

Variable Selection via Thompson Sampling

Yi Liu* and Veronika Ročková†

July 3, 2020

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 multiply 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.

Keywords: *BART, Combinatorial Bandits, Interpretable Machine Learning, Spike-and-Slab, Thompson Sampling, Variable Selection*

1 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

*Yi Liu is a 3rd year PhD student at the Department of Statistics at the University of Chicago

†Veronika Rockova is Assistant Professor in Econometrics and Statistics at the Booth School of Business of the University of Chicago. The author gratefully acknowledges the support from the James S. Kemper Foundation Research Fund at the Booth School of Business.

of variation. Unfortunately, many today’s most powerful machine learning prediction algorithms lack the capacity to perform variable screening in a principled and/or reliable way. For example, deep learning (DL) is widely accepted as one of the best performing artificial intelligence (AI) platforms. However, DL prediction mappings lack an intuitive algebraic form which renders their interpretability/explainability (i.e. insight into the black box decision process) far from straightforward. Substantial effort has been recently devoted to enhancing the explainability of machine (deep) learning through the identification of key variables that drive predictions (Garson (1991); Olden and Jackson (2002); Zhang et al. (2000); Lu et al. (2018)). For instance, Lu et al. (2018) integrated the idea of a knock-off filter with deep neural networks and derived ‘DeepPINK’ for variable selection that controls the false discovery rate. In a similar vein, Burns et al. (2019) proposed a formal testing procedure for whether the model prediction is significantly different after features have been replaced with uninformative counterfactuals. Horel and Giesecke (2019) proposed a test statistic for a single-layer feed forward network. While possessing nice theoretical guarantees, many of these procedures are not yet feasible for large-scale applications.

A variable can be important because its change has a causal impact or because leaving it out reduces overall prediction capacity (Jiang and Owen (2003)). Such leave-one-covariate-out type inference has a long tradition, going back to at least Breiman (2001). 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. Lei et al. (2018) 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 Fisher et al. (2019) for a recent overview. In Bayesian forests, such as BART (Chipman et al. (2001)), 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 (Garson (1991); Ye and Sun (2018)). Owen and Prieur (2017) introduce a variable importance based on a Shapley value and Hooker (2007) 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 Ishwaran et al. (2007); Kazemitabar et al. (2017)).

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 $\mathbf{Y}^{(n)} = (Y_1, \dots, Y_n)'$ that is linked stochastically to a fixed set of predictors $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})'$ for $1 \leq i \leq n$ through

$$Y_i = f_0(\mathbf{x}_i) + \epsilon_i \quad \text{where } \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2), \quad (1)$$

and where f_0 is an unknown regression function. The variable selection problem occurs when there is a subset $\mathcal{S}_0 \subset \{1, \dots, p\}$ of $q_0 = |\mathcal{S}_0|$ 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 $q = |\mathcal{S}|$ and then a conditionally uniform prior on \mathcal{S} , given q , i.e. $\pi(\mathcal{S}|q) = \frac{1}{\binom{p}{q}}$. 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

$$\mathbb{P}(\gamma_i | \theta) = \theta, \quad \theta \sim \text{Beta}(a, b) \text{ for some } a, b > 0. \quad (2)$$

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 (Mitchell and Beauchamp (1988); Rockova and George (2018); Rossell and Telesca (2017); Vannucci and Stingo (2010); Brown et al. (1998)), computational aspects (Carbonetto et al. (2012); Rockova and George (2014); Bottolo et al. (2010), George and McCulloch (1993), George and McCulloch (1997)) and/or variable selection consistency results (Castillo et al. (2015); Johnson and Rossell (2012); Narisetty et al. (2014)). In this work, we leave behind the linear model framework and focus on interpretable machine learning linking spike-and-slab methods with binary bandits.

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 (Barbieri and Berger, 2004). 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 2 revisits known facts about multi-armed bandits. Section 3 develops the bandits framework for variable selection and Section 4 proposes Thompson Variable Selection and presents a regret analysis. Section 5 presents two implementations (offline and online) on two benchmark simulated data. Section 6 presents a thorough simulation study and Section 7 showcases TVS performance on real data. We conclude with a discussion in Section 8.

2 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 $\theta^* = \max_{1 \leq i \leq p} \theta_i$ the best average reward and with $\Delta_i = \theta^* - \theta_i$ 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 Lai and Robbins (1985) and Thompson Sampling (TS) of Thompson (1933). Tracing back to Agrawal (1995), UCB consists of computing, at each round t and for each arm i , a reward index (e.g. an upper bound of the mean reward of the considered arm that holds with high confidence) and then selecting the arm with the largest index. Thompson Sampling, on the other hand, is a Bayesian-inspired heuristic algorithm that achieves a logarithmic expected regret (Agrawal and Goyal, 2012) in the Bernoulli bandit problem. Starting with a non-informative prior $\theta_i \stackrel{iid}{\sim} \text{Beta}(1, 1)$ for $1 \leq i \leq p$, this algorithm: (a) updates the distribution of θ_i as $\text{Beta}(a_i(t) + 1, b_i(t) + 1)$, 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 $\theta_i(t)$. Agrawal and Goyal (2012) extended this algorithm to the general case where rewards are not necessarily Bernoulli but general random variables on the interval $[0, 1]$. Scott (2010) and Sabes and Jordan (1996) proposed a Randomized Probability Matching variant which allocates observations to arms according to their probability of being the best arm.

The MAB problem is most often formulated as a single-play problem, where only one arm can be selected at each round. Komiyama et al. (2015) 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, $\theta_1 > \theta_2 > \dots > \theta_p$. The player is interested in maximizing the sum of expected rewards over drawn arms, where the optimal action is

playing the top L arms $\mathcal{S}_0 = \{1, \dots, L\}$. 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. $\mathbb{E}[\mathcal{R}(T)] = \mathbb{E} \sum_{t=1}^T (\sum_{i \in \mathcal{S}_0} \theta_i - \sum_{i \in \mathcal{S}_t} \theta_i)$ Fixing L , the number of arms played, Komiyama et al. (2015) 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.

Combinatorial bandit problems (Chen et al. (2013); Gai et al. (2012); Cesa-Bianchi and Lugosi (2012)) 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. Wang and Chen (2018); Combes and Proutiere (2014); Kveton et al. (2015); Combes and Proutiere (2014); Kveton et al. (2015).

3 Variable Selection as a Bandit Problem

Bayesian model selection is regarded as more or less synonymous to finding the MAP (maximum-a-posteriori) model $\hat{\mathcal{S}} = \arg \max_{\mathcal{S}} \pi(\mathcal{S} | \mathbf{Y}^{(n)})$. Even when the marginal likelihood is available, this model can be 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 $\theta_i \sim \text{Beta}(a_i, b_i)$ for $1 \leq i \leq 2^p$ for some¹ $a_i > 0$ 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 become standard practice to report the *median probability model* (MPM) (Barbieri and Berger (2004)) consisting of those variables whose

¹chosen to correspond to marginals of a Dirichlet distribution

posterior inclusion probability $\pi_i \equiv \mathbb{P}(\gamma_i = 1 \mid \mathbf{Y}^{(n)})$ is at least 0.5. More formally, MPM is defined, for $\boldsymbol{\pi} = (\pi_1, \dots, \pi_p)'$, as

$$\widehat{\mathcal{S}}_{MPM} = \arg \max_{\mathcal{S}} r_{\boldsymbol{\pi}}(\mathcal{S}) = \{i : \pi_i \geq 0.5\} \quad \text{where} \quad r_{\boldsymbol{\pi}}(\mathcal{S}) = \left\{ \prod_{i \in \mathcal{S}} \pi_i \prod_{i \notin \mathcal{S}} (1 - \pi_i) \right\}. \quad (3)$$

This model is the optimal predictive model in linear regression under some assumptions (Barbieri et al., 2018). 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 (George and McCulloch (1997); Narisetty et al. (2014); Bhattacharya et al. (2016)), here we explore new curious connections to bandits in order to develop a much faster stochastic optimization routine for finding MPM-alike models when the true model is not necessarily linear.

While the MAP model suggests treating *each model* as a bandit arm, the MPM model suggests treating *each variable* 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 $r_{\boldsymbol{\pi}}(\mathcal{S})$ can be regarded as a mean regret of a combinatorial arm (more below) which, given $\boldsymbol{\pi}$, has MPM as its computational oracle.

We view Bayesian spike-and-slab selection through the lens of *combinatorial bandit problems* by treating variable selection indicators γ_i 's in (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 (2) which deploys one θ for all arms, each arm $i \in \{1, \dots, p\}$ now has *its own* prior inclusion probability θ_i , i.e.

$$\mathbb{P}(\gamma_i = 1 \mid \theta_i) = \theta_i, \quad \theta_i \stackrel{\text{ind}}{\sim} \text{Beta}(a_i, b_i) \quad \text{for some } a_i, b_i > 0. \quad (4)$$

In the original spike-and-slab setup (2), the mixing weight θ served as a global shrinkage parameter determining the level of sparsity and linking coordinates to borrow strength (Rockova and George (2018)). In our new bandit formulation (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.

3.1 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 (Shapley (1953); Owen and Prieur (2017)), a construct from cooperative game theory for the attribution problem that distributes the value created by a team to its individual members. The Shapley value has become a popular method for prediction attribution in machine learning (Sundararajan and Najmi (2019); Mase et al. (2019)). Although the Shapley value has a strong intuitive appeal, computation remains a challenge (Castro et al. (2009); Song et al. (2016)).

We try a different route. Instead of collecting a global reward first and then redistributing it, our strategy consists of first collecting individual rewards $\gamma_i^t \in \{0, 1\}$ for each played arm $i \in \mathcal{S}_t$ and then weaving them into a global reward $R(\mathcal{S}_t)$. We assume that γ_i^t 's are iid from (4) for each $i \in \{1, \dots, p\}$. Unlike traditional combinatorial bandits that define the global reward $R(\mathcal{S}_t) = \sum_{i \in \mathcal{S}_t} \gamma_i^t$ as a sum of individual outcomes (Gai et al., 2012), 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 $\mathbb{E}[R(\mathcal{S}_t)] = \prod_{i \in \mathcal{S}_t} \theta_i \prod_{i \notin \mathcal{S}_t} (1 - \theta_i) = r_{\theta}(\mathcal{S}_t)$ and has the “median probability model” as its computational oracle, as can be seen from (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

$$R_C(\mathcal{S}_t) = \sum_{i \in \mathcal{S}_t} \log(C + \gamma_i^t). \quad (5)$$

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 $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)'$, i.e.

$$r_{\boldsymbol{\theta}}^C(\mathcal{S}_t) = \mathbb{E}[R_C(\mathcal{S}_t)] = \sum_{i \in \mathcal{S}_t} \left[\theta_i \log\left(\frac{C+1}{C}\right) - \log\left(\frac{1}{C}\right) \right]. \quad (6)$$

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

$$\mathcal{S}_O = \left\{ i : \theta_i \geq \frac{\log(1/C)}{\log[(C+1)/C]} \right\}. \quad (7)$$

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 $\log(1+C) = -\log(C)$. 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 4

3.2 The Local Rewards

Having introduced the global reward (5), 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 $\{1, \dots, p\}$ and with \mathcal{D} the “data” at hand consisting of $|\mathcal{D}|$ observations (Y_i, \mathbf{x}_i) from (1). We introduce a *feedback rule*

$$r(\mathcal{S}_t, \mathcal{D}) : \mathbb{S} \times \mathbb{R}^{|\mathcal{D}|} \rightarrow \{0, 1\}^{|\mathcal{S}_t|}, \quad (8)$$

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 $\mathcal{D}_n = \{(Y_i, \mathbf{x}_i) : 1 \leq i \leq n\}$, 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 5.1. As a lead example of this strategy in this paper, we consider a stochastic reward based on BART (Chipman et al. (2001)). 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

$$\gamma_i^t = \mathbb{I}(M^{th} \text{ sample from the BART posterior splits on the variable } x_i). \quad (9)$$

The mean reward $\theta_i = \mathbb{P}(\gamma_i^t = 1) = \mathbb{P}[i \in \mathcal{F} | \mathcal{D}_n]$ 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 (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 $|\mathcal{S}_t| \ll p$) and *only* for M burn-in MCMC iterations, computational gains are dramatic (as we will see in Section 5.1).

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 5.2. 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 $\mathcal{D}_t = \{(Y_i, \mathbf{x}_i) : (t-1)s + 1 \leq i \leq ts\}$ for $t = 1, \dots, T$. 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

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.

least once on average,

$$\gamma_i^t = \mathbb{I}[IM(i; \mathcal{D}_t, \mathcal{S}_t) \geq 1]. \quad (10)$$

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 (1), uses a variable x_i at least once on average over M iterations. We illustrate this online variant in Section 5.2.

Remark 1 *Instead of binary local rewards (8), one can also consider continuous rewards $\gamma_i^t \in [0, 1]^{|S_t|}$ by rescaling variable importance measures obtained by a machine learning method (e.g. random forests (Louppe et al. (2013)), deep learning (Horel and Giesecke (2019)), and BART (Chipman et al. (2010))). Our Thompson sampling algorithm can be then modified by dichotomizing these rewards through independent Bernoulli trials with probabilities equal to the continuous rewards (Agrawal and Goyal, 2012).*

4 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 θ_i^* 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 (Chen et al. (2013), Gai et al. (2010)) where, instead of playing one arm at each play opportunity t , we play a random subset $\mathcal{S}_t \subseteq \{1, \dots, p\}$ 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 incorporates past reward experiences up to time t (as we discussed in Section 2). 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 (Wang and Chen (2018)) dictates playing the *subset* that maximizes the expected global reward, given the vector of sampled

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

Algorithm 1: <i>Thompson Variable Selection with BART</i>
Define $\tilde{C} = \frac{\log(1/C)}{\log((1+C)/C)}$ for some $0 < C < 1$ and pick $M, a, b > 0$
Initialize $a_i(0) := a$ and $b_i(0) := b$ for $1 \leq i \leq p$ and for $t = 1, \dots, T$
Choose Step
C: Set $\mathcal{S}_t = \emptyset$ and for $i = 1 \dots p$ do
C1: Sample $\theta_i(t) \sim \text{Beta}(a_i(t), b_i(t))$
C2: Compute $\mathcal{S}_t = \{i : \theta_i(t) \geq \tilde{C}\}$ from (7)
C2*: Compute \mathcal{S}_t from (13)
Reward Step
R: Collect local rewards for each $1 \leq i \leq p$ from (9) (offline) or (10) (online)
Update Step
U: If $\gamma_i^t = 1$ then set $a_i(t+1) = a_i(t) + 1$, else $b_i(t+1) = b_i(t) + 1$

Algorithm 1: Thompson Variable Selection with BART (* is an alternative with known q^*)

probabilities $\boldsymbol{\theta}(t) = (\theta_1(t), \dots, \theta_p(t))'$. The availability of the computational oracle (from Lemma 1) makes this step awkwardly simple as it boils down to computing \mathcal{S}_O in (7). Unlike multi-play bandits where the number of played arms is predetermined (Komiya (2015)), this strategy allows one to adapt to the size of the model. We do, however, consider a variant of the computational oracle (see (13) below) for when the size $q^* = |\mathcal{S}^*|$ of the ‘true’ model $\mathcal{S}^* = \arg \max_{\mathcal{S}} r_{\boldsymbol{\theta}^*}^C(\mathcal{S})$ is known. The *Choose Step* is then followed by the *Reward Step* (step R in Table 1) which assigns a prize to the chosen subset \mathcal{S}_t by collecting individual rewards γ_i^t (for the offline setup in (9) or for the online setup (10)). Finally, each TVS iteration concludes with an *Update Step* which updates the beta posterior distribution (step U in Table 1).

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 posterior⁴ distributions that incorporate past rewards and the observed data \mathcal{D} . One natural way to distill evidence for variable selection is through the means $\boldsymbol{\pi}(t) = (\pi_1(t), \dots, \pi_p(t))'$ of these beta distributions

$$\pi_i(t) = \frac{a_i(t)}{a_i(t) + b_i(t)}, \quad 1 \leq i \leq p, \quad (11)$$

which serve as a proxy for posterior inclusion probabilities. Similarly as with the classical median probability model (Barbieri and Berger (2004)), one can deem important those variables with $\pi_i(t)$ above 0.5 (this corresponds to one specific choice of C in Lemma 1). More generally, at each iteration t TVS outputs a model $\hat{\mathcal{S}}_t$, which satisfies $\hat{\mathcal{S}}_t =$

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

$\arg \max_{\mathcal{S}} r_{\pi(t)}^C$. From Lemma 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 5 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.

4.1 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 Agrawal and Goyal (2012) where the first finite-time analysis was presented for single-play bandits. Later, Leike et al. (2016) 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, Komiyama et al. (2015) extended TS to multi-play problems with a fixed number of played arms and showed that it achieves the optimal regret bound. Recently, Wang and Chen (2018) 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

$$Reg(T) = \mathbb{E} \left[\sum_{t=1}^T (r_{\theta^*}^C(\mathcal{S}^*) - r_{\theta^*}^C(\mathcal{S}_t)) \right], \quad (12)$$

where $\mathcal{S}^* = \arg \max_{\mathcal{S}} r_{\theta^*}^C(\mathcal{S}_t)$ with $q^* = |\mathcal{S}^*|$, $\theta_i^* = \mathbb{E}[\gamma_i^t]$ and where the expectation is taken over the unknown drawing policy. Choosing C as in Lemma 1, one has $\log(1+C) = -\log(C) = D$ and thereby

$$Reg(T) = D \mathbb{E} \left[\sum_{t=1}^T \sum_{i=1}^p (2\theta_i^* - 1) [\mathbb{I}(i \in \mathcal{S}^* \setminus \mathcal{S}_t) - \mathbb{I}(i \in \mathcal{S}_t \setminus \mathcal{S}^*)] \right].$$

Note that $(2\theta_i^* - 1)$ is positive iff $i \in \mathcal{S}^*$. Upper bounds for the regret (12) under the drawing policy of our TVS Algorithm 1 can be obtained under various assumptions.

Assuming that the size q^* of the optimal model \mathcal{S}^* is known, one can modify Algorithm 1 to confine the search to models of size up to q^* . Denoting $\mathcal{I} = \{\mathcal{S} \subset \{1, \dots, p\} : |\mathcal{S}| \leq q^*\}$, one plays the optimal set of arms *within the set* \mathcal{I} , i.e. replacing the computational oracle in (7) with $\mathcal{S}_O^{q^*} = \arg \max_{\mathcal{S} \in \mathcal{I}} r_{\boldsymbol{\theta}}(\mathcal{S})$. We denote this modification with $C2^*$ in Table 1. It turns out that this oracle can also be easily computed, where the solution consists of (up to) the top q^* arms that pass the selection threshold, i.e.

$$\mathcal{S}_O^{q^*} = \left\{ i : \theta_i \geq \frac{\log(1/C)}{\log[(1+C)/C]} \right\} \cap J(\boldsymbol{\theta}) = \{(i_1, \dots, i_{q^*})' \in \mathbb{N}^{q^*} : \theta_{i_1} \geq \theta_{i_2} \geq \dots \geq \theta_{i_{q^*}}\}. \quad (13)$$

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

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

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

Proof: Since \mathcal{I} is a matroid (Kveton et al. (2014)) and our mean regret function is Lipschitz continuous and it depends only on expected rewards of revealed arms, one can apply Theorem 4 of Wang and Chen (2018).

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

Lemma 3 *Define the maximal reward gap $\Delta_{max} = \max_{\mathcal{S}} \Delta_{\mathcal{S}}$ where $\Delta_{\mathcal{S}} = [r_{\boldsymbol{\theta}}(\mathcal{S}^*) - r_{\boldsymbol{\theta}}(\mathcal{S})]$ and for each arm $i \in \{1, \dots, p\}$ define $\eta_i \equiv \max_{\mathcal{S}: i \in \mathcal{S}} \frac{8B^2|\mathcal{S}|}{\Delta_{\mathcal{S}} - 2B(q^{*2} + 2)\varepsilon}$ 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*

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

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

Proof: Follows from Theorem 1 of Wang and Chen (2018).

We now extend Lemma 3 to the case when the rewards obtained from pulling different arms are related to one another. Gupta et al. (2020) 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 (Pandey et al. (2007)) 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

$$\theta_i(\mathcal{S}) = \mathbb{P}(\gamma_i^t = 1 | \mathcal{S}_t = \mathcal{S}). \quad (14)$$

We are interested in obtaining a regret bound for the Algorithm 1 assuming (14) in which case the expected global regret (6) writes as

$$r_{\boldsymbol{\theta}}^C(\mathcal{S}_t) = \mathbb{E}[R_C(\mathcal{S}_t)] = \sum_{i \in \mathcal{S}_t} \left[\theta_i(\mathcal{S}_t) \log \left(\frac{C+1}{C} \right) - \log \left(\frac{1}{C} \right) \right]. \quad (15)$$

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

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

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

Under this assumption we provide the following regret bound.

Theorem 1 Suppose that \mathcal{S}^* is strongly identifiable with $\alpha > 0$. Choosing C as in Lemma 1, the regret of Algorithm 1 satisfies

$$\text{Reg}(T) \leq \Delta_{\max} \left[\frac{8p \log(T)}{\alpha^2} + c(\alpha)q^* + \left(2 + \frac{4}{\alpha^2} \right) p \right], \quad (16)$$

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

Proof: Appendix (Section A)

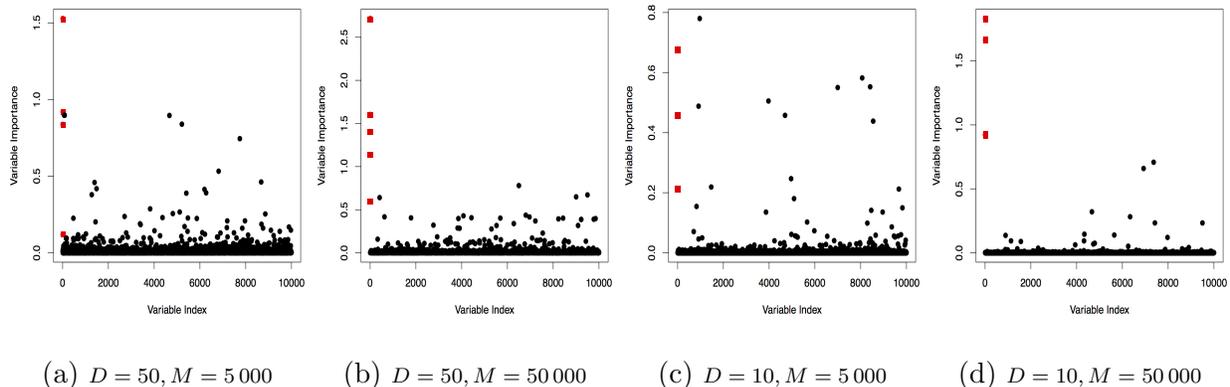


Figure 1: BART variable importance using `sparse=TRUE` and various number of trees D and MCMC iterations M . Red squares (the first five covariates) are signals and black dots are noise variables.

5 Interpretable Machine Learning with TVS

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 (8)), we will focus on two specific choices that we broadly categorize into *offline* variants for when $p \gg n$ (Section 5.1) and *streaming/online* variants for when $n \gg p$ (Section 5.2).

5.1 Offline TVS

As a lead example in this section, we consider the benchmark Friedman data set (Friedman (1991)) with a vastly larger number of $p = 10\,000$ predictors $\mathbf{x}_i \in [0, 1]^p$ obtained by iid sampling from `Uniform(0, 1)` and responses $\mathbf{Y} = (Y_1, \dots, Y_n)'$ obtained from (1) with $\sigma^2 = 1$ and

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

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 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 (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 1(d)). We use the TVS implementation in Table 1 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 2, which depicts ‘posterior inclusion probabilities’ $\pi_i(t)$ defined in (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, \dots, x_5 are depicted in red while the noise variables are black. Figure 2 shows a very successful demonstration for several reasons. The first panel (Figure 2a) shows a very weak learner (as was seen from Figure 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

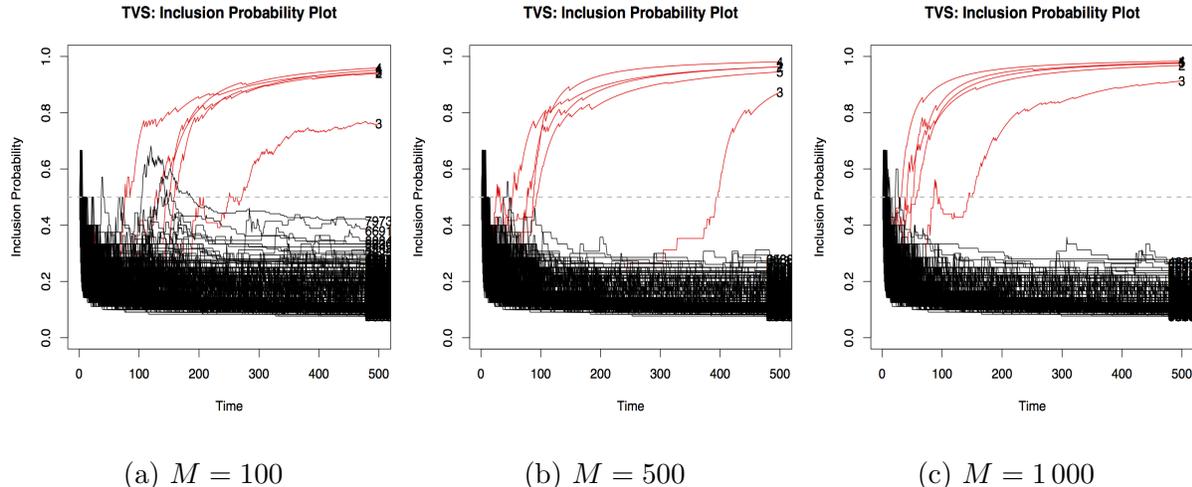


Figure 2: The evolution of inclusion probabilities (11) over “time” t for the Friedman data set. The plot depicts posterior inclusion probabilities $\pi_i(t)$ in (11) over time (number of TVS iterations). Red lines indicate the 5 signal variables and black lines indicate the noise variables.

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 ($M = 1000$ 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 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 2a). 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 $\lambda_0 \in \{0.1 + k \times 10 : k = 1, \dots, 10\}$) which relies on a linear model missed the quadratic predictor but identified all 4 remaining signals with no false positives.

5.2 Online TVS

As the second TVS example, we focus on the case with many more observations than covariates, i.e. $n \gg p$. As we already pointed out in Section 3.2, we assume that the

dataset $\mathcal{D}_n = \{(Y_i, \mathbf{x}_i) : 1 \leq i \leq n\}$ has been partitioned into mini-batches \mathcal{D}_t of size $s = n/T$. We deploy our online TVS method (Table 1 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 (Foster and Stine (2008)). Using the notation $r_t \equiv r(\mathcal{S}_t, \mathcal{D}_t)$ in (8) with $\mathcal{D}_t = \{(Y_i, \mathbf{x}_i) : (t-1)s + 1 \leq i \leq ts\}$ and having processed $t-1$ mini-batches, one can treat the beta posterior as a new prior for the incoming data points, where

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

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 distribution⁵ 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 (Liang et al., 2018, 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 $e_i, z_{ij} \stackrel{iid}{\sim} \mathcal{N}(0, 1)$. This creates a sample correlation of about 0.5 between all variables. The responses $\mathbf{Y} = (Y_1, \dots, Y_n)'$ are then obtained from (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 $n = 20000$ and $p = 1000$. 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 $\lambda_0 \in \{0.1 + k \times 10 : k = 1, \dots, 10\}$) only identifies variables x_1, x_2 and x_5 . Next, we deploy the BART method with variable selection (Linero 2016) by setting

⁵Treating the rewards as data.

	$s = 100$		$s = 200$		$s = 500$		$s = 1\,000$		$s = 5\,000$		$s = 10\,000$		$s = 20\,000$	
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 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.

`sparse=TRUE` (Linero and Yang (2018)) and 50 trees⁶ in the BART software (Chipman et al. (2010)). The choice of 50 trees for variable selection was recommended in Bleich et al. (2014) 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 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 1. This illustrates how selection based on the importance measure is difficult to automate. While visually inspecting the importance measure for $M = 20\,000$ and $s = 20\,000$ (the entire dataset) in Figure 3d 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 (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 (50 trees). Choosing $s = 100$ and $M = 10\,000$, BART may

⁶Results with 10 were not nearly as satisfactory.

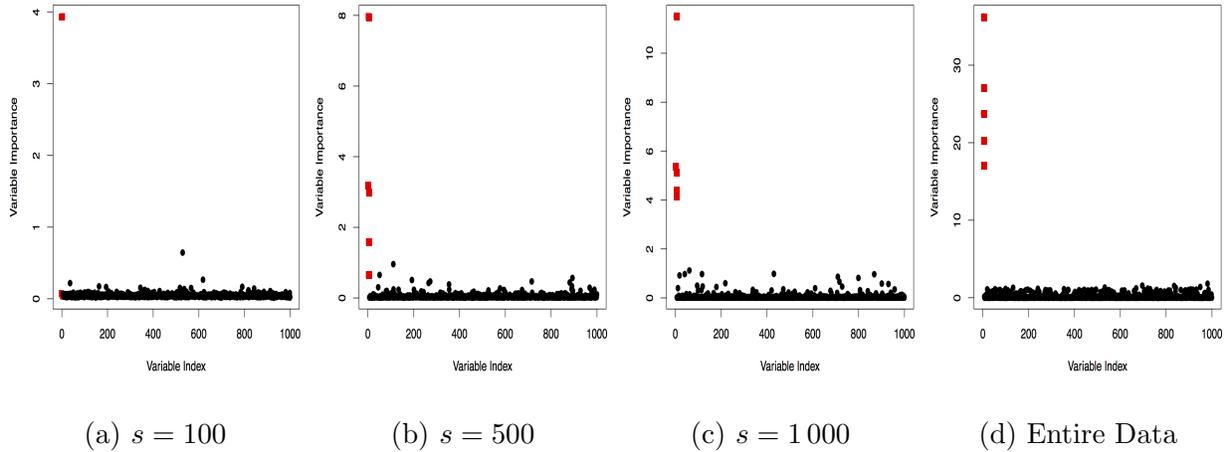


Figure 3: BART variable importance using $T = 20\,000$ (using 50 trees and `sparse=TRUE`) and various data subsets. Red squares (the first five covariates) are signals and black dots are noise variables.

not be able to obtain perfect signal isolation on a single data batch (see Figure 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 4(a)). Comparing Figure 3(b) and Figure 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 stabilize. For example, with $s = 1\,000$ one obtains a far stronger learner (Figure 3(c)), but the separation may not be as clear after only $T = n/s = 20$ TVS iterations (Figure 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 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 5 at the beginning are slightly different from Figure 4. Despite the random nature, however, we have seen the separation apparent from Figure 5 occur consistently across multiple runs of the method.

Several observations can be made from the timing and performance comparisons presented in Table 2. When the batch size is not large enough, repeated runs will not help.

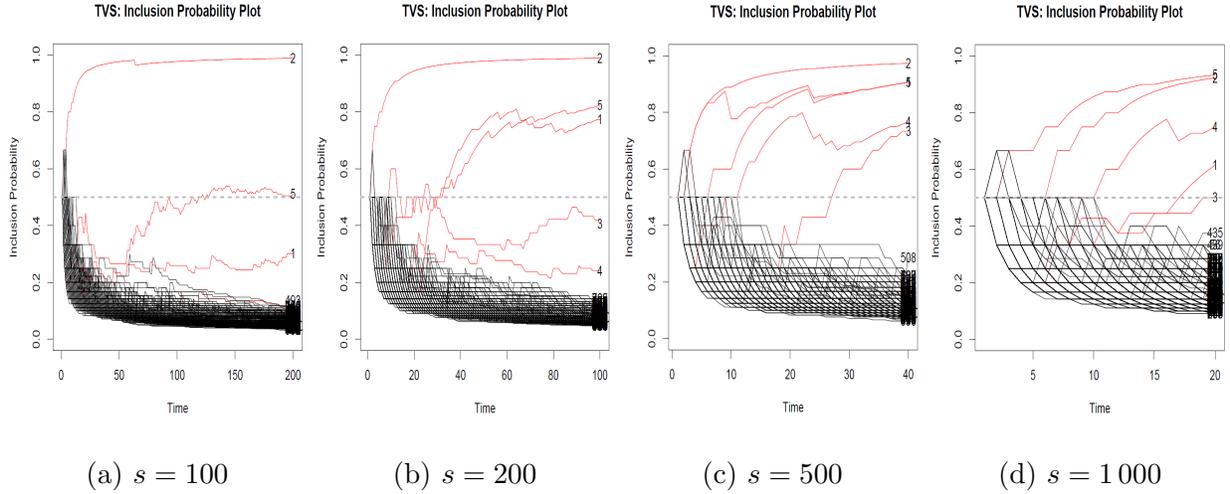


Figure 4: TVS inclusion probabilities $T = 10\,000$ (using 10 trees and `sparse=TRUE`) and various batch sizes after single pass through the data.

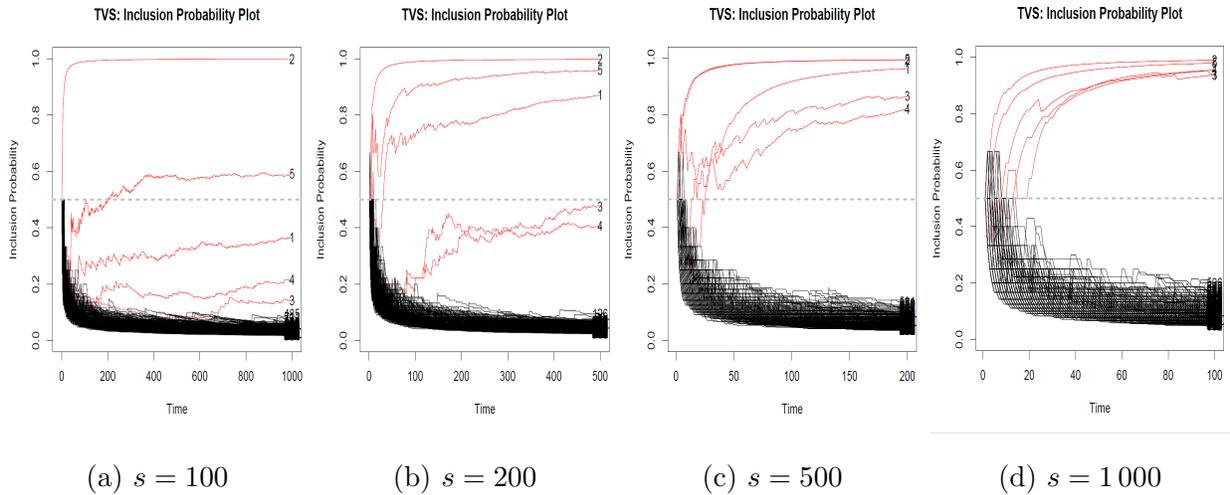


Figure 5: TVS inclusion probabilities $T = 10\,000$ (using 10 trees and `sparse=TRUE`) and various batch sizes after 5 passes through the data.

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.

	$s = 100$		$s = 200$		$s = 500$		$s = 1000$	
	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
$T = 1000$								
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
$T = 10000$								
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 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.

6 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.

6.1 Offline Cases

For a more comprehensive performance evaluation, we consider the following 4 mean functions $f_0(\cdot)$ to generate outcomes using (1). For each setup, we summarize results over 50 datasets of a dimensionality $p \in \{1000, 10000\}$ and a sample size $n = 300$.

- **Linear Setup:** The regressors \mathbf{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 (1) with $\sigma^2 = 5$) via the mean function $f_0(\mathbf{x}_i) = x_{i1} + 2x_{i2} + 3x_{i3} - 2x_{i4} - x_{i5}$.

- **Friedman Setup:** The Friedman setup was described earlier in Section 4. In addition, we now introduce correlations of roughly 0.3 between the explanatory variables.
- **Forest Setup:** We generate \mathbf{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 (1) with $\sigma^2 = 0.5$.
- **Liang et al (2016) Setup:** This setup was described earlier in Section 5.2. We now use $\sigma^2 = 0.5$.

We run TVS with $M = 500$ and $M = 1000$ 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 (Linero and Yang (2018)) 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 $M = 50000$ 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 Bleich et al. (2014) 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 (Rockova and George (2018)) 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 report the average performance (over 50 datasets) for $p = 10000$ in Figure 6 and the rest (for $p = 1000$) 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

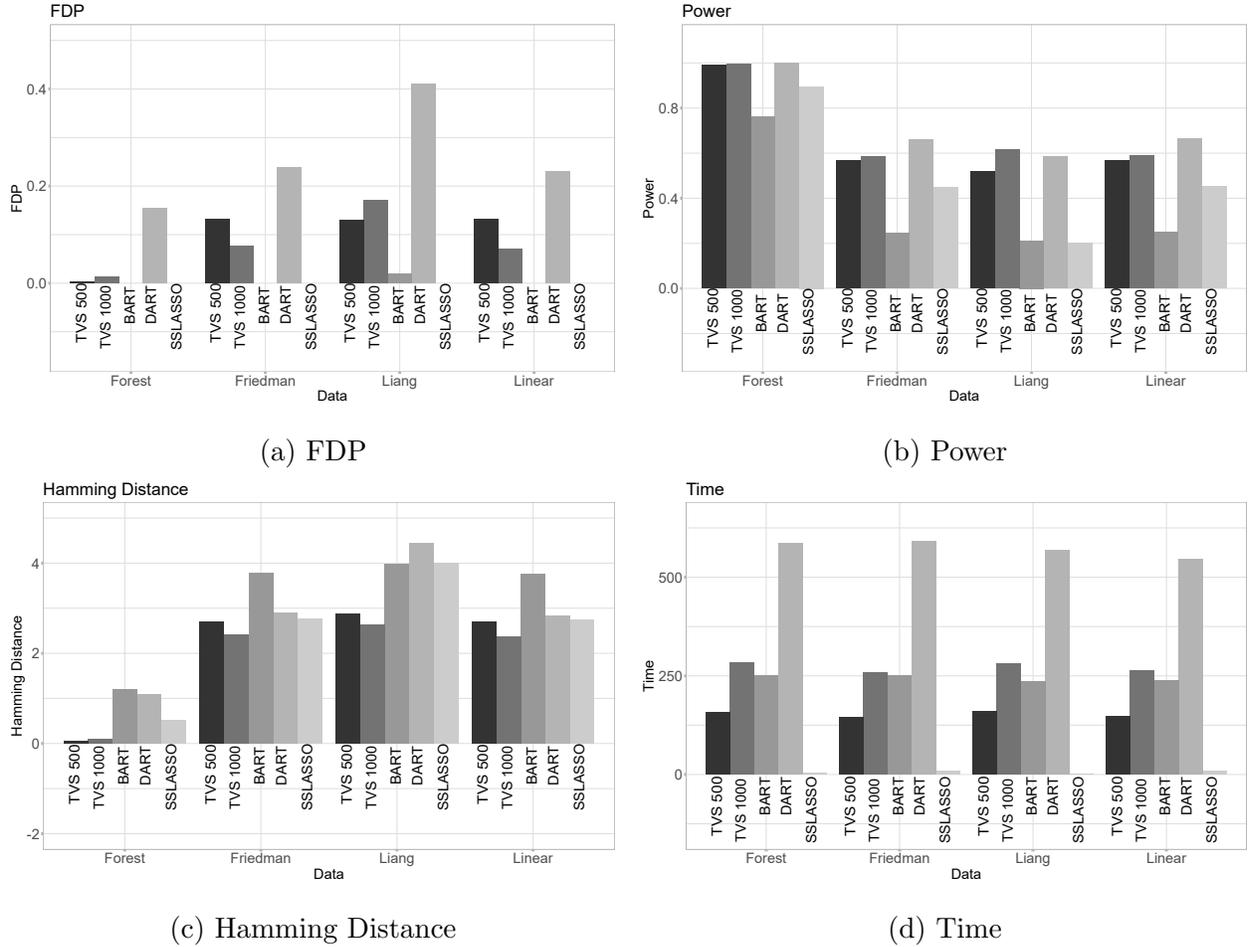


Figure 6: Graphs denoting FDP (6a), Power (6b), Hamming Distance (6c), and Time (6d) for the 4 choices of f_0 assuming $p = 10\,000$ and $n = 300$. The x -axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1000$.

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 $M = 1000$, 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 .

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 $\hat{\mathcal{S}}_t$ 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 ~ 200 iterations for $p = 1000$

and ~ 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 B.1)

6.2 Online Cases

We now consider a simulation scenario where $n \gg p$, i.e. $p = 1\,000$ and $n = 10\,000$. As described earlier in Section 5.2, we partition the data into minibatches $(\mathbf{Y}^{(b)}, \mathbf{X}^{(b)})$ of size s , where $\mathbf{Y}^{(b)} = (Y_i : (b-1)s + 1 \leq i \leq bs)$ and $\mathbf{X}^{(b)} = [\mathbf{x}_i : (b-1)s + 1 \leq i \leq bs]'$ for $b = 1, \dots, n/s$ 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 6.1. 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 7 and the rest is in the Appendix (Section B.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 7 reports the results with a stopping criterion). The results are very encouraging. While SSLASSO's performance is very strong, we notice that in the non-linear setup of Liang et al. (2018) there are false non-discoveries.

7 Application on Real Data

7.1 HIV Data

We will apply (offline) TVS on a benchmark Human Immunodeficiency Virus Type I (HIV-I) data described and analyzed in Rhee et al. (2006) and Barber et al. (2015). This publicly

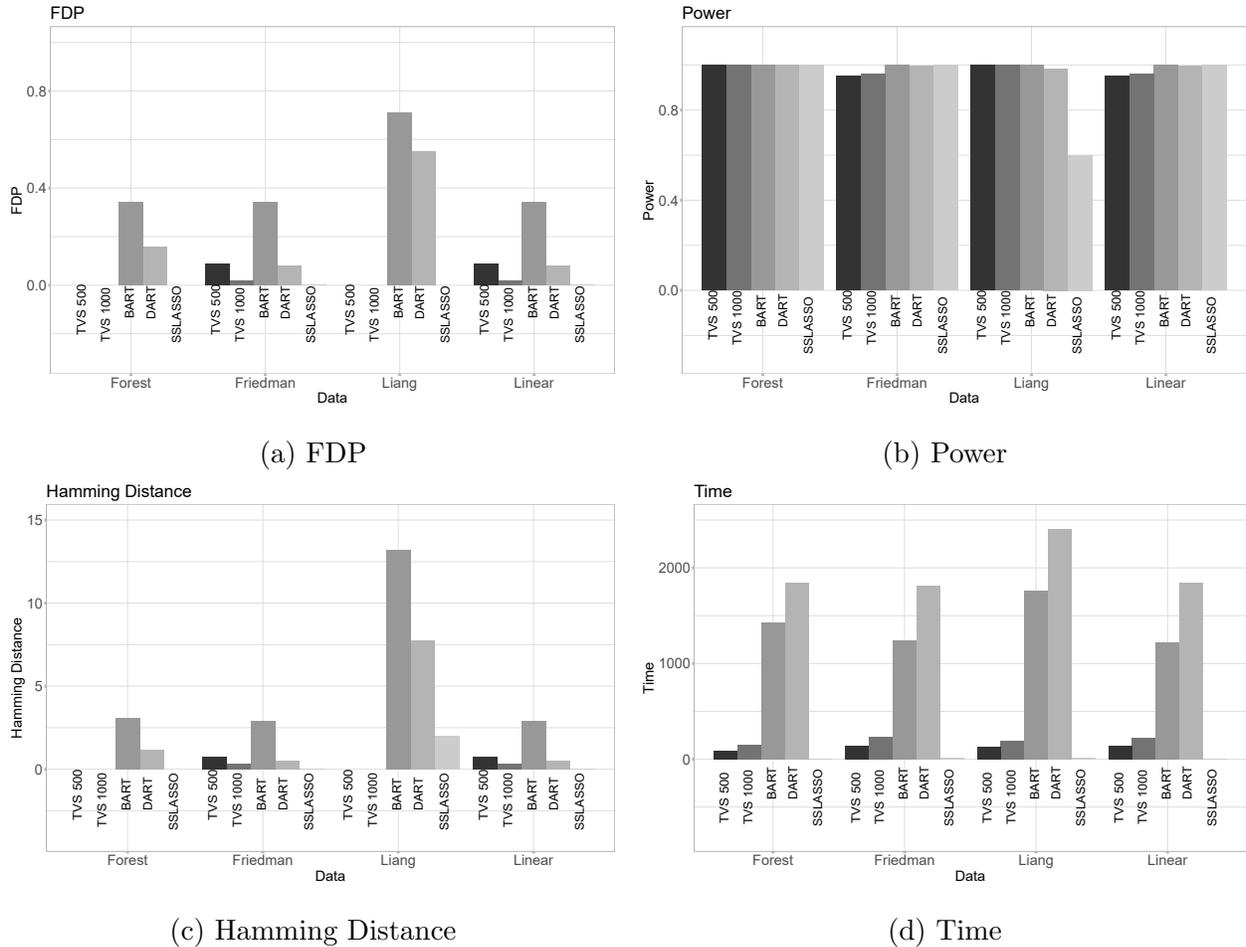


Figure 7: Graphs denoting FDP (7a), Power (7b), Hamming Distance (7c), and Time (7d) for the 4 choices of f_0 assuming $p = 1000$ and $n = 10000$. The x -axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1000$, both with $s = 1000$.

available⁷ 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 Barber et al. (2015) we analyze each drug separately. The response Y_i is given by the log-fold increase of lab-tested

⁷Stanford HIV Drug Resistance Database https://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/

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, (Rhee et al., 2005) 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 as a good approximation to the ground truth. Similarly as Barber et al. (2015), 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 $M = 1000$ inner BART iterations. In Figure 8, we differentiate those mutations whose position were identified by Rhee et al. (2005) and mutations which were not identified with blue and red colors, respectively. From the plot of inclusion probabilities in Figure 8a, 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 (Barber et al., 2015), LASSO (10-fold cross-validation), and Spike-and-Slab LASSO (Rockova and George (2018)), choosing $\lambda_1 = 0.1$ and $\lambda_0 \in \{\lambda_1 + 10 \times k; k = 0, 1, \dots, p\}$. 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 discovered Positions for each method are plotted in Figure 8b. 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 C).

⁸As suggested in the analysis of Barber et al. (2015), when analyzing each drug, only mutations that appear 3 or more times in the samples are taken into consideration.

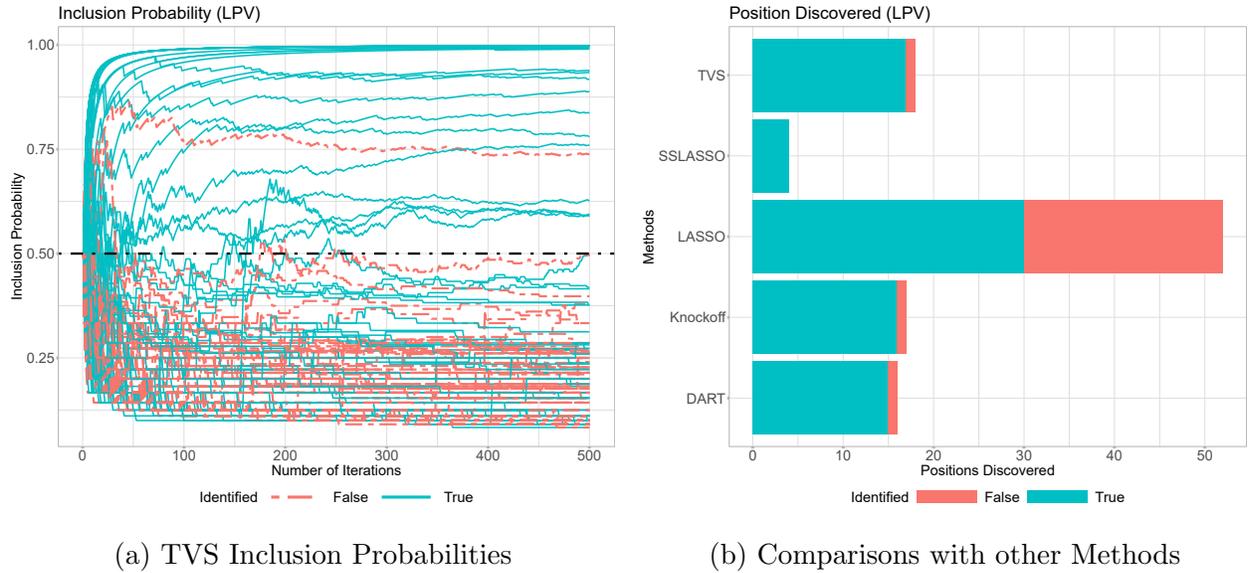


Figure 8: Thompson Variable Selection on the LPV dataset. (Left) Trajectories of the inclusion probabilities $\pi_j(t)$. (Right) Number of signals discovered, where blue denotes the experimentally validated signals and red are the unvalidated ones.

7.2 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, 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 $Y_i = \mathbb{I}(\text{December-sales}_i > 0)$ 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

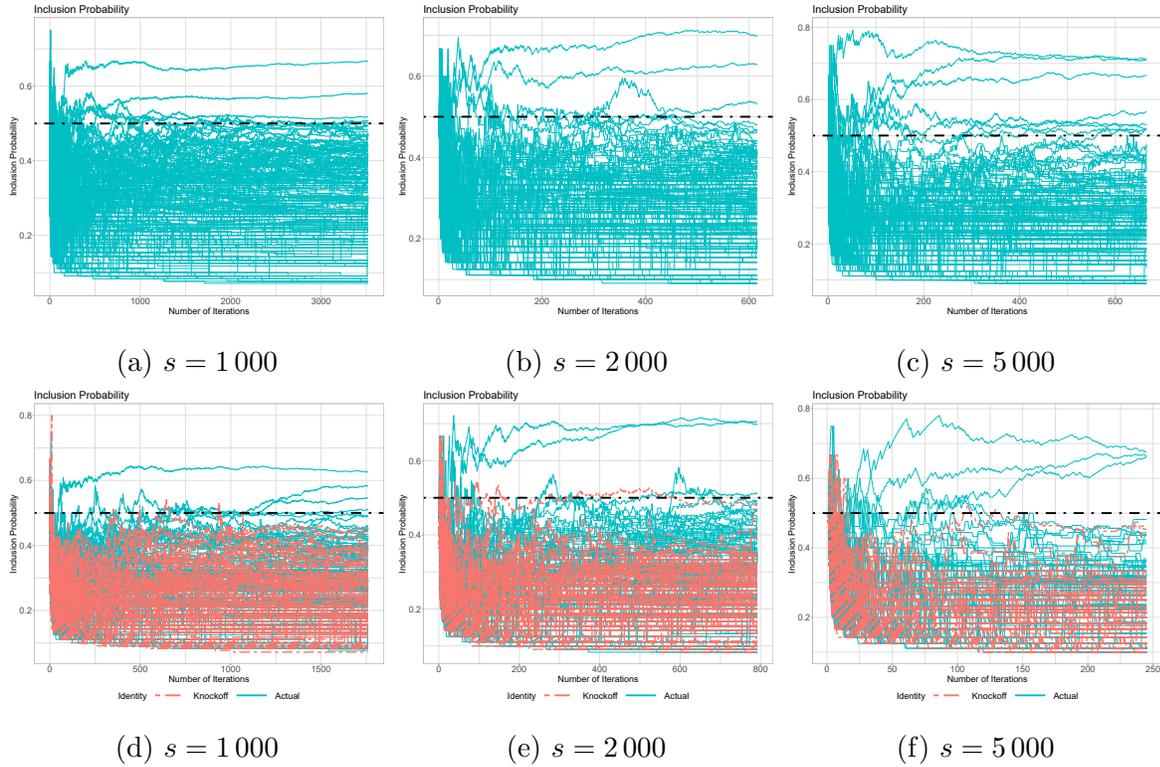


Figure 9: Thompson Variable Selection on the marketing data. (Left) Trajectories of the inclusion probabilities $\pi_j(t)$ without knockoffs. (Right) Trajectories of the inclusion probabilities $\pi_j(t)$ with knockoffs (in red).

with 114 variables whose names and descriptive statistics are reported in Section D 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 7.1, there is no proxy for the ground truth. To understand 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 (Patterson and Sesia (2018)) using a Gaussian Distributions with the same mean and covariance structure (Candes et al. (2018)). We run TVS with a batch size $s \in \{1000, 2000, 5000\}$ and $M = 1000$ inner iterations until the posterior probabilities have stabilized. The inclusion probabilities are plotted in Figure 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.

	$s = 1\,000$		$s = 2\,000$		$s = 5\,000$	
	Yes	No	Yes	No	Yes	No
Knockoff						
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 \times mail indicator	0.41	0.50	0.15	0.28	0.13	0.10
promo_nov period: total sales \times mail indicator	0.15	0.35	0.46	0.41	0.66	0.71
indicator of holiday mailer 2002 promotion response \times mail indicator	0.19	0.38	0.44	0.21	0.44	0.52

Table 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.

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.

8 Discussion

Our work pursues an intriguing connection between spike-and-slab variable selection and binary bandit problems. This pursuit has led 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, Liu et al. (2018) 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.

References

- Agrawal, R. (1995). Sample mean based index policies by $o(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability* 27(4), 1054–1078.
- Agrawal, S. and N. Goyal (2012). Analysis of Thompson Sampling for the Multi-armed Bandit Problem. In *Conference on Learning Theory*.
- Barber, R. F., E. J. Candès, et al. (2015). Controlling the false discovery rate via knockoffs. *The Annals of Statistics* 43(5), 2055–2085.
- Barbieri, M., J. O. Berger, E. I. George, and V. Rockova (2018). The median probability model and correlated variables. arXiv:1807.08336.

- Barbieri, M. M. and J. O. Berger (2004). Optimal predictive model selection. *Annals of Statistics* 32(3), 870–897.
- Bhattacharya, A., A. Chakraborty, and B. K. Mallick (2016). Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika* 103(4), 985.
- Bleich, J., A. Kapelner, E. George, and S. Jensen (2014). Variable selection for BART: An application to gene regulation. *Annals of Applied Statistics* 8(3), 1750–1781.
- Bottolo, L., S. Richardson, et al. (2010). Evolutionary stochastic search for bayesian model exploration. *Bayesian Analysis* 5(3), 583–618.
- Breiman, L. (2001). Random forests. *Machine learning* 45(1), 5–32.
- Brown, P. J., M. Vannucci, and T. Fearn (1998). Multivariate bayesian variable selection and prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60(3), 627–641.
- Candes, E., Y. Fan, L. Janson, and J. Lv (2018). Panning for gold:model-xknockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80(3), 551–577.
- Carbonetto, P., M. Stephens, et al. (2012). Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian analysis* 7(1), 73–108.
- Castillo, I., J. Schmidt-Hieber, A. Van der Vaart, et al. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics* 43(5), 1986–2018.
- Castro, J., D. Gómez, and J. Tejada (2009). Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research* 36(5), 1726–1730.
- Cesa-Bianchi, N. and G. Lugosi (2012). Combinatorial bandits. *Journal of Computer and System Sciences* 78(5), 1404–1422.
- Chen, W., Y. Wang, and Y. Yuan (2013). Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*.

- Chipman, H., E. I. George, and R. E. McCulloch (2001). The Practical Implementation of Bayesian Model Selection. In *Institute of Mathematical Statistics Lecture Notes - Monograph Series*. Institute of Mathematical Statistics.
- Chipman, H. A., E. I. George, and R. E. McCulloch (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics* 4(1), 266–298.
- Combes, R. and A. Proutiere (2014). Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*.
- Fisher, A., C. Rudin, and F. Dominici (2019). All models are wrong, but many are useful: Learning a variables importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177), 1–81.
- Foster, D. P. and R. A. Stine (2008). 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.
- Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics* 19(1), 1–141.
- Gai, Y., B. Krishnamachari, and R. Jain (2012). Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking* 20(5), 1466–1478.
- Garson, G. D. (1991). 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.
- George, E. I. and R. E. McCulloch (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association* 88(423), 881–889.
- George, E. I. and R. E. McCulloch (1997). Approaches for Bayesian variable selection. *Statistica Sinica* 7, 339–373.

- Gupta, S., G. Joshi, and O. Yağın (2020). Correlated multi-armed bandits with a latent random source. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3572–3576. IEEE.
- Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* 16(3), 709–732.
- Horel, E. and K. Giesecke (2019). Towards explainable AI: Significance tests for neural networks. arXiv:1902.06021.
- Ishwaran, H. et al. (2007). Variable importance in binary regression trees and forests. *Electronic Journal of Statistics* 1, 519–537.
- Jeřábek, E. (2004). Dual weak pigeonhole principle, boolean complexity, and derandomization. *Annals of Pure and Applied Logic* 129(1-3), 1–37.
- Jiang, T. and A. B. Owen (2003). Quasi-regression with shrinkage. *Mathematics and Computers in Simulation* 62(3-6), 231–241.
- Johnson, V. E. and D. Rossell (2012). Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association* 107(498), 649–660.
- Kazemitabar, J., A. Amini, A. Bloniarz, and A. S. Talwalkar (2017). Variable importance using decision trees. In *Advances in Neural Information Processing Systems*.
- Komiyama, J., J. Honda, and H. Nakagawa (2015). Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*.
- Kveton, B., Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson (2014). Matroid bandits: fast combinatorial optimization with learning. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 420–429.
- Kveton, B., Z. Wen, A. Ashkan, and C. Szepesvari (2015). Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems*.

- Lai, T. and H. Robbins (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6(1), 4–22.
- Lei, J., M. Gsell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association* 113(523), 1094–1111.
- Leike, J., T. Lattimore, L. Orseau, and M. Hutter (2016). Thompson Sampling is Asymptotically Optimal in General Environments. In *Conference on Uncertainty in Artificial Intelligence*. AUAI Press.
- Liang, F., Q. Li, and L. Zhou (2018). Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association* 113(523), 955–972.
- Linero, A. R. (2018). Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association* 113(522), 626–636.
- Linero, A. R. and Y. Yang (2018). Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80(5), 1087–1110.
- Liu, Y., V. Rockova, and Y. Wang (2018). ABC Variable Selection with Bayesian Forests. arXiv:1806.02304.
- Louppe, G., L. Wehenkel, A. Sutera, and P. Geurts (2013). Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*.
- Lu, Y., Y. Fan, J. Lv, and W. S. Noble (2018). DeepPINK: reproducible feature selection in Deep Neural Networks. In *Advances in Neural Information Processing Systems*.
- Mase, M., A. B. Owen, and B. Seiler (2019). Explaining black box decisions by shapley cohort refinement. arXiv:1911.00467.
- Mitchell, T. J. and J. J. Beauchamp (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association* 83(404), 1023–1032.

- Narisetty, N. N., X. He, et al. (2014). Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics* 42(2), 789–817.
- Olden, J. D. and D. A. Jackson (2002). Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling* 154(1-2), 135–150.
- Owen, A. B. and C. Prieur (2017). On shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification* 5(1), 986–1002.
- Pandey, S., D. Chakrabarti, and D. Agarwal (2007). Multi-armed bandit problems with dependent arms. In *International Conference on Machine learning*.
- Patterson, E. and M. Sesia (2018). *knockoff: The Knockoff Filter for Controlled Variable Selection*. Statistics Department, Stanford University. R package version 0.3.2.
- Rhee, S.-Y., W. J. Fessel, A. R. Zolopa, L. Hurley, T. Liu, J. Taylor, D. P. Nguyen, S. Slome, D. Klein, M. Horberg, et al. (2005). 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.
- Rhee, S.-Y., J. Taylor, G. Wadhera, A. Ben-Hur, D. L. Brutlag, and R. W. Shafer (2006). Genotypic predictors of Human Immunodeficiency Virus type 1 drug resistance. *Proceedings of the National Academy of Sciences* 103(46), 17355–17360.
- Rockova, V. and E. I. George (2014). EMVS: The EM Approach to Bayesian Variable Selection. *Journal of the American Statistical Association* 109(506), 828–846.
- Rockova, V. and E. I. George (2018). The spike-and-slab LASSO. *Journal of the American Statistical Association* 113(521), 431–444.
- Rossell, D. and D. Telesca (2017). Nonlocal priors for high-dimensional estimation. *Journal of the American Statistical Association* 112(517), 254–265.
- Sabes, P. N. and M. I. Jordan (1996). Reinforcement learning by probability matching. In *Advances in Neural Information Processing Systems*.

- Scott, S. L. (2010). A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26(6), 639–658.
- Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games* 2(28), 307–317.
- Song, E., B. L. Nelson, and J. Staum (2016). Shapley effects for global sensitivity analysis: Theory and Computation. *SIAM/ASA Journal on Uncertainty Quantification* 4(1), 1060–1083.
- Sundararajan, M. and A. Najmi (2019). The many shapley values for model explanation. arXiv:1908.08474.
- Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two sample. *Biometrika* 25(3/4), 285–294.
- Vannucci, M. and F. C. Stingo (2010). Bayesian models for variable selection that incorporate biological information. *Bayesian Statistics* 9, 1–20.
- Wang, S. and W. Chen (2018). Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pp. 5114–5122.
- Ye, M. and Y. Sun (2018). Variable selection via penalized neural network: a drop-out-one loss approach. In *International Conference on Machine Learning*.
- Zhang, T., S. S. Ge, and C. C. Hang (2000). Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. *Automatica* 36(12), 1835–1846.

Supplementary Material

A Proof of Theorem 1

We will denote with $\mathcal{S}^* = \arg \max_{\mathcal{S}} r_{\theta^*}^C(\mathcal{S})$ the optimal model where

$$r_{\theta^*}^C(\mathcal{S}) = \sum_{i \in \mathcal{S}} \left[\theta_i^*(\mathcal{S}) \log \left(\frac{C+1}{C} \right) - \log \left(\frac{1}{C} \right) \right].$$

First, we define the reward gap of a set of arms \mathcal{S}_t as

$$\Delta_{\mathcal{S}_t} = \mathbb{E}[r_{\theta^*}^C(\mathcal{S}^*) - r_{\theta}^C(\mathcal{S}_t)].$$

and write the expected regret in (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,

$$\hat{\mu}_i(t) = \frac{a_i(t) - 1}{N_i(t)} = \frac{1}{N_i(t)} \sum_{k < t} \mathbb{I}(i \in \mathcal{S}_k) \gamma_i^t \quad (17)$$

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. $\gamma_i^t = 1$ when $i \in \mathcal{S}_t$. We will be using the following usual Chernoff-Hoeffding bounds which we state without a proof.

Lemma 4 (Chernoff-Hoeffding Bound) *Let X_1, \dots, 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 (Komiya et al., 2015; Wang and Chen, 2018) we will bound the expected regret separately on the intersection of combinations of

the following events:

$$\begin{aligned}\mathcal{A}(t) &= \{\mathcal{S}_t \neq \mathcal{S}^*\}, \\ \mathcal{B}(t) &= \left\{ \exists i \in \mathcal{S}^* \text{ s.t. } \hat{\mu}_i(t) < 0.5 + \frac{\alpha}{2} \text{ or } \exists i \in \mathcal{S}_t \setminus \mathcal{S}^* \text{ s.t. } \hat{\mu}_i(t) > 0.5 - \frac{\alpha}{2} \right\}, \\ \mathcal{C}(t) &= \left\{ \exists i \in \mathcal{S}^* \text{ s.t. } \hat{\mu}_i(t) - \theta_i(t) > \frac{\alpha}{2} \text{ or } \exists i \in \mathcal{S}_t \setminus \mathcal{S}^* \text{ s.t. } \theta_i(t) - \hat{\mu}_i(t) > \frac{\alpha}{2} \right\}, \\ \mathcal{D}(t) &= \bigcap_{i \in \mathcal{S}_t} \left\{ N_i(t) > \frac{8 \log T}{\alpha^2} \right\},\end{aligned}$$

where α occurred in Assumption 1. First, we focus on the following term

$$Reg_1(T) = \sum_{t=1}^T \mathbb{E}[\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}(t))]. \quad (18)$$

The following Lemma finds an upper bound on $Reg_1(T)$:

Lemma 5 *Under the Assumption 1, the TVS sampling policy in Table 1 with $C = (\sqrt{5} - 1)/2$ yields, for $\Delta_{max} = \max_{\mathcal{S}} \Delta_{\mathcal{S}}$*

$$Reg_1(T) \leq \Delta_{max} \sum_{i \in \mathcal{S}^*} \mathbb{E} \left(\sum_{t=1}^T \mathbb{I} \left\{ i \in \mathcal{S}_t, \hat{\mu}_i(t) < 0.5 + \frac{\alpha}{2} \right\} \right) \quad (19)$$

$$+ \Delta_{max} \sum_{i \notin \mathcal{S}^*} \mathbb{E} \left(\sum_{t=1}^T \mathbb{I} \left\{ i \in \mathcal{S}_t, \hat{\mu}_i(t) > 0.5 - \frac{\alpha}{2} \right\} \right) \quad (20)$$

$$\leq \Delta_{max} \times p \left(1 + \frac{4}{\alpha_{max}^2} \right). \quad (21)$$

Proof: We will first prove that

$$\mathbb{E} \left(\sum_{t=1}^T \mathbb{I} \left\{ i \in \mathcal{S}_t, \hat{\mu}_i(t) < 0.5 + \frac{\alpha}{2} \right\} \right) \leq 1 + \frac{4}{\alpha_{max}^2}. \quad (22)$$

The second inequality for the term in (38) can be obtained analogously. With $\theta_i(\mathcal{S}_t)$ as in (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 $i \in \mathcal{S}^*$ 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 τ_1, τ_2, \dots the times such that $i \in \mathcal{S}_t$, define $\tau_0 = 0$ and write

$$\begin{aligned}
\mathbb{E} \left(\sum_{t=1}^T \mathbb{I}\{i \in \mathcal{S}_t, \hat{\mu}_i(t) < 0.5 + \alpha/2\} \right) &\leq 1 + \sum_{w=0}^T \mathbb{P}(\hat{\mu}_i(\tau_w) < 0.5 + \alpha/2, N_i(t) = w) \\
&\leq 1 + \sum_{w=0}^T \exp(-w d(0.5 + \alpha/2, \bar{\theta}(T))) \\
&\leq 1 + \sum_{w=0}^T \exp(-w d(0.5 + \alpha/2, 0.5 + \alpha)) \\
&\leq 1 + \sum_{w=0}^{\infty} \exp(-w d(0.5 + \alpha/2, 0.5 + \alpha)) \\
&\leq 1 + \frac{\exp(-d(0.5 + \alpha/2, 0.5 + \alpha))}{1 - \exp(-d(0.5 + \alpha/2, 0.5 + \alpha))} \\
&\leq 1 + \frac{4}{\alpha^2}.
\end{aligned}$$

This concludes the proof of (45). The second term can be bounded analogously, which concludes the proof of the Lemma. \square

Next, we focus on the following term

$$\text{Reg}_2(T) = \sum_{t=1}^T \mathbb{E}[\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t))]. \quad (23)$$

To bound this term, we will be using the following Lemma (Lemma 4 from Wang and Chen (2018)) which we, again, state without a proof.

Lemma 6 *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 1. Using the TVS algorithm from Table 1, we have the following two inequalities for any base arm i :*

$$\begin{aligned}
\mathbb{P} \left[\theta_i(t) - \hat{\mu}_i(t) > \sqrt{\frac{2 \log T}{N_i(t)}} \right] &\leq \frac{1}{T}, \\
\mathbb{P} \left[\hat{\mu}_i(t) - \theta_i(t) > \sqrt{\frac{2 \log T}{N_i(t)}} \right] &\leq \frac{1}{T}.
\end{aligned}$$

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 Wang and Chen (2018).

The following lemma bounds the regret term (23).

Lemma 7 *Using the TVS algorithm from Table 1, we have the following bound:*

$$\text{Reg}_2(T) = \sum_{t=1}^T \mathbb{E}[\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t))] \leq \Delta_{\max} \times p.$$

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

$$\left\{ \exists i \in \mathcal{S}_t \setminus \mathcal{S}^* : \theta_i(t) - \hat{\mu}_i(t) > \sqrt{\frac{2 \log T}{N_i(t)}} \text{ or } \exists i \in \mathcal{S}_t \cap \mathcal{S}^* : \hat{\mu}_i(t) - \theta_i(t) > \sqrt{\frac{2 \log T}{N_i(t)}} \right\}.$$

We can then directly apply Lemma 6 to write

$$\begin{aligned} \sum_{t=1}^T \Delta_{\mathcal{S}_t} \mathbb{P}(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)) &\leq \Delta_{\max} \sum_{t=1}^T \mathbb{P}(\mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)) \\ &\leq \Delta_{\max} \sum_{t=1}^T p/T. \quad \square \end{aligned}$$

Finally, we focus on the following term

$$\text{Reg}_3 = \sum_{t=1}^T \mathbb{E}(\Delta_{\mathcal{S}_t} \times \mathbb{I}(\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}^c(t))). \quad (24)$$

Lemma 8 *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*

$$f_i^*(j, s) \leq \binom{j}{s} (0.5 + \alpha)^s (0.5 - \alpha)^{j-s} \quad (25)$$

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 $p_j = \mathbb{P}(\gamma_i^{\tau_j} = 1 | \mathcal{S}_{\tau_j})$ and, for a given j and s write $f_{j,s}^*(p_1, \dots, p_j) := f_i^*(j, s)$. 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 $\mathbf{B} = (B_1, B_2, B_3, \dots, B_j)' \in \{0, 1\}^j$ 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 $|\mathbf{B}| = \sum_{k=1}^j B_k$, we can write

$$f_{j,s}^*(p_1, \dots, p_j) = \sum_{\mathbf{B}: |\mathbf{B}|=s} \prod_{l=1}^j p_l^{B_l} (1 - p_l)^{1-B_l}.$$

We want to show that $\mathbf{p}^* = (p_1^*, \dots, p_j^*)' = \arg \max f_{j,s}^*(p_1, \dots, p_j)$ when $p_i^* = 0.5 + \alpha$ for all $1 \leq i \leq j$. First, we notice is that this is a multi-linear polynomial in the sense that $\frac{\partial f_{j,s}^*(p_1, \dots, p_j)}{\partial p_k}$ is independent of p_k . Keeping every other coordinate constant, the value p_k^* that maximizes $f_{j,s}^*(\cdot)$ in the k^{th} direction has to be either $0.5 + \alpha$ or 1. The vector $(p_1^*, \dots, p_j^*)'$ maximizing $f_{j,s}^*(p_1, \dots, p_j)$ 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 $p_k^* = 1$ and $j - r$ be the number of coordinates k for which $p_k^* = 0.5 + \alpha$ (notice that $r \leq s$). Since $f_{j,s}^*(p_1, \dots, p_j)$ 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 $p_1^* = p_2^* = \dots = p_r^* = 1$ and $p_{r+1}^* = p_{r+2}^* = \dots = p_j^* = 0.5 + \alpha$. In this case, we have the constraint on the binary indicators \mathbf{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

$$f_{j,s,r}^*(\mathbf{p}^*) = \binom{j-r}{s-r} (0.5 + \alpha)^{s-r} (0.5 - \alpha)^{j-s}.$$

It is sufficient to prove that this function is maximized at $r = 0$. We have

$$\begin{aligned} \frac{f_{j,s,r+1}^*(\mathbf{p}^*)}{f_{j,s,r}^*(\mathbf{p}^*)} &= \frac{\binom{j-r-1}{s-r-1} (0.5 + \alpha)^{s-r-1} (0.5 - \alpha)^{j-s}}{\binom{j-r}{s-r} (0.5 + \alpha)^{s-r} (0.5 - \alpha)^{j-s}} = \frac{s-r}{(j-r)(1/2 + \alpha)} \\ &\leq \frac{j/2 - r}{(j-r)(1/2 + \alpha)} \quad (\text{using the assumption } s \leq \lfloor j/2 \rfloor) \\ &= 1 - \frac{(1/2 - \alpha)r + \alpha j}{(j-r)(1/2 + \alpha)} < 1 \end{aligned}$$

since $1/2 - \alpha \geq 0$ and $\alpha > 0$. Since this is true for all r , the function $\frac{f_{j,s,r+1}^*(p_1, \dots, p_j)}{f_{j,s,r}^*(p_1, \dots, p_j)}$ is maximized at $r = 0$. This concludes the proof. \square

Lemma 9 *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*

$$\mathbb{E} [\tau_{j+1} - \tau_j] \leq \begin{cases} 4 + \frac{1}{\alpha} & \text{when } j \leq \frac{8}{\alpha} \\ 1 + \frac{1}{e^{\alpha^2 j/4} - 1} + e^{-\alpha^2 j/2} \left(C_1 + C_2 \frac{1-2\alpha}{4\alpha^2(j+1)} \right) & \text{when } j > \frac{8}{\alpha}, \end{cases} \quad (26)$$

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

$$\theta_i(t) \sim \mathcal{B}[a_i(t), b_i(t)],$$

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

$$\mathbb{E}[\tau_{j+1} - \tau_j | \hat{\mu}_i(\tau_j^i)] = \frac{1}{\mathbb{P}(B_{ij} > 0.5)} = \frac{1}{p_{i,j}(0.5)}$$

where $B_{ij} \sim \mathcal{B}[a_i(\tau_{j+1}), b_i(\tau_{j+1})]$. 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 Agrawal and Goyal (2012)) 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:

$$F_{\alpha,\beta}^{\text{beta}}(y) = 1 - F_{\alpha+\beta-1,y}(\alpha - 1) \quad (27)$$

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 (27) 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

$$\mathbb{E}\left[\frac{1}{p_{i,j}(0.5)}\right] = \sum_{s=0}^j \frac{f_i^*(j, s)}{F_{j+1,0.5}(s)}. \quad (28)$$

First, we consider the case when $j \leq \frac{8}{\alpha}$. In the following calculations, we will use the result from Lemma 8. 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

$$\mathbb{E} \left[\frac{1}{p_{i,j}(0.5)} \right] \leq 2 \sum_{s=0}^j \frac{f_i^*(j, s)}{F_{j,0.5}(s)} \quad (29)$$

$$\leq 2 \sum_{s=0}^{\lfloor j/2 \rfloor} \frac{\binom{j}{s} p_{\max}^s (1 - p_{\max})^{j-s}}{f_{j,0.5}(s)} + 4 \sum_{s=\lceil j/2 \rceil}^j f_i^*(j, s) \quad (30)$$

$$\leq 2 \sum_{s=0}^{\lfloor j/2 \rfloor} \frac{p_{\max}^s (1 - p_{\max})^{j-s}}{1/2^j} + 4 \quad (31)$$

$$\leq 2 \frac{(1 - p_{\max})^j}{1/2^j} \sum_{s=0}^{\lfloor j/2 \rfloor} R^s + 4 \quad (32)$$

$$\leq 2 \left(\frac{R^{\lfloor j/2 \rfloor + 1} - 1}{R - 1} \right) \frac{(1 - p_{\max})^j}{1/2^j} + 4 \quad (33)$$

$$\leq 2 \left(\frac{R}{R - 1} \right) R^{j/2} \frac{(1 - p_{\max})^j}{1/2^j} + 4 \quad (34)$$

$$\leq \frac{1}{\alpha} e^{-j d(1/2, p_{\max})} + 4 \quad (35)$$

$$\leq \frac{1}{\alpha} + 4, \quad (36)$$

where from (34) to (35) 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

$$\begin{aligned} \frac{(1 - p_{\max})^j}{1/2^j} R^{\lfloor j/2 \rfloor} &\leq \frac{(1 - p_{\max})^j}{1/2^j} R^{j/2} = e^{j \log(1 - p_{\max}) - j \log(1/2) + j/2 \log(p_{\max}) - j/2 \log(1 - p_{\max})} \\ &= e^{-j \left\{ \frac{1}{2} \log\left(\frac{1}{2}/p_{\max}\right) + \frac{1}{2} \log\left[\frac{1}{2}/(1 - p_{\max})\right] \right\}} = e^{-j d(1/2, p_{\max})} \leq e^{-\alpha^2 j}. \end{aligned}$$

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

$$\Sigma(0, \lfloor j/2 \rfloor - 1) \leq c_2 \left[e^{-\alpha^2 j} \frac{1 - 2\alpha}{4\alpha^2(j+1)} \right] + c_3 e^{-2\alpha^2 j} \quad (37)$$

$$\Sigma(\lfloor j/2 \rfloor, \lfloor j/2 \rfloor) \leq 3 e^{-\alpha^2 j} \quad (38)$$

$$\Sigma(\lceil j/2 \rceil, \lfloor (1/2 + \alpha/2)j \rfloor) \leq c_3 \left(e^{-\alpha^2 j/2} \right) \quad (39)$$

$$\Sigma(\lceil (1/2 + \alpha/2)j \rceil, j) \leq 1 + \frac{1}{e^{\alpha^2 j/4} - 1}, \quad (40)$$

where $c_2 > 0$ and $c_3 > 0$ are constants unrelated to Algorithm 1. This will complete the proof. We now prove the bounds in the last display. We start with the first inequality in

(37). When $s \leq (j+1)/2 - \sqrt{(j+1)/4}$, we use the following bound for the Binomial CDF (Jeřábek (2004))

$$F_{j+1,0.5}(s) \geq \frac{1}{c_2} \left[\frac{j+1-s}{j+1-2s} \binom{j+1}{s} \frac{1}{2^{j+1}} \right],$$

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 8 and denoting $p_{\max} = 1/2 + \alpha$ and $R = \frac{p_{\max}}{1-p_{\max}}$)

$$\begin{aligned} \Sigma(0, \lfloor j/2 \rfloor - 1) &\leq c_2 \sum_{s=0}^{\lceil (j+1)/2 - \sqrt{(j+1)/4} \rceil} \frac{f_i^*(j, s)}{\frac{j+1-s}{j+1-2s} \binom{j+1}{s} \frac{1}{2^{j+1}}} + c_3 \sum_{s=\lceil (j+1)/2 - \sqrt{(j+1)/4} \rceil+1}^{\lfloor j/2 \rfloor - 1} f_i^*(j, s) \\ &\leq c_2 \sum_{s=0}^{\lfloor j/2 \rfloor - 1} \frac{f_i^*(j, s)}{\frac{j+1-s}{j+1-2s} \binom{j+1}{s} \frac{1}{2^{j+1}}} + c_3 \sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j, s) \\ &\leq c_2 \left(\frac{(1-p_{\max})^j}{1/2^{j+1}} \sum_{s=0}^{\lfloor j/2 \rfloor - 1} \left(1 - \frac{2s}{j+1} \right) R^s \right) + c_3 \sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j, s). \end{aligned} \quad (41)$$

Now we bound the first term in (41) to obtain

$$\begin{aligned} &\frac{(1-p_{\max})^j}{1/2^{j+1}} \sum_{s=0}^{\lfloor j/2 \rfloor - 1} \left(1 - \frac{2s}{j+1} \right) R^s \\ &= \frac{(1-p_{\max})^j}{1/2^{j+1}} \left\{ \frac{R^{\lfloor j/2 \rfloor} - 1}{R-1} - \frac{2}{j+1} \left[\frac{(\lfloor j/2 \rfloor - 1)R^{\lfloor j/2 \rfloor}}{R-1} - \frac{R^{\lfloor j/2 \rfloor} - R}{(R-1)^2} \right] \right\} \\ &\leq \frac{(1-p_{\max})^j}{1/2^{j+1}} \left\{ \frac{R^{\lfloor j/2 \rfloor}}{R-1} - \frac{2}{j+1} \left[\frac{(\lfloor j/2 \rfloor - 1)R^{\lfloor j/2 \rfloor}}{R-1} - \frac{R^{\lfloor j/2 \rfloor}}{(R-1)^2} \right] \right\} \\ &\leq \frac{(1-p_{\max})^j}{1/2^{j+1}} \left[\frac{2}{j+1} \frac{R^{\lfloor j/2 \rfloor}}{(R-1)^2} + \frac{2[(j+1)/2 - \lfloor j/2 \rfloor + 1]}{j+1} \frac{R^{\lfloor j/2 \rfloor}}{R-1} \right] \\ &\leq \frac{(1-p_{\max})^j}{1/2^{j+1}} \frac{6}{j+1} \frac{R^{\lfloor j/2 \rfloor + 1}}{(R-1)^2} \\ &\leq e^{-\alpha^2 j} \frac{12}{j+1} \frac{R}{(R-1)^2}, \end{aligned}$$

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)

$$\frac{(1-p_{\max})^j}{1/2^j} R^{\lfloor j/2 \rfloor} \leq e^{-j d(1/2, p_{\max})} \leq e^{-\alpha^2 j}.$$

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 (41), 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 4 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, $\sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j, s) \leq e^{-2\alpha^2 j}$. We put the two terms together to finally obtain

$$\Sigma(0, \lfloor j/2 \rfloor - 1) \leq c_2 \left[e^{-\alpha^2 j} \frac{1-2\alpha}{4\alpha^2(j+1)} \right] + c_3 \sum_{s=0}^{\lfloor j/2 \rfloor - 1} f_i^*(j, s) \quad (42)$$

$$\leq c_2 \left[e^{-\alpha^2 j} \frac{1-2\alpha}{4\alpha^2(j+1)} \right] + c_3 e^{-2\alpha^2 j}. \quad (43)$$

Next, to bound the term $\Sigma(\lfloor j/2 \rfloor, \lfloor j/2 \rfloor)$ in (38), we use Lemma 8 and the fact that $p_{\max} > 1/2$ to find that for $s = \lfloor j/2 \rfloor$

$$\begin{aligned} \Sigma(\lfloor j/2 \rfloor, \lfloor j/2 \rfloor) &= \frac{f_i^*(j, s)}{F_{j+1, 0.5}(s)} \leq \frac{f_i^*(j, s)}{f_{j+1, 0.5}(s)} \leq 2 \left(1 - \frac{s}{j+1} \right) R^s \left(\frac{1-p_{\max}}{1/2} \right)^j \\ &\leq \frac{2}{j+1} \left(\frac{j}{2} + 2 \right) R^{j/2} \left(\frac{1-p_{\max}}{1/2} \right)^j \\ &\leq \left(1 + \frac{3}{j+1} \right) e^{-\alpha^2 j} \leq 2 e^{-\alpha^2 j}, \end{aligned}$$

where we used the assumption $j \geq 1/\alpha > 2$.

In order to bound the third term $\Sigma(\lceil j/2 \rceil, \lceil (\alpha+1)j/2 \rceil)$ in (39), 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

Jeřábek (2004) is satisfied and we can apply the bound $F_{j+1,0.5}(s) \geq \frac{1}{c_3}$. Then we have

$$\begin{aligned} \Sigma(\lceil j/2 \rceil, \lfloor (\alpha + 1)j/2 \rfloor) &\leq c_3 \left(\sum_{s=\lceil j/2 \rceil}^{\lfloor (\alpha+1)j/2 \rfloor} f_i^*(j, s) \right) \leq c_3 \left(\sum_{s=0}^{\lfloor (\alpha+1)j/2 \rfloor} f_i^*(j, s) \right) \\ &\leq c_3 \left(e^{-\alpha^2 j/2} \right), \end{aligned}$$

where the last inequality stems from the Chernoff-Hoeffding inequality in Lemma 4 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 (40) we use the Hoeffding inequality in Lemma 4 with $\lambda = (\alpha + 1)j/[2(j + 1)] - 1/2 \leq s/(j + 1) - 1/2$ to find (for a r.v. $X \sim \text{Bin}(j + 1, 1/2)$) that

$$\begin{aligned} F_{j+1,0.5}(s) &\geq 1 - \mathbb{P} \left(\frac{X}{j+1} - \frac{1}{2} > \lambda \right) \geq 1 - e^{-2(j+1)\lambda^2} = 1 - e^{(-\frac{j\alpha^2}{2} + \frac{j\alpha^2}{2(j+1)} + \alpha \frac{j}{j+1} - \frac{1}{2(j+1)})} \\ &\geq 1 - e^{-\alpha^2 j/2}, \end{aligned}$$

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

$$\Sigma(\lceil (\alpha + 1)j/2 \rceil, j) \leq \sum_{s=\lceil (\alpha+1)j/2 \rceil}^j \frac{f_i^*(j, s)}{F_{j+1,0.5}(s)} \leq \frac{1}{1 - e^{-\alpha^2 j/4}} = 1 + \frac{1}{e^{\alpha^2 j/4} - 1}.$$

Now, denoting $C_1 = 2 + 2c_3$ and $C_2 = c_2$ we get the statement in the Lemma. This concludes our proof. \square

Using Lemma 9, we can achieve a similar bound in Lemma 6 of Wang and Chen (2018),

Lemma 10 *Under Assumption 1, the TVS Algorithm 1 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*

$$\left\lceil \frac{8 \log(T)}{\alpha^2} \right\rceil + \left\lceil \frac{8}{\alpha} \right\rceil \left(3 + \frac{1}{\alpha} \right) + \tilde{C} \frac{e^{-4\alpha}}{1 - e^{-\alpha^2/2}} + \frac{8}{\alpha^2} \frac{1}{e^{2\alpha} - 1} + \frac{e^{-1}}{1 - e^{-\alpha/8}},$$

where $\tilde{C} = C_1 + C_2 \frac{1-2\alpha}{32\alpha}$ for some $C_1 > 0$ and $C_2 > 0$ not related to the Algorithm 1.

Proof: We use the notation from Lemma 9, where τ_j^i is the time when the arm i has been pulled for the j^{th} time. Denoting with $\tilde{T} = \lfloor \frac{8 \log(T)}{\alpha^2} \rfloor$, we want to find an upper bound for $\mathbb{E}[\tau_{\tilde{T}}^i]$. We can write

$$\mathbb{E}[\tau_{\tilde{T}}^i] = \sum_{j=0}^{\tilde{T}} \mathbb{E}[\tau_{j+1}^i - \tau_j^i] = \sum_{j=0}^{\tilde{T}} \mathbb{E}\left(\frac{1}{p_{i,j}(0.5)}\right)$$

and using Lemma 9 we obtain

$$\mathbb{E}[\tau_{\tilde{T}}^i] \leq \sum_{j=0}^{\lfloor \frac{8}{\alpha} \rfloor} \left(4 + \frac{1}{\alpha}\right) + \sum_{j=\lfloor \frac{8}{\alpha} \rfloor + 1}^{\tilde{T}} \left[1 + \frac{1}{e^{\alpha^2 j/4} - 1} + e^{-\alpha^2 j/2} \left(C_1 + C_2 \frac{1 - 2\alpha}{4\alpha^2(j+1)}\right)\right]. \quad (44)$$

First, we note that $\sum_{j=0}^{\lfloor \frac{8}{\alpha} \rfloor} (4 + \frac{1}{\alpha}) + \sum_{j=\lfloor \frac{8}{\alpha} \rfloor + 1}^{\tilde{T}} 1 \leq \lceil \frac{8 \log(T)}{\alpha^2} \rceil + \lceil \frac{8}{\alpha} \rceil (3 + \frac{1}{\alpha})$. Next, we write

$$\sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\tilde{T}} e^{-\frac{j\alpha^2}{2}} \leq \frac{e^{-4\alpha}}{1 - e^{-\alpha^2/2}}$$

and

$$\sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\tilde{T}} e^{-\frac{j\alpha^2}{2}} \left(C_1 + C_2 \frac{1 - 2\alpha}{4\alpha^2(j+1)}\right) < \sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\tilde{T}} e^{-\frac{j\alpha^2}{2}} \left(C_1 + C_2 \frac{1 - 2\alpha}{32\alpha}\right) \leq \tilde{C} \frac{e^{-4\alpha}}{1 - e^{-\alpha^2/2}}.$$

Finally, we use the fact that $\frac{1}{e^x - 1} \leq e^{-x/2}$ for $x \geq 1$ to obtain

$$\begin{aligned} \sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\tilde{T}} \frac{1}{e^{\frac{j\alpha^2}{4}} - 1} &\leq \sum_{j=\lceil \frac{8}{\alpha} \rceil + 1}^{\lfloor \frac{8}{\alpha^2} \rfloor} \frac{1}{e^{\frac{j\alpha^2}{4}} - 1} + \sum_{j=\lceil \frac{8}{\alpha^2} \rceil}^{\lfloor \frac{8 \log(T)}{\alpha^2} \rfloor} \frac{1}{e^{\frac{j\alpha^2}{4}} - 1} \\ &\leq \frac{8}{\alpha^2} \frac{1}{e^{2\alpha} - 1} + \sum_{j=\lceil \frac{8}{\alpha^2} \rceil}^{\lfloor \frac{8 \log(T)}{\alpha^2} \rfloor} e^{-\alpha^2 j/8} \leq \frac{8}{\alpha^2} \frac{1}{e^{2\alpha} - 1} + \frac{e^{-1}}{1 - e^{-\alpha^2/8}} \quad \square \end{aligned}$$

Using these lemmas we can prove the following lemma about $Reg_3(T)$.

Lemma 11 *We denote with $Reg_3(T)$ the regret term in (24). Under Assumption 1, the TVS Algorithm 1 with $C = (\sqrt{5} - 1)/2$ yields*

$$Reg_3(T) \leq \Delta_{\max} \left\{ \frac{8p \log(T)}{\alpha^2} + \tilde{C} q^* \frac{e^{-4\alpha}}{1 - e^{-\alpha^2/2}} + q^* \left[\frac{8}{\alpha^2} \frac{1}{e^{2\alpha} - 1} + \frac{e^{-1}}{1 - e^{-\alpha/8}} + \left\lceil \frac{8}{\alpha} \right\rceil \left(3 + \frac{1}{\alpha}\right) \right] \right\},$$

where $\tilde{C} = C_1 + C_2 \frac{1-2\alpha}{32\alpha}$ for some $C_1 > 0$ and $C_2 > 0$ not related to the Algorithm 1.

Proof: We start with the following facts $Reg_3(T) \leq \sum_{t=1}^T \Delta_{\mathcal{S}_t} \mathbb{E} [\mathbb{I}(\mathcal{A}(t) \cap \mathcal{D}^c(t))]$ 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

$$Reg_3(T) \leq \sum_{t=1}^T \left[\sum_{i \in \mathcal{S}_t \cap \mathcal{S}^*} \Delta_{\mathcal{S}_t} \mathbb{E} \left[\mathbb{I} \left(\mathcal{A}(t) \cap \left\{ N_i(t) \leq \frac{8 \log(T)}{\alpha^2} \right\} \right) \right] \right] \quad (45)$$

$$+ \sum_{t=1}^T \left[\sum_{i \in \mathcal{S}_t \setminus \mathcal{S}^*} \Delta_{\mathcal{S}_t} \mathbb{E} \left[\mathbb{I} \left(\mathcal{A}(t) \cap \left\{ N_i(t) \leq \frac{8 \log(T)}{\alpha^2} \right\} \right) \right] \right]. \quad (46)$$

If $i \in \mathcal{S}_t \setminus \mathcal{S}^*$, 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 (46) can be thus bounded by $(p - q^*) \Delta_{max} \frac{8 \log(T)}{\alpha^2}$.

If $i \in \mathcal{S}^* \cap \mathcal{S}_t$, then the arm i contributes to the regret when $\mathcal{S}_t \neq \mathcal{S}^*$. However, by Lemma 9 and Lemma 10 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 $i \in \mathcal{S}^* \cap \mathcal{S}_t$ is bounded by $\Delta_{max} \left(\left\lceil \frac{8 \log(T)}{\alpha^2} \right\rceil + \left\lceil \frac{8}{\alpha} \right\rceil \left(3 + \frac{1}{\alpha} \right) + \tilde{C} \frac{e^{-4\alpha}}{1 - e^{-\alpha^2/2}} + \frac{8}{\alpha^2} \frac{1}{e^{2\alpha} - 1} + \frac{e^{-1}}{1 - e^{-\alpha/8}} \right)$. Because there are q^* signal arms, we can combine (45) and (46) to arrive at the bound in the statement of this lemma. \square

We now put the various pieces together to finally prove Theorem 1.

Proof: We start by noticing that

$$\begin{aligned} \mathbb{I}[\mathcal{A}(t)] &= \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t)] \\ &= \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t)] + \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}^c(t)]. \end{aligned}$$

Now we note that $\mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}^c(t)] = 0$ because

$$\mathcal{B}^c(t) \cap \mathcal{C}^c(t) = \{ \forall i \in \mathcal{S}^* \text{ we have } \theta_i(t) > 0.5 \text{ and } \forall i \in \mathcal{S}_t \setminus \mathcal{S}^* \text{ we have } \theta_i(t) < 0.5 \} = \mathcal{A}^c(t).$$

Thereby we can write

$$\mathbb{I}[\mathcal{A}(t)] = \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}^c(t)] + \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)]$$

which leads to the following decomposition

$$\begin{aligned}
Reg(T) &= \sum_{t=1}^T \Delta_{\mathcal{S}_t} \mathbb{E} [\mathbb{I}(\mathcal{A}(t))] \\
&\leq \sum_{t=1}^T \mathbb{E} \left[\Delta_{\mathcal{S}_t} \times \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}(t)] + \Delta_{\mathcal{S}_t} \times \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}(t)] \right. \\
&\quad \left. + \Delta_{\mathcal{S}_t} \times \mathbb{I}[\mathcal{A}(t) \cap \mathcal{B}^c(t) \cap \mathcal{C}(t) \cap \mathcal{D}^c(t)] \right] \\
&= Reg_1(T) + Reg_2(T) + Reg_3(T).
\end{aligned}$$

Now, we bound $Reg_1(T)$, $Reg_2(T)$ and $Reg_3(T)$ with Lemma 5, Lemma 7, and Lemma 11.

This gives us Theorem 1.

B Additional Simulation Results

B.1 Offline Cases

The Figure 10 below shows simulation results for $p = 1000$ 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 4.

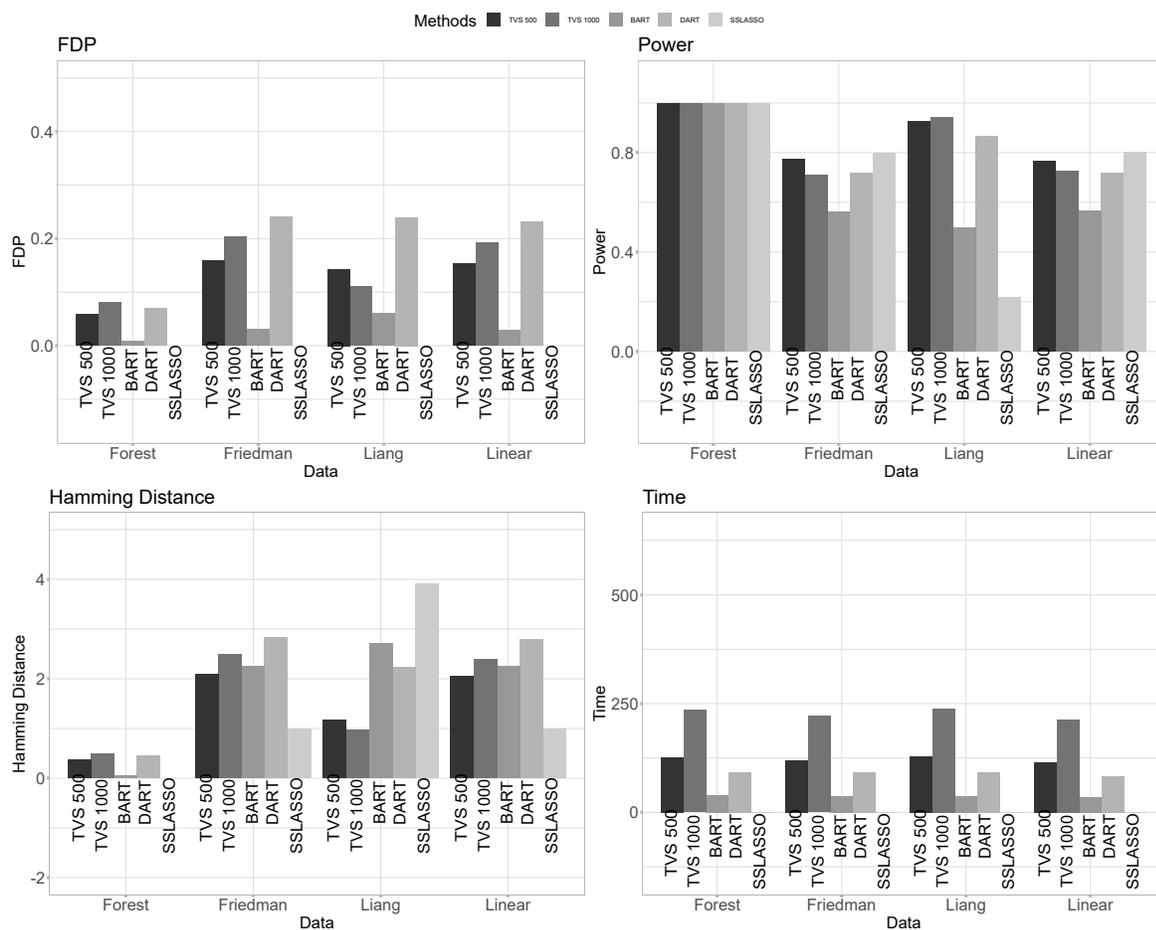


Figure 10: Graphs denoting FDP (6a), Power (6b), Hamming Distance (6c), and Time (6d) for the 4 choices of f_0 assuming $p = 1000$ and $n = 300$. The x -axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1000$.

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

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 4: The table records the number of iterations needed for TVS to converge in the simulation study in Section 6.1.

Probability 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 6.1. 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 5: 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 6: 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 7: 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 8: Forest 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	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
$p = 10000$										
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 Online Cases

In Table 9, 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 9: The table records the number of rounds needed for TVS to converge in the simulation study in Section 6.2.

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 Linero (2018)). Due to space constraints, we showed only the best settings for BART and DART in Section 6.2. 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 10: 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	67.74	20.00	0.01	0.03	0.79	0.07	1.08	0.27
	500	500	1	8.19	1.23	0.15	0.19	0.45	0.16	3.18	1.06
	500	500	5	36.33	5.31	0.01	0.06	0.73	0.10	1.40	0.57
	500	500	10	71.33	10.17	0.01	0.04	0.80	0.06	1.04	0.28
	1000	500	Till Converge	107.26	23.12	0.00	0.00	0.78	0.06	1.10	0.30
	1000	500	1	13.82	2.01	0.11	0.16	0.52	0.13	2.76	0.98
	1000	500	5	64.90	9.04	0.00	0.00	0.77	0.07	1.16	0.37
	1000	500	10	128.42	17.63	0.00	0.00	0.79	0.06	1.04	0.28
	500	1000	Till Converge	138.19	46.93	0.09	0.09	0.95	0.09	0.74	0.60
	500	1000	1	6.21	1.03	0.22	0.23	0.33	0.14	3.94	1.00
	500	1000	5	29.34	4.86	0.07	0.10	0.76	0.14	1.54	0.84
	500	1000	10	59.04	9.31	0.04	0.08	0.88	0.11	0.82	0.66
	1000	1000	Till Converge	225.04	81.37	0.02	0.05	0.96	0.08	0.32	0.47
	1000	1000	1	10.73	1.63	0.18	0.20	0.48	0.19	3.16	1.17
	1000	1000	5	52.29	7.94	0.04	0.08	0.82	0.12	1.12	0.63
	1000	1000	10	105.34	15.54	0.02	0.06	0.88	0.11	0.70	0.54
DART	D	Avg Split / Prob									
	10	Avg Split		654.69	108.53	0.28	0.15	0.72	0.13	2.92	1.24
	10	Prob		654.69	108.53	0.30	0.15	0.73	0.13	3.06	1.30
	50	Avg Split		1844.29	339.63	0.08	0.10	1.00	0.03	0.52	0.61
	50	Prob		1844.29	339.63	0.11	0.12	1.00	0.03	0.80	0.99
	200	Avg Split		6036.34	1215.50	0.03	0.07	1.00	0.00	0.16	0.42
BART	200	Prob		6036.34	1215.50	0.06	0.10	1.00	0.00	0.38	0.70
	10	Avg Split		1222.79	130.54	0.34	0.12	1.00	0.00	2.88	1.51
	10	Prob		1222.79	130.54	0.63	0.06	1.00	0.00	9.00	2.18
	50	Avg Split		2330.77	279.31	0.37	0.13	1.00	0.00	3.32	1.66
	50	Prob		2330.77	279.31	0.66	0.06	1.00	0.00	10.26	2.95
	200	Avg Split		8204.19	802.74	0.29	0.14	1.00	0.00	2.36	1.43
200	Prob		8204.19	802.74	0.73	0.06	1.00	0.00	14.38	3.76	
SSLASSO				9.98	1.36	0.00	0.02	1.00	0.00	0.02	0.14

Table 11: 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 12: 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
DART	D	Avg Split / Prob									
	10	Avg Split		1106.09	157.22	0.85	0.05	0.58	0.18	19.10	3.31
	10	Prob		1106.09	157.22	0.87	0.04	0.58	0.18	22.90	4.07
	50	Avg Split		2411.91	295.07	0.55	0.18	0.98	0.05	7.76	4.63
	50	Prob		2411.91	295.07	0.67	0.14	0.98	0.05	11.86	5.27
	200	Avg Split		5922.15	790.70	0.27	0.19	1.00	0.00	2.48	2.60
BART	200	Prob		5922.15	790.70	0.42	0.20	1.00	0.00	5.18	5.29
	10	Avg Split		1766.25	155.04	0.71	0.07	1.00	0.00	13.22	3.77
	10	Prob		1766.25	155.04	0.91	0.01	1.00	0.00	48.66	6.26
	50	Avg Split		2836.01	220.24	0.71	0.07	1.00	0.00	13.34	4.35
	50	Prob		2836.01	220.24	0.91	0.01	1.00	0.00	52.74	9.04
	200	Avg Split		7662.60	648.20	0.74	0.05	1.00	0.00	15.34	3.92
200	Prob		7662.60	648.20	0.90	0.01	1.00	0.00	46.92	6.89	
SSLASSO				17.31	1.12	0.00	0.00	0.60	0.00	2.00	0.00

Table 13: 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 14: 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	68.71	22.88	0.01	0.03	0.79	0.07	1.08	0.27
	500	500	1	8.24	0.96	0.15	0.19	0.45	0.16	3.18	1.06
	500	500	5	36.50	4.24	0.01	0.06	0.73	0.10	1.40	0.57
	500	500	10	71.61	8.29	0.01	0.04	0.80	0.06	1.04	0.28
	1000	500	Till Converge	107.02	19.16	0.00	0.00	0.78	0.06	1.10	0.30
	1000	500	1	13.86	1.68	0.11	0.16	0.52	0.13	2.76	0.98
	1000	500	5	65.18	7.62	0.00	0.00	0.77	0.07	1.16	0.37
	1000	500	10	128.95	14.79	0.00	0.00	0.79	0.06	1.04	0.28
	500	1000	Till Converge	138.53	41.50	0.09	0.09	0.95	0.09	0.74	0.60
	500	1000	1	6.30	0.86	0.22	0.23	0.33	0.14	3.94	1.00
	500	1000	5	29.90	4.30	0.07	0.10	0.76	0.14	1.54	0.84
	500	1000	10	60.46	8.95	0.04	0.08	0.88	0.11	0.82	0.66
	1000	1000	Till Converge	229.65	84.59	0.02	0.05	0.96	0.08	0.32	0.47
	1000	1000	1	10.98	1.59	0.18	0.20	0.48	0.19	3.16	1.17
	1000	1000	5	53.85	8.12	0.04	0.08	0.82	0.12	1.12	0.63
	1000	1000	10	109.13	16.27	0.02	0.06	0.88	0.11	0.70	0.54
DART	D	Avg Split / Prob									
	10	Avg Split		658.81	97.99	0.28	0.15	0.72	0.13	2.92	1.24
	10	Prob		658.81	97.99	0.30	0.15	0.73	0.13	3.06	1.30
	50	Avg Split		1812.87	314.26	0.08	0.10	1.00	0.03	0.52	0.61
	50	Prob		1812.87	314.26	0.11	0.12	1.00	0.03	0.80	0.99
	200	Avg Split		5625.74	1182.41	0.03	0.07	1.00	0.00	0.16	0.42
BART	200	Prob		5625.74	1182.41	0.06	0.10	1.00	0.00	0.38	0.70
	10	Avg Split		1239.32	144.56	0.34	0.12	1.00	0.00	2.88	1.51
	10	Prob		1239.32	144.56	0.63	0.06	1.00	0.00	9.00	2.18
	50	Avg Split		2309.34	254.22	0.37	0.13	1.00	0.00	3.32	1.66
	50	Prob		2309.34	254.22	0.66	0.06	1.00	0.00	10.26	2.95
	200	Avg Split		8223.55	763.94	0.29	0.14	1.00	0.00	2.36	1.43
200	Prob		8223.55	763.94	0.73	0.06	1.00	0.00	14.38	3.76	
SSLASSO				10.58	1.03	0.00	0.02	1.00	0.00	0.02	0.14

Table 15: 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.

$n = 50\,000$												
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	92.68	30.09	0.00	0.00	0.82	0.07	0.92	0.34	
	500	500	1	38.99	5.26	0.04	0.08	0.76	0.11	1.38	0.64	
	500	500	5	189.31	25.28	0.00	0.00	0.88	0.10	0.60	0.49	
	500	500	10	377.61	51.01	0.00	0.00	0.96	0.08	0.18	0.39	
	1000	500	Till Converge	144.33	27.93	0.00	0.00	0.80	0.03	0.98	0.14	
	1000	500	1	68.24	8.90	0.00	0.00	0.76	0.08	1.22	0.42	
	1000	500	5	340.58	43.83	0.00	0.00	0.88	0.10	0.60	0.49	
	1000	500	10	688.34	89.35	0.00	0.00	0.97	0.07	0.16	0.37	
	500	1000	Till Converge	157.31	57.97	0.09	0.08	0.96	0.08	0.70	0.46	
	500	1000	1	31.79	5.21	0.08	0.11	0.76	0.13	1.56	0.76	
	500	1000	5	159.72	24.19	0.09	0.08	0.98	0.05	0.60	0.49	
	500	1000	10	320.84	47.18	0.14	0.07	1.00	0.03	0.84	0.42	
	1000	1000	Till Converge	233.72	80.83	0.02	0.05	0.96	0.08	0.30	0.46	
	1000	1000	1	56.07	8.56	0.04	0.08	0.81	0.14	1.14	0.67	
	1000	1000	5	284.78	40.28	0.02	0.05	0.99	0.04	0.14	0.35	
	1000	1000	10	573.66	79.79	0.02	0.06	1.00	0.00	0.14	0.35	
$n = 100\,000$												
TVS	500	500	Till Converge	90.43	31.65	0.00	0.00	0.82	0.07	0.92	0.34	
	500	500	1	76.42	10.84	0.00	0.00	0.81	0.06	0.96	0.28	
	500	500	5	378.99	51.48	0.00	0.00	0.99	0.05	0.06	0.24	
	500	500	10	762.54	102.71	0.00	0.00	1.00	0.03	0.02	0.14	
	1000	500	Till Converge	160.73	61.01	0.00	0.00	0.80	0.06	1.02	0.32	
	1000	500	1	135.59	17.46	0.00	0.00	0.79	0.05	1.06	0.24	
	1000	500	5	676.30	87.33	0.00	0.00	0.94	0.09	0.28	0.45	
	1000	500	10	1362.59	173.47	0.00	0.00	0.99	0.05	0.06	0.24	
	500	1000	Till Converge	170.39	49.94	0.07	0.09	0.96	0.08	0.60	0.61	
	500	1000	1	62.11	9.69	0.06	0.09	0.84	0.12	1.10	0.74	
	500	1000	5	315.54	45.39	0.13	0.07	1.00	0.00	0.78	0.42	
	500	1000	10	636.30	89.61	0.15	0.05	1.00	0.00	0.92	0.27	
	1000	1000	Till Converge	301.02	115.25	0.01	0.04	0.98	0.05	0.14	0.35	
	1000	1000	1	111.02	15.94	0.01	0.05	0.89	0.11	0.62	0.60	
	1000	1000	5	564.63	76.20	0.03	0.06	1.00	0.00	0.18	0.39	
	1000	1000	10	1134.53	151.71	0.09	0.08	1.00	0.00	0.52	0.50	

Table 16: 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
DART	D	Avg Split / Prob									
	10	Avg Split		767.58	98.89	0.13	0.13	0.93	0.10	1.14	1.14
	10	Prob		767.58	98.89	0.14	0.14	0.93	0.10	1.22	1.17
	50	Avg Split		1841.16	299.40	0.16	0.16	1.00	0.00	1.18	1.34
	50	Prob		1841.16	299.40	0.32	0.16	1.00	0.00	2.78	1.80
	200	Avg Split		5706.31	1011.88	0.59	0.09	1.00	0.00	7.78	2.94
	200	Prob		5706.31	1011.88	0.68	0.07	1.00	0.00	11.18	3.32
BART	10	Avg Split		1430.25	201.17	0.34	0.16	1.00	0.00	3.06	1.89
	10	Prob		1430.25	201.17	0.68	0.05	1.00	0.00	11.10	2.44
	50	Avg Split		2617.58	347.14	0.48	0.13	1.00	0.00	5.20	2.19
	50	Prob		2617.58	347.14	0.82	0.03	1.00	0.00	23.30	3.70
	200	Avg Split		8229.39	618.28	0.83	0.03	1.00	0.00	25.68	4.84
	200	Prob		8229.39	618.28	0.91	0.01	1.00	0.00	53.90	6.39
SSLASSO				1.62	0.16	0.00	0.00	1.00	0.00	0.00	0.00

Table 17: 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	

C 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 https://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/. Barber et al. (2015) published cleaning codes (available at <https://web.stanford.edu/group/candes/knockoffs/software/knockoffs/tutorial-4-r.html>) which we adopt. We provide a basic overview of the dataset in Table 18.

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
EFV		1.08	312	732
NVP		1.80	313	744

Table 18: Basic summary statistics of the HIV dataset. DS refers to the decrease in susceptibility of the drug once the mutations has occurred.

In Section 7.1, we illustrated TVS on only the drug LPV. Here, we present the rest of the results. As is done in Barber et al. (2015), 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 19: 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 20: 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 21: 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

D 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