \end{cases}$$

$$\lim_{G(x)\to 0} \sup F(x)\exp\left(\frac{1}{KG(x)\mid \log KG(x)\mid^r}\right) = \begin{cases} 0 & \text{if } r>1 \quad a.s. \\ \infty & \text{if } r\leq 1 \quad a.s \end{cases}$$

The same results hold if F and G are replaced by $1-F$ and $1-G$.

Remark C.1.1. *This means that for $r>1$*

$$F(x) \leq \exp\left(-\frac{1}{KG(x)\mid \log KG(x)\mid^r}\right), \tag{C.1}$$

$$1-F(x) \leq \exp\left(-\frac{1}{K(1-G(x))\mid \log K(1-G(x))\mid^r}\right). \tag{C.2}$$

Using the notation in lemma C.1.5

$$\begin{aligned}\tilde{A}(\Lambda_m^c) &= F\left(\frac{1}{m+M^*}\right) + 1 - F(m+M^*) \\ &\leq \exp\left(-\frac{1}{KG(\frac{1}{m+M^*}) \mid \log KG(\frac{1}{m+M^*}) \mid^r}\right) \\ &+ \exp\left(-\frac{1}{K(1-G(m+M^*)) \mid \log K(1-G(m+M^*)) \mid^r}\right).\end{aligned}$$

In our proof, $K = 1$, since all the λ's comes from the same Gamma prior.

With this, we compute the $\tilde{A}(\Lambda_m^c)$.

Lemma C.1.6. *Let the central measure be Gamma(a, b) where $a \geq 2$ and $b \geq 2$. There exists C' such that $\tilde{A}(\Lambda_m^c) \leq \exp(-C'm)$ for m great enough. i.e. there is M^* such that for $m \geq M^*$, $\tilde{A}(\Lambda_m^c) \leq \exp(-C'm)$.*

Proof:

We use the result from lemma C.1.4 and C.1.5 by substituting $x = \frac{1}{m}$ in $G(x)$ with its upper bound $\mathbb{P}(X < \frac{1}{m})$ and $x = m$ in $1 - G(x)$ with its upper bound $\mathbb{P}(X > m)$. $K = 1$ in our case since all the λ's come from the same Gamma prior. In the following calculation, we represent the condition $r > 1$ by setting $r = 1.01$.

Bounding $\tilde{A}(\Lambda_m^c)$ is equivalent to bounding

$$\tilde{A}(\Lambda_m^c) \leq \tilde{A}((0, 1/m)) + \tilde{A}((m, \infty)).$$

Using the result from lemma C.1.5, we bound $\tilde{A}((0, 1/m))$ by equation (C.1), substitute

$G(x) = G(\frac{1}{m})$ with its upper bound

$$G\left(\frac{1}{m}\right) \leq \frac{b^a}{\Gamma(a)} \frac{1}{m^a} \exp\left(-\frac{b}{m}\right).$$

We now have

$$\begin{aligned}
\tilde{A}((0,1/m)) &\leq \exp\left(-\frac{1}{\frac{b^a}{\Gamma(a)}\frac{1}{m^a}\exp\left(-\frac{b}{m}\right) \mid \log(\frac{b^a}{\Gamma(a)}\frac{1}{m^a}\exp\left(-\frac{b}{m}\right)) \mid^r}\right) \\
&= \exp\left(-\frac{\Gamma(a)m^a \exp\left(\frac{b}{m}\right)}{b^a \mid a\log(b) - \log(\Gamma(a)) - a\log(m) - \frac{b}{m}) \mid^r}\right) \\
&= \exp\left(-\frac{\Gamma(a)m^{a-1} \exp\left(\frac{b}{m}\right)}{b^a \mid a\log(b) - \log(\Gamma(a)) - a\log(m) - \frac{b}{m}) \mid^r} m\right)
\end{aligned}$$

In order to establish the existence of the desired C_1, we need to show that above inequality is valid for $m \geq 2$ and $a \geq 2$. For large m, we note that

$$\frac{\Gamma(a)m^{a-1} \exp\left(\frac{b}{m}\right)}{b^a \mid a\log(b) - \log(\Gamma(a)) - a\log(m) - \frac{b}{m}) \mid^r} = O\left(\frac{m^{a-1}}{\mid \log(m) \mid^r}\right),$$

which is bounded below. Therefore there is a C_1 such that

$$C_1 \leq \frac{\Gamma(a)m^{a-1} \exp\left(\frac{b}{m}\right)}{b^a \mid a\log(b) - \log(\Gamma(a)) - a\log(m) - \frac{b}{m}) \mid^r}$$

for all m. This gives us

$$\begin{aligned}
\tilde{A}((0,1/m)) &\leq \exp\left(-\frac{\Gamma(a)m^a \exp\left(\frac{b}{m}\right)}{b^a \mid a\log(b) - \log(\Gamma(a)) - a\log(m) - \frac{b}{m}) \mid^r}\right) \\
&\leq \exp\left(-C_1 m\right)
\end{aligned}$$

for m large enough.

Next, we bound $\tilde{A}(m,\infty)$ by equation (C.2), substitute $1-G(x)=1-G(m)$ with its upper bound

$$1-G\left(m\right)\leq\exp\left(-mv-a\log(1-v/b)\right).$$

We now have

$$\begin{aligned}\tilde{A}(m,\infty)&\leq\exp\left(-\frac{1}{\exp\left(-mv-a\log(1-v/b)\right)\mid mt+a\log(1-v/b)\mid^{r}}\right)\\&\leq\exp\left(-\frac{\exp\left(mv+a\log(1-v/b)\right)}{\mid mv+a\log(1-v/b)\mid^{r}}\right).\end{aligned}$$

Let $v=b/2$. We now have

$$\tilde{A}(m,\infty)\leq\exp\left(-\frac{\exp\left(mb/2-a\log(2)\right)}{\mid mb/2-a\log(2)\mid^{r}}\right). \tag{C.3}$$

Because $\exp(mb/2)$ grows at an exponential rate, it is clear from (C.3) there exists C_2 such that

$$\tilde{A}(m,\infty)\leq\exp(-C_2m).$$

Let $C'=\min(C_1,C_2)$, giving

$$\tilde{A}((0,1/m))+\tilde{A}((m,\infty))\leq2\exp(-C'm)\leq\exp\left(-\frac{C'}{2}m\right)$$

for large enough m.

The $C'/2$ found in the proof is what is needed to satisfy condition 4.

We now give a rigorous proof of Theorem 5.3.1.

Proof:

We first define $\psi(t;\ \lambda,\varphi)=\lambda\exp(-\lambda t)$ and $\varphi=0$. Let π the prior assigned to φ be the unit mass on $\{0\}$. For $m\geq2$, define $\Lambda_m=(1/(m+M^*),(m+M^*))$ for M^* large

enough to satisfy the conditions of lemma C.1.3 and lemma C.1.6. Let $a_m = 2(m + M^*)$, $A_m = m + M^*$, $b_m = 1$, and $B_m = 1$.

We have now shown that conditions 1, 2, 3, and 4 in theorem C.1.1 are satisfied. Lemma C.1.1 shows that condition 1 is satisfied . Lemma C.1.2 shows that condition 2 is satisfied. Lemma C.1.3 shows that condition 3 is satisfied and Lemma C.1.6 shows that condition 4 is satisfied. By the result of theorem C.1.1, we have shown strong consistency in total variations.