



Variable Selection Via Thompson Sampling

Yi Liu & Veronika Ročková

To cite this article: Yi Liu & Veronika Ročková (2021): Variable Selection Via Thompson Sampling, Journal of the American Statistical Association, DOI: [10.1080/01621459.2021.1928514](https://doi.org/10.1080/01621459.2021.1928514)

To link to this article: <https://doi.org/10.1080/01621459.2021.1928514>

 View supplementary material 

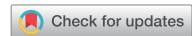
 Published online: 06 Jul 2021.

 Submit your article to this journal 

 Article views: 412

 View related articles 

 View Crossmark data 
CrossMark



Variable Selection Via Thompson Sampling

Yi Liu^a and Veronika Ročková^b

^aDepartment of Statistics, University of Chicago, Chicago, IL; ^bBooth School of Business, University of Chicago, Chicago, IL

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 nonparametric models and large datasets with very many predictors and/or very many observations. Depending on the choice of a reward, TVS can be deployed in offline as well as online setups with streaming data batches. Tailoring multiplay bandits to variable selection, we provide regret bounds without necessarily assuming that the arm mean rewards are 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.

ARTICLE HISTORY

Received June 2020
Accepted March 2021

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 of variation. Many today's most powerful prediction tools, however, lack an intuitive algebraic form which renders their interpretability (i.e., insight into the black-box decision process) far from straightforward. Substantial effort has been recently devoted to enhancing the explainability of machine learning through the identification of key variables that drive predictions (Garson 1991; Zhang, Ge, and Hang 2000; Olden and Jackson 2002; Lu et al. 2018; Horel and Giesecke 2019; Burns, Thomason, and Tansey 2020). While these procedures may possess nice theoretical guarantees, they may not yet be feasible for large-scale applications. This work develops a new computational platform for understanding black-box predictions which is based on reinforcement learning and which can be applied to very large datasets.

A variable can be important because its change has a causal impact or because leaving it out reduces overall prediction capacity (Mase, Owen, and Seiler 2019). 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) proposed 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, Rudin, and Dominici (2019) for a recent overview. In Bayesian forests, such as BART (Chipman, George, and McCulloch 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) introduced a variable importance based on a Shapley value and Hooker (2007) investigated 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 nonparametric 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

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

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 , that is, $\pi(\mathcal{S}|q) = \frac{1}{\binom{p}{q}}$. 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; Ročková and George 2018; Rossell and Telesca 2017; Vannucci and Stingo 2010; Brown, Vannucci, and Fearn 1998), computational aspects (George and McCulloch 1993, 1997; Bottolo et al. 2010; Carbonetto et al. 2012; Ročková and George 2014) and/or variable selection consistency results (Johnson and Rossell 2012; Narisetty et al. 2014; Castillo, Schmidt-Hieber, and Van der Vaart 2015). Traditionally, spike-and-slab methodology relies on the underlying model to be linear, which may be woefully inaccurate, and can be computationally slow. In this work, we leave behind the linear model framework and focus on interpretable machine learning linking spike-and-slab methods with binary bandits. The two major methodological benefits are (a) ability to capture nonlinear effects and (b) scalability to very large datasets. Existing nonlinear variable selection approaches include grouped shrinkage/selection of basis-expansion coefficients (Lin and Zhang 2006; Ravikumar et al. 2009; Radchenko and James 2010; Scheipl 2011), regularization of the derivative expectation operator (Lafferty and Wasserman 2008) or model-free knockoffs (Candes et al. 2018). The main distinguishing feature of our approach is the development of a new computational platform via a spike-and-slab wrapper that extends the reach of machine learning to large-scale data.

This article 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 toward 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 (MPM) 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 article 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 TVS 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 TVS, 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*, that is, 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).

Thompson Sampling is a Bayesian-inspired heuristic algorithm that achieves a logarithmic expected regret (Agrawal and Goyal 2012) in the Bernoulli bandit problem. Starting with a noninformative prior $\theta_i \stackrel{\text{iid}}{\sim} \text{Beta}(1, 1)$ 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 $[0, 1]$.

The MAB problem is most often formulated as a single-play problem, where only one arm can be selected at each round.

Komiyama, Honda, and Nakagawa (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, that is, $\mathbb{E}[\mathcal{R}(T)] = \mathbb{E} \sum_{t=1}^T (\sum_{i \in \mathcal{S}_0} \theta_i - \sum_{i \in \mathcal{S}_t} \theta_i)$. Fixing L , the number of arms played, Komiyama, Honda, and Nakagawa (2015) proposed a Thompson sampling algorithm for this problem and show that it has a logarithmic expected regret with respect to time and a linear regret with respect to the number of arms. Our metamorphosis of multi-armed bandits into a variable selection algorithm will ultimately require that the number L of arms played is random and that the rewards at each time t can be dependent.

Finally, we complete the review of MAB techniques with *combinatorial bandits* (Chen, Wang, and Yuan 2013; Gai, Krishnamachari, and Jain 2012; Cesa-Bianchi and Lugosi 2012) which are the closest relative to our proposed method here. Combinatorial bandits can be seen as a generalization of multi-play bandits, where any arbitrary combination of arms \mathcal{S} (called super-arms) is played at each round and where the reward $r(\mathcal{S})$ can be revealed for the entire collective \mathcal{S} (a full-bandit feedback) or for each contributing arm $i \in \mathcal{S}$ (a semi-bandit feedback), see, for example, Wang and Chen (2018), Combes and Proutiere (2014), Kveton et al. (2015), Combes and Proutiere (2014), and Kveton et al. (2015). We will draw upon connections between combinatorial bandits and variable selection multiple times throughout Section 3 and 4.

3. Variable Selection as a Bandit Problem

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

Bayesian model selection with spike-and-slab priors has often been synonymous to finding the MAP model $\widehat{\mathcal{S}} = \arg \max_{\mathcal{S}} \pi(\mathcal{S} | \mathbf{Y}^{(n)})$. Even when the marginal likelihood is available, this model can computationally unattainable for p as small as 20. In order to accelerate Bayesian variable selection using multi-armed bandits techniques one idea immediately comes to mind. One could treat each of the 2^p models as a base arm. Assigning prior model probabilities according to $\theta_i \sim \text{Beta}(a_i, b_i)$ 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 been standard practice to report the *MPM* (Barbieri and Berger 2004) consisting of those variables whose posterior inclusion probability $\pi_i \equiv \mathbb{P}(\gamma_i = 1 | \mathbf{Y}^{(n)})$ is at least 0.5. More formally, MPM is defined, for $\boldsymbol{\pi} = (\pi_1, \dots, \pi_p)'$, as

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

This is now the default model selection rule with spike-and-slab priors (2). The MPM model is the optimal predictive model in linear regression under some assumptions (Barbieri et al. 2020). 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, Chakraborty, and Mallick 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.

Having reviewed the two traditional Bayesian model choice reporting methods (MAP and MPM), we can now forge connections to multi-armed bandits. While the MAP model suggests treating *each model S* as a bandit arm, the MPM model suggests treating *each variable γ_i* as a bandit arm. Under the MAP framework, the player would be required to play a single arm (i.e., a model) at each step. The MPM framework, on the other hand, requires playing a random subset of arms (i.e., a model) at each play opportunity. This is appealing for at least two reasons: (i) there are fewer arms to explore more efficiently, (ii) the quantity $r_{\boldsymbol{\pi}}(\mathcal{S})$ in Equation (3) can be regarded as a mean regret of a combinatorial arm (more below) which, given $\boldsymbol{\pi}$, has MPM as its computational oracle. The computational oracle is defined as the regret minimizer when an oracle furnishes yield probabilities θ_i (see forthcoming Lemma 1). Based on the discussion above, we regard the MPM framework as more intuitively appealing for bandit techniques. We thereby reframe spike-and-slab selection with priors (2) as a bandit problem treating each variable as an arm. This idea is formalized below.

We view Bayesian spike-and-slab selection through the lens of *combinatorial bandit problems* (reviewed earlier in Section 2) by treating variable selection indicators γ_i ’s in Equation (2) as Bernoulli rewards. From now on, we will refer to each θ_i as an unknown mean reward, that is, a probability that the i th variable exerts influence on the outcome. In sharp contrast to Equation (2) which deploys one θ for all arms, each arm $i \in \{1, \dots, p\}$ now has *its own* prior inclusion probability θ_i , that is,

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

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 (Ročková 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

¹chosen to correspond to marginals of a Dirichlet distribution

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.

We try a different route. Rather than distributing, we will aggregate. Namely, instead of collecting a global reward first and then redistributing it, we first collect individual rewards $\gamma_i^t \in \{0, 1\}$ for each played arm $i \in \mathcal{S}_t$ and then weave them into a global reward $R(\mathcal{S}_t)$. We assume that γ_i^t 's are iid from Equation (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, Krishnamachari, and Jain 2012), we consider a global reward for variable selection motivated by the MPM.

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 “MPM” as its computational oracle, as can be seen from Equation (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, that is, $\arg \max_{\mathcal{S}} R_C(\mathcal{S}) = \{i : \gamma_i^t = 1\}$. 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 $\theta = (\theta_1, \dots, \theta_p)'$, that is,

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

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

Lemma 1. Denote with $\mathcal{S}_O = \arg \max_{\mathcal{S}} r_\theta^C(\mathcal{S})$ 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 MPM $\{i : \theta_i \geq 0.5\}$.

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

Note that the choice of $C = (\sqrt{5} - 1)/2$ incurs *the same* penalty/opportunity cost for false positives and negatives since $\log(1+C) = -\log(C)$. In streaming feature selection, for example, Zhou et al. (2006) accommodated measurement cost and place cheaper variables earlier in the stream. In our framework, we can allow for different cost C_i (e.g., a measurement cost) for each variable $1 \leq i \leq p$. The existence of the computational oracle for the expected reward $r_\theta^C(\mathcal{S})$ is very comforting and will be exploited in our Thompson sampling algorithm introduced in Section 4

3.2. The Local Rewards

The global reward (5) is a deterministic functional of the local rewards. We have opted for the reward functional (5) because the regret minimizer is the MPM when the yield probabilities are provided (see Lemma 1). We now clarify the definition of local rewards γ_i^t . We regard \mathcal{S}_t as a smaller pool of candidate variables, which can contain false positives and false negatives. The goal is to play a game by sequentially trying out different subsets and reward true signals so that they are selected in the next round and to discourage false positives from being included again in the future. Denote with \mathbb{S} the set of all subsets of $\{1, \dots, p\}$ and with \mathcal{D} the “data” at hand consisting of $|\mathcal{D}|$ observations (Y_i, \mathbf{x}_i) from Equation (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 $r(\mathcal{S}_t, \mathcal{D})$: (i) a *stochastic feedback rule* $r(\cdot)$ assuming that data \mathcal{D} is given, and (ii) 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, George, and McCulloch 2001). We refer to Hill, Linero, and Murray (2020) and references therein for a nice recent overview of BART. In particular, we use the following binary local reward $r(\mathcal{S}_t, \mathcal{D}_n) = (\gamma_i^t : i \in \mathcal{S}_t)'$ where

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

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 $|\mathcal{S}_t| << p$) 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 article, 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 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.

3.3. Other Feedback Rules

TVS is not confined to a BART reward function. Deploying any variable selection method using only a subset \mathcal{S}_t will yield a binary feedback rule. For example, the LASSO method yields $\hat{\beta}_i$ for $i \in \mathcal{S}_t$ which can be turned into the following feedback $\gamma_i^t = \mathbb{I}(\hat{\beta}_i \neq 0)$ for $i \in \mathcal{S}_t$. In offline setups, randomness of γ_i^t 's can be induced by taking a bootstrap replicate of \mathcal{D}_n at each play time $1 \leq t \leq T$. Another possibility for a binary reward is dichotomizing p -values, similarly as in alpha-investing for streaming variable selection by Zhou et al. (2006). Instead of binary local rewards (8), one can also consider continuous rewards $\gamma_i^t \in [0, 1]^{|\mathcal{S}_t|}$ by rescaling variable importance measures obtained by a machine learning method (e.g., random forests (Louppe et al. 2013), deep learning (Horel and Giesecke 2019), and BART (Chipman, George, and McCulloch 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).

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.

4. Introducing TVS

This section introduces TVS, a reinforcement learning algorithm for subset selection in nonparametric 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, Wang, and Yuan 2013; Gai, Krishnamachari, and Jain 2012) 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 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}_0 in Equation (7). Unlike multi-play bandits where the number of played arms is predetermined (Komiyama 2015), this strategy allows one to adapt to the size of the model. We do, however, consider a variant of the computational oracle (see Equation (13)) 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 Equation (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 TVS is to learn the distribution of mean rewards θ_i 's by playing a game, that is, 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 MPM (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 $\widehat{\mathcal{S}}_t$, which satisfies $\widehat{\mathcal{S}}_t = \arg \max_{\mathcal{S}} r_{\boldsymbol{\pi}(t)}^C$. From Lemma 1, this model can be simply computed by truncating individual $\pi_i(t)$'s. Upon convergence, that is, when trajectories $\pi_i(t)$ stabilize over time, TVS

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

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

Table 1. Computing times (in sec) and Hamming distance of BART on subsets of observations using all p covariates.

T	$s = 100$		$s = 200$		$s = 500$		$s = 1000$		$s = 5000$		$s = 10,000$		$s = 20,000$	
	Time	HAM	Time	HAM	Time	HAM	Time	HAM	Time	HAM	Time	HAM	Time	HAM
5000	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

NOTE: Hamming distance compares the true model with a model obtained by truncating the BART importance measure at 1.

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, Honda, and Nakagawa ([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

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

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

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

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. Below, we first review two available regret bounds following from Wang and Chen ([2018](#)) for when (a) q^* is known and arms are independent ([Lemma 2](#)) and (b) arms are related and q^* is unknown ([Lemma 3](#)). Later in our [Theorem 1](#), we relax these assumptions and provide a regret bound assuming that q^* unknown and, at the same time, arms are related.

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} , that is, replacing the computational oracle in (7) with $\mathcal{S}_O^{q^*} = \arg \max_{\mathcal{S} \in \mathcal{I}} r_{\theta}(\mathcal{S})$. 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, that is,

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

We have the following regret bound which, unlike the majority of existing results for TS, *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*, one can derive 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 $\Delta_{\mathcal{S}} = [r_{\theta}(\mathcal{S}^*) - r_{\theta}(\mathcal{S})]$ and for each arm $i \in \{1, \dots, p\}$ define $\eta_i \equiv \max_{\mathcal{S}: i \in \mathcal{S}} \frac{8B^2|\mathcal{S}|}{\Delta_{\mathcal{S}} - 2B(q^*+2)\varepsilon}$ 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

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

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

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

The bandit literature has largely focused on studying the regret in terms of time T rather than the number of arms p . Note that the dependence on p in Lemma 3 is cubic, which (albeit relevant for large n setups) makes the bound less useful when p is very large. Lai and Robbins (1985) showed a lower regret bound (in terms of p) that is $\mathcal{O}(p)$ for any bandit algorithm. Agrawal and Goyal (2012) (Remark 3) further showed that Thompson Sampling does achieve this lower bound in a single-play bandit problem. Our Theorem 1 below shows a linear dependence on p for combinatorial bandits with correlated arms when q^* is unknown.

We now extend Lemma 3 to the case when the rewards obtained from pulling different arms are related to one another. Gupta, Joshi, and Yagan (2020) introduced a correlated single-play bandit version of TS using pseudo-rewards (upper bounds on the conditional mean reward of each arm). Similarly as with structured bandits (Pandey, Chakrabarti, and Agarwal 2007) we instead interweave the arms by allowing their mean rewards to depend on S_t , that is, instead of a single success probability θ_i we now have

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

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

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

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

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

Under this assumption we provide the following regret bound.

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

Proof. Appendix (Section A, supplementary material)

Algorithm 1:	Thompson Variable Selection with BART
INPUT	
	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 each arm $1 \leq i \leq p$.	
LOOP	
For $t = 1, \dots, T$ repeat steps C(1)-(3), R and U.	
Choose Step	
C(1): Set $S_t = \emptyset$ and for $i = 1 \dots p$ do	
C(2): Sample $\theta_i(t) \sim \text{Beta}(a_i(t), b_i(t))$	
C(3): (Unknown q^*) Compute $S_t = \{i : \theta_i(t) \geq \tilde{C}\}$ from (7)	
C(3)*: (Known q^*) Compute S_t from (13)	
Reward Step	
R: Collect local rewards γ_i^t 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 $b_i(t+1) = b_i(t) + 1$	
OUTPUT	
Evidence probabilities $\pi_i(t) = a_i(t) / [a_i(t) + b_i(t)]$ for $1 \leq i \leq p$ and $1 \leq t \leq T$.	

Algorithm 1: TVS with BART (* is an alternative with known q^*)

Two of the most common problems studied in reinforcement learning with bandits are (i) regret minimization, and (b) best arm identification (Even-Dar, Mannor, and Mansour 2006; Bubeck, Munos, and Stoltz 2009). Variable selection could be loosely regarded as the “top q^* -arms” identification problem when q^* is unknown. While TS is devised to minimize regret, not necessarily to select the best arms (see, e.g., Russo 2016), we nevertheless show that our sampling policy satisfies a version of variable selection consistency in the sense that the event $\{S_t = S^*\}$ occurs all but finitely many times as $t \rightarrow \infty$. This result is summarized in the following theorem.

Theorem 2. Under Assumption 1 with p fixed, the TVS sampling policy in Table 1 with $C = (\sqrt{5} - 1)/2$ satisfies $\mathbb{P}(\liminf_{t \rightarrow \infty} \{S_t = S^*\}) = 1$.

Proof. Appendix (Section B, supplementary material)

Remark 1. The proof proceeds by showing that the probability of infinitely many events $\mathcal{A}(t) = \{S_t \neq S^*\}$ occurring is zero, that is, $\mathbb{P}(\limsup_{t \rightarrow \infty} \mathcal{A}(t)) = 0$. Deploying the Borel–Cantelli lemma, it suffices to show that $\sum_{t=1}^{\infty} \mathbb{P}(\mathcal{A}(t)) < \infty$. Splitting the arms into active and nonactive ones, the proof then follows a strategy similar to the proof of Theorem 1.

5. TVS in Action

This section serves to illustrate TVS on benchmark simulated datasets and to document its performance. While various implementations are possible (by choosing different rewards $r(S_t, D)$ 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).

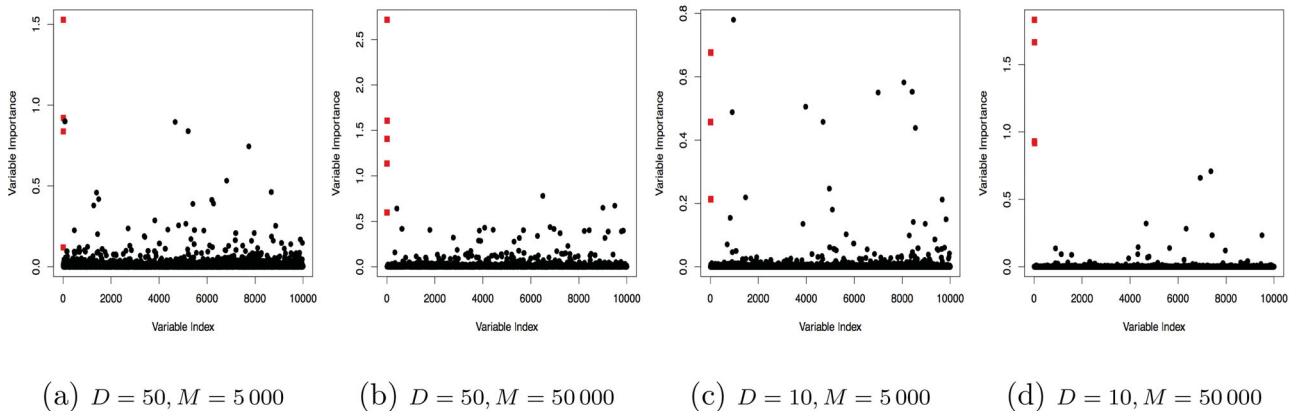


Figure 1. BART variable importance using `sparse=TRUE` and various number of trees D and MCMC iterations M . Red squares (the first five covariates) are signals and black dots are noise variables.

5.1. Offline TVS

As a lead example in this section, we consider the benchmark (Friedman 1991) dataset with a vastly larger number of $p = 10,000$ predictors $\mathbf{x}_i \in [0, 1]^p$ obtained by iid sampling from $\text{Uniform}(0, 1)$ 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 $M \in \{5000, 50,000\}$ 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 four true signals, leaving out the quadratic signal variable x_3 . The computation took around 8.5 min.

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* 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 Equation (9). We start with a noninformative 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 $M \in \{100, 500, 1000\}$ 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 Equation (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 1(a)) 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 five 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 toward the end of the plot. Using more MCMC iterations M , fewer TVS iterations are needed to obtain a cleaner separation between signal and noise (noise π_i ’s are closer to zero while signal π_i ’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 min for 20,000 iterations with 40 sec 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, that is, $n \gg p$. As we already pointed out in Section 3.2, we assume that the dataset

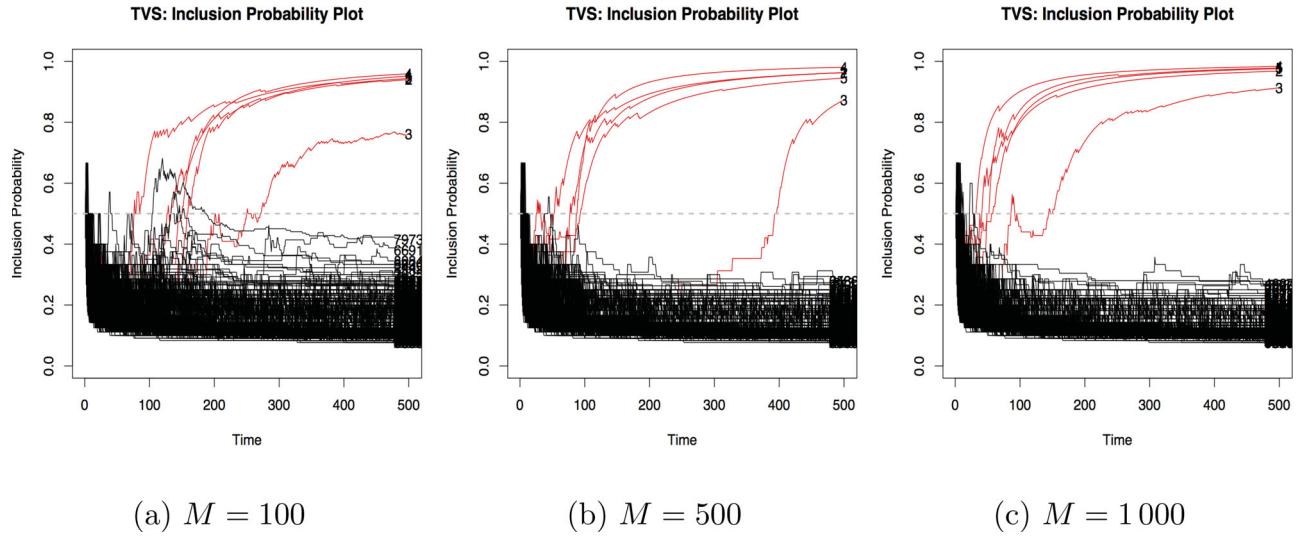


Figure 2. The evolution of inclusion probabilities (11) over “time” t for the Friedman dataset. The plot depicts posterior inclusion probabilities $\pi_i(t)$ in (11) over time (number of TVS iterations). Red lines indicate the 5 signal variables and black lines indicate the noise variables.

$\mathcal{D}_n = \{(Y_i, \mathbf{x}_i) : 1 \leq i \leq n\}$ 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, Li, and Zhou 2018, sec. 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 $e_i, z_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1)$. 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 Equation (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

nonlinear contributions of $x_1 - x_4$. Unlike Liang et al. (2018) who set $n = p = 500$, we make the problem considerably more difficult by choosing $n = 20,000$ 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 2018) by setting `sparse=TRUE` (Linero and Yang 2018) and 50 trees⁶ in the BART software (Chipman, George, and McCulloch 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 $s \in \{100, 500, 1000\}$ 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 $s \geq 1000$ 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 3(d) is very instructive for selection, it took more than 70 min 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

⁵Treating the rewards as data.

⁶Results with 10 were not nearly as satisfactory.

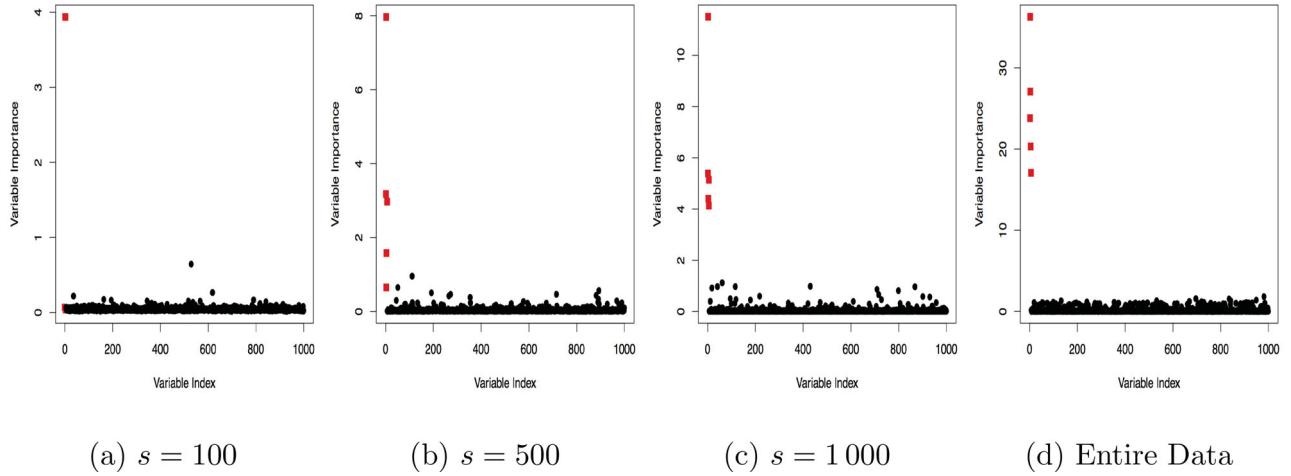


Figure 3. BART variable importance using $T=20,000$ (using 50 trees and `sparse=TRUE`) and various data subsets. Red squares (the first five covariates) are signals and black dots are noise variables.

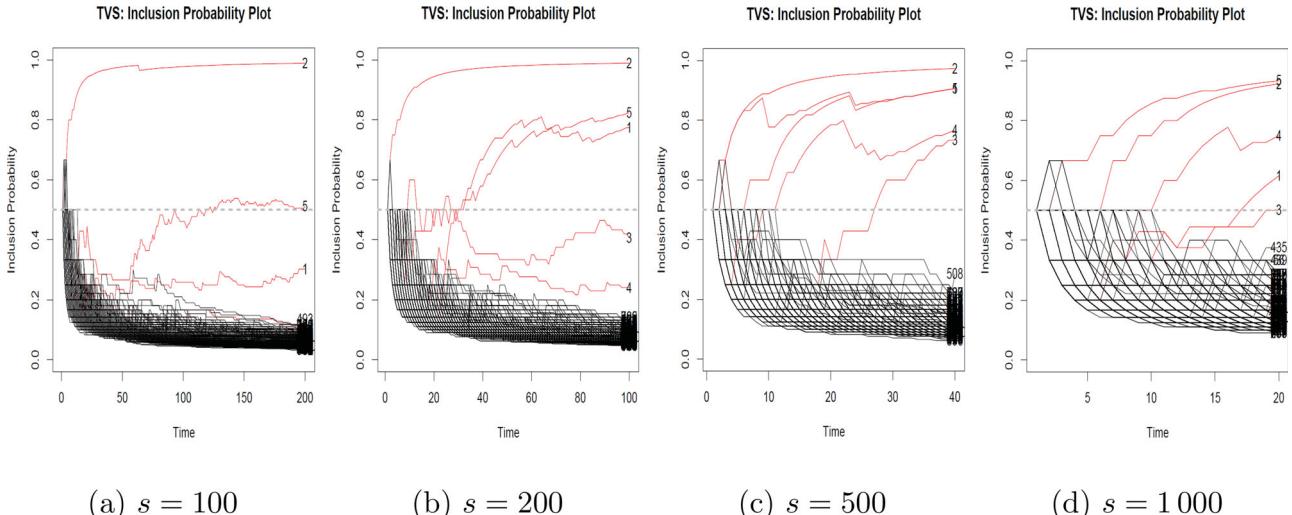


Figure 4. TVS inclusion probabilities $T=10,000$ (using 10 trees and `sparse=TRUE`) and various batch sizes after single pass through the data.

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 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 Figures 3(b) and 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 $s = 1000$ 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. 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.

6. Simulation Study

We further evaluate the performance of TVS in a more comprehensive simulation study. We compare TVS with several related nonparametric variable selection methods and with classical parametric ones. We assess these methods based on the fol-

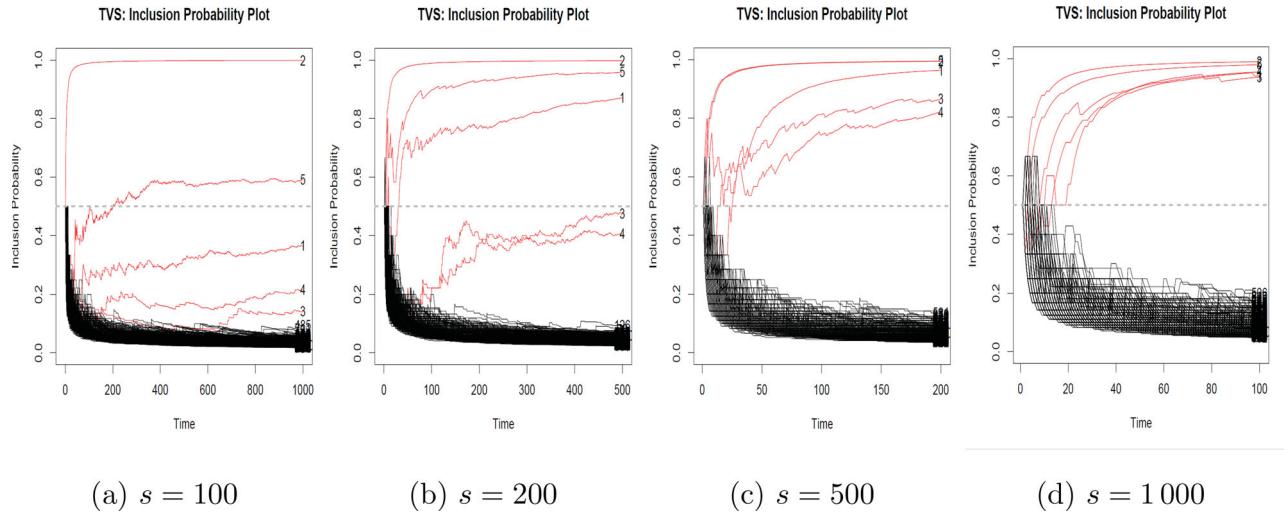


Figure 5. TVS inclusion probabilities $T=10,000$ (using 10 trees and `sparse=TRUE`) and various batch sizes after 5 passes through the data.

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.

	$s = 100$		$s = 200$		$s = 500$		$s = 1000$	
	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
$T = 10,000$								
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

NOTE: The Hamming distance compares the true model with a model truncating the last TVS inclusion probability to 0.5.

lowing 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 $p \in \{1000, 10,000\}$ 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 4.1. 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. (2018) 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 $M = 50,000$ MCMC iterations using the default prior settings with $D = 20, D = 50$ and $D = 200$ trees for BART and $D = 10, D = 50$, and $D = 200$ trees for DART. We considered two variable selection criteria: (i) 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), (ii) 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, that is, BART with $D = 20$

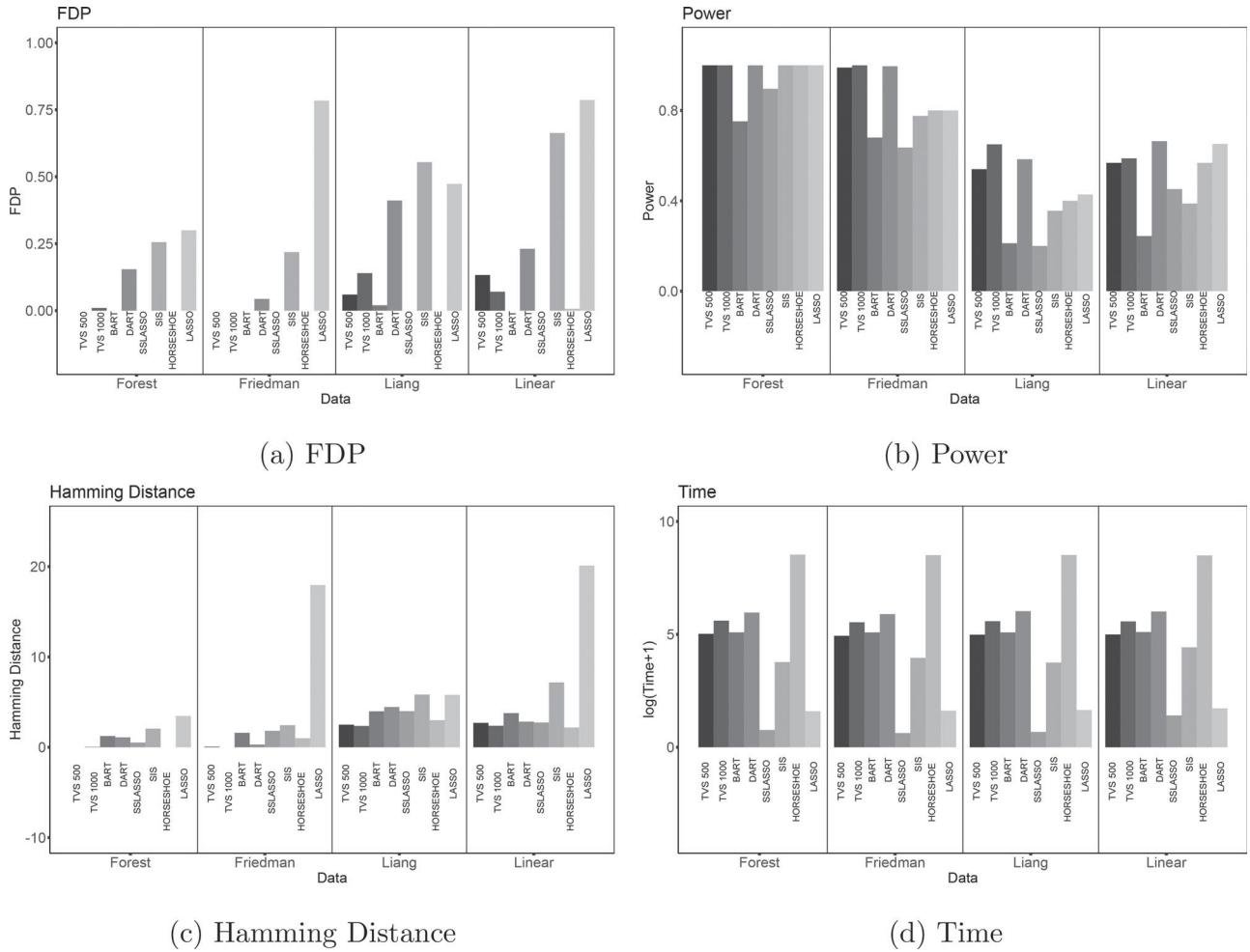


Figure 6. Graphs denoting FDP (a), Power (b), Hamming Distance (c), and Time (d) for the 4 choices of f_0 assuming $p=10,000$ and $n = 300$. The x-axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M = 1000$.

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 (Ročková 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 have also implemented sure independence screen (SIS) (Fan and Lv 2008) as a variable filter before applying BART variable selection.⁷ Next, we applied one of the benchmark Bayesian variable selection methods, the horseshoe prior (Carvalho, Polson, and Scott 2010; van der Pas et al. 2017) implementation from the R-package `horseshoe`. We run the Markov chain for 55,000 iterations, discarding the first 5000 as a burnin period. Variable selection is performed by checking whether 0 is contained in the credible set (see van der Pas et al. 2019). Finally, we implement the LASSO (Tibshirani 2011) and report the model chosen according to the 1-s.e. rule (Friedman, Hastie, and Tibshirani 2001).

We report the average performance (over 50 datasets) for $p=10,000$ in Figure 6 and the rest (for $p=1000$) in the appendix

(supplementary material). 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 the best across-the-board. DART (with $D = 50$) performs consistently well in terms of variable selection, but the timing comparisons are less encouraging. BART (with $D = 20$) takes a relatively comparable amount of time as TVS with $M = 1000$, but suffers from less power. SS-LASSO's performance is strong, in particular for the less nonlinear data setups. The performance of TVS is seen to improve with M . In its simplest form, SIS is a screening method based on a linear model assumption. Albeit faster than TVS, we observe that SIS generally over-selects. In addition, SIS screens out key signals with nonlinear effects (Liang and Friedman datasets). Note, however, that our SIS implementation relies on the linear model where performance could be further improved using nonlinear variants such as distance correlation learning (Li, Zhong, and Zhu 2012). The Horseshoe prior performs very well but is much slower. LASSO seems to overfit, in spite of the 1-se rule.

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{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,

⁷SIS on its own yielded too many false positives.

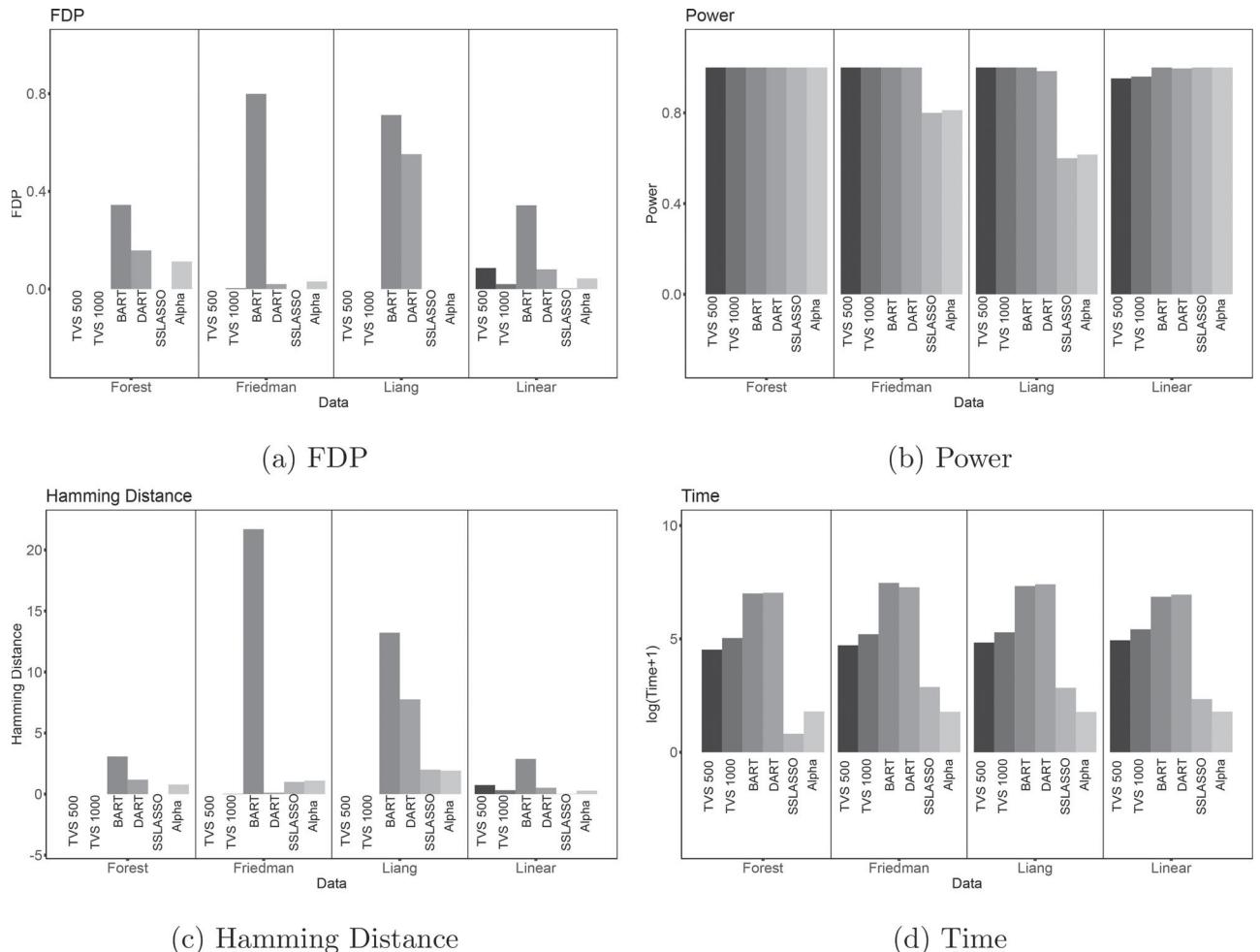


Figure 7. Graphs denoting FDP (a), Power (b), Hamming Distance (c), and Time (c) for the 4 choices of f_0 assuming $p=1000$ and $n = 10,000$. The x-axis denotes the choice of f_0 and the various methods are marked with various shades of gray. For TVS, we have two choices $M = 500$ and $M=1000$, both with $s = 1000$.

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 C.1, supplementary material)

6.2. Online Cases

We now consider a simulation scenario where $n \gg p$, that is, $p = 1000$ 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 $\mathbf{Y}^{(b)} = (\mathbf{y}_i : (b-1)s+1 \leq i \leq bs)$ 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 $\hat{\mathcal{S}}_t$ 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 C.2, supplementary material). 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 nonnegligible 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. In the appendix (Section C.2, Tables 8–12, supplementary material), we report TVS results for n as large as $n = 100,000$ (which is beyond the scope of BART), showing very satisfactory and scalable performance. For TVS, we found the batch size $s = 1000$ 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. Streaming feature selection methods are still being developed (Ramdas et al. 2017; Javanmard et al. 2018; Wang, Shen, and Li 2018; Fahy and Yang 2019). We have compared our online variant with the α -investing method for streaming variable selection of Zhou et al. (2006). This method dynamically adjusts p -value thresholds for adding features to the model. We run this method using p -values from applying linear regression on batches of streaming data. The results are shown in Figure 7. This procedure performs well in Forest and Linear set-ups but misses the

key nonlinear variables in Friedman and Liang setups. While SSLASSO's performance is very strong, we notice that in the nonlinear setup of Liang, Li, and Zhou (2018), there are false nondiscoveries.

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 available⁸ dataset consists of genotype and resistance measurements (decrease in susceptibility on a log scale) to three drug classes: (i) protease inhibitors (PIs), (ii) nucleoside reverse transcriptase inhibitors (NRTIs), and (iii) nonnucleoside 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 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.⁹

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 8, 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 (Ročková 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 dis-

covered Positions for each method are plotted in Figure 8(b). While LASSO selects many more experimentally validated mutations, it also includes many unvalidated ones. TVS, on the other hand, has a very small number of “false discoveries” while maintaining good power. Additional results are included in the appendix (Section D, supplementary material).

7.2. Durable Goods Marketing Dataset

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 prefiltering leaves us with 114 variables whose names and descriptive statistics are reported in Section E in the appendix (supplementary material). 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 $s = 1000$, 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.

Finally, we used the same set of variables (including knockoff variables) for different variable selection methods and recorded

⁸Stanford HIV Drug Resistance Database https://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/

⁹As 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.

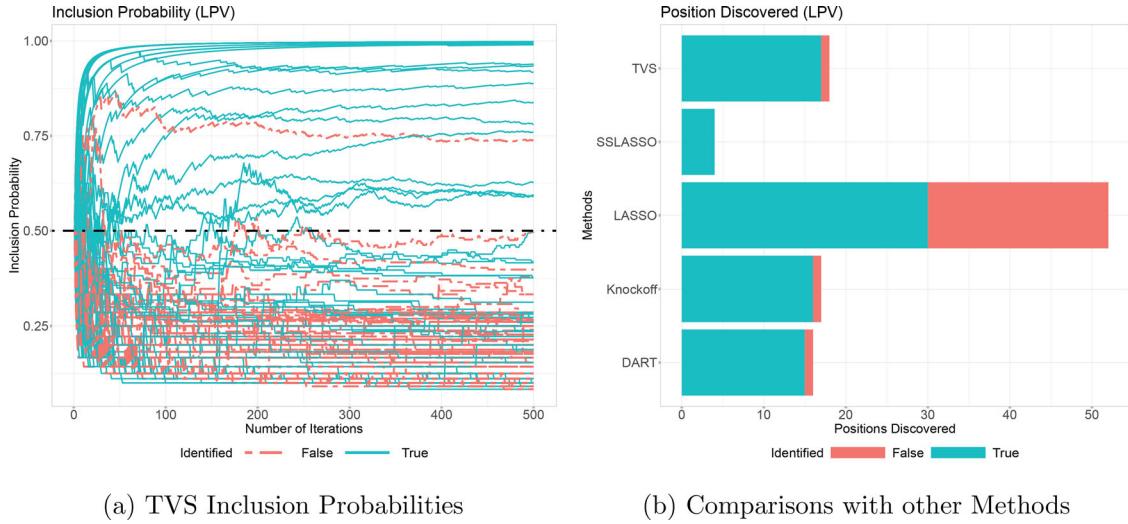


Figure 8. TVS on the LPV dataset. (Left) Trajectories of the inclusion probabilities $\pi_j(t)$. (Right) Number of signals discovered, where blue denotes the experimentally validated signals and red are the unvalidated ones.

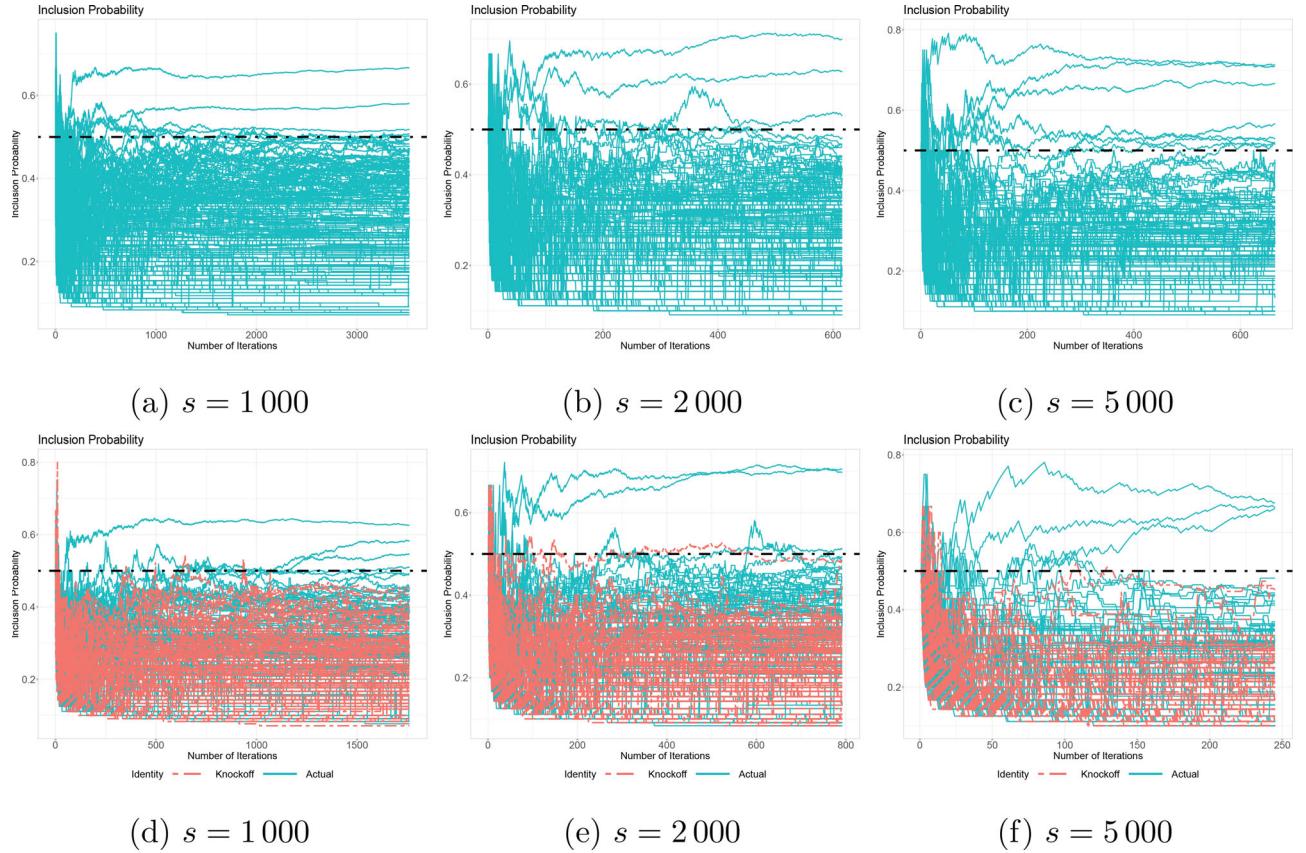


Figure 9. TVS on the marketing data. (Left) Trajectories of the inclusion probabilities $\pi_j(t)$ without knockoffs. (Right) Trajectories of the inclusion probabilities $\pi_j(t)$ with knockoffs (in red).

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, that is, 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

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.

Knockoff	$s = 1\,000$		$s = 2\,000$		$s = 5\,000$	
	Yes	No	Yes	No	Yes	No
total number of medium ticket items in previous 60 months	0.30	0.51	0.51	0.46	0.44	0.51
total number of small ticket items in previous 60 months	0.49	0.52	0.40	0.41	0.25	0.53
number of months shopped once in previous 12 months	0.34	0.41	0.44	0.42	0.41	0.57
number of months shopped once in previous 24 months	0.63	0.67	0.70	0.63	0.66	0.71
count of unique purchase trips in previous 24 months	0.55	0.26	0.48	0.53	0.68	0.67
total number of items purchased in previous 12 months	0.51	0.30	0.24	0.21	0.24	0.11
promo_nov period: total sales	0.58	0.58	0.71	0.70	0.33	0.45
mailed in holiday 2001 mailer	0.25	0.43	0.08	0.41	0.42	0.52
percent audio category sales of total sales \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 4. The number of selected knockoffs in the marketing data. BART and DART are run on a subset of 2 000 observations.

Method	TVS	TVS	TVS	BART	DART	LASSO
s	1000	2000	5000	0	0	45

has led to a proposal of TVS, a reinforcement learning wrapper algorithm for fast variable selection in high dimensional non-parametric problems. In related work, Liu, Ročková, and Wang (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 (noninformative) 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 an MPM 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, on both real and simulated data.

The TVS approach, coupled with BART, captures nonlinearities and may thus be beneficial over linear techniques. In addition, TVS scales to very large datasets.

TS is ultimately designed to minimize regret, not to select the best arms. Bubeck, Munos, and Stoltz (2009) pointed out that algorithms satisfying regret bounds of order $\log(T)$ can be far from optimal for finding the best arm. Russo (2016) proposed a “top-two” sampling version of TS which is tailored for single best-arm identification. Alternative algorithms have been proposed including the successive elimination algorithm of Even-Dar, Mannor, and Mansour (2006) for single top-arm identification and the SAR (successive accepts rejects) algorithm of Bubeck, Wang, and Viswanathan (2013) for top m -arm

identification. These works suggest possible refinements of our approach for directly targeting multiple best arms in correlated combinatorial bandits.

Supplementary Material

The Supplemental Materials contain proofs of Theorem 1 and 2 (Section A and B, respectively), additional simulation results (Section C) and additional details on the real data analyses (Section D and E).

Funding

The authors gratefully acknowledge the support from the James S. Kemper Foundation Research Fund at the Booth School of Business and the National Science Foundation (DMS:1944740).

References

- Agrawal, S. and Goyal, N. (2012), “Analysis of Thompson Sampling for the Multi-Armed Bandit Problem,” in *COLT 2012: 25th Annual Conference on Learning Theory* in Edinburgh. [2,5,6,7]
- Barber, R. F., Candès, E. J. (2015), “Controlling the False Discovery Rate Via Knockoffs,” *The Annals of Statistics*, 43, 2055–2085. [14]
- Barbieri, M., Berger, J. O., George, E. I., and Ročková, V. (2020), “The Median Probability Model and Correlated Variables,” *Bayesian Analysis* (to appear). [3]
- Barbieri, M. M., and Berger, J. O. (2004), “Optimal Predictive Model Selection,” *Annals of Statistics*, 32, 870–897. [2,3,5]
- Bhattacharya, A., Chakraborty, A., and Mallick, B. K. (2016), “Fast Sampling With Gaussian Scale Mixture Priors in High-Dimensional Regression,” *Biometrika*, 103, 985. [3]
- Bleich, J., Kapelner, A., George, E., and Jensen, S. (2014), “Variable Selection for BART: An Application to Gene Regulation,” *Annals of Applied Statistics*, 8, 1750–1781. [9,11]
- Bottolo, L., and Richardson, S. (2010), “Evolutionary Stochastic Search for Bayesian Model Exploration,” *Bayesian Analysis*, 5, 583–618. [2]
- Breiman, L. (2001), “Random Forests,” *Machine Learning*, 45, 5–32. [1]
- Brown, P. J., Vannucci, M., and Fearn, T. (1998), “Multivariate Bayesian Variable Selection and Prediction,” *Journal of the Royal Statistical Society, Series B*, 60, 627–641. [2]
- Bubeck, S., Munos, R., and Stoltz, G. (2009), “Pure Exploration in Multi-Armed Bandits Problems,” in *International Conference on Algorithmic Learning Theory*, Porto, Portugal. [7,16]
- Bubeck, S., Wang, T., and Viswanathan, N. (2013), “Multiple Identifications in Multi-Armed Bandits,” in *International Conference on Machine Learning*, Atlanta, Georgia. [16]
- Burns, C., Thomason, J., and Tansey, W. (2020), “Interpreting Black Box Models Via Hypothesis Testing,” in *Proceedings of the 2020 ACM-IMS on Foundations of Data Science*, 47–57. [1]

- Candes, E., Fan, Y., Janson, L., and Lv, J. (2018), "Panning for Gold: Model Knockoffs for High Dimensional Controlled Variable Selection," *Journal of the Royal Statistical Society, Series B*, 80, 551–577. [2,14]
- Carbonetto, P., Stephens, M. (2012), "Scalable Variational Inference for Bayesian Variable Selection in Regression, and Its Accuracy in Genetic Association Studies," *Bayesian Analysis*, 7, 73–108. [2]
- Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), "The Horseshoe Estimator for Sparse Signals," *Biometrika*, 97, 465–480. [12]
- Castillo, I., Schmidt-Hieber, J., and Van der Vaart, A. (2015), "Bayesian Linear Regression With Sparse Priors," *The Annals of Statistics*, 43, 1986–2018. [2]
- Cesa-Bianchi, N., and Lugosi, G. (2012), "Combinatorial Bandits," *Journal of Computer and System Sciences*, 78, 1404–1422. [3]
- Chen, W., Wang, Y., and Y. Yuan (2013), "Combinatorial Multi-Armed Bandit: General Framework and Applications," in International Conference on Machine Learning. [3,5]
- Chipman, H., George, E. I., and McCulloch, R. E. (2001), "The Practical Implementation of Bayesian Model Selection," in *Institute of Mathematical Statistics Lecture Notes –Monograph Series*. Institute of Mathematical Statistics, 38, pp. 65–116. [1,4]
- Chipman, H. A., George, E. I., and McCulloch, R. E. (2010), "BART: Bayesian Additive Regression Trees," *The Annals of Applied Statistics*, 4, 266–298. [5,9]
- Combes, R., and Proutiere, A. (2014), "Unimodal Bandits: Regret Lower Bounds and Optimal Algorithms," in International Conference on Machine Learning, Beijing, China. [3]
- Even-Dar, E., Mannor, S., and Mansour, Y. (2006), "Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems," *Journal of Machine Learning Research*, 7, 1079–1105. [7,16]
- Fahy, C., and Yang, S. (2019), *Dynamic Feature Selection for Clustering High Dimensional Data Streams* (Vol. 7), IEEE. [13]
- Fan, J., and Lv, J. (2008), "Sure Independence Screening for Ultrahigh Dimensional Feature Space," *Journal of the Royal Statistical Society, Series B*, 70, 849–911. [12]
- Fisher, A., Rudin, C., and Dominici, F. (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, 1–81. [1]
- Foster, D. P., and Stine, R. A. (2008), " α -Investing: A Procedure for Sequential Control of Expected False Discoveries," *Journal of the Royal Statistical Society, Series B*, 70, 429–444. [9]
- Friedman, J., Hastie, T., and Tibshirani, R. (2001), *The Elements of Statistical Learning* (Vol. 1), Springer Series in Statistics, New York: Springer. [12]
- Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19, 1–141. [8]
- Gai, Y., Krishnamachari, B., and Jain, R. (2012), "Combinatorial Network Optimization With Unknown Variables: Multi-Armed Bandits With Linear Rewards and Individual Observations," *IEEE/ACM Transactions on Networking*, 20, 1466–1478. [3,4,5]
- 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, 399–434. [1]
- George, E. I. and R. E. McCulloch (1993), "Variable Selection Via Gibbs Sampling," *Journal of the American Statistical Association*, 88, 881–889. [2]
- (1997), "Approaches for Bayesian Variable Selection," *Statistica Sinica*, 7, 339–373. [2,3]
- Gupta, S., Joshi, G., and Yagan, O. (2020), "Correlated Multi-Armed Bandits With a Latent Random Source," in IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain. [7]
- Hill, J., Linero, A., and Murray, J. (2020), "Bayesian Additive Regression Trees: A Review and Look Forward," *Annual Review of Statistics and Its Application* 7, 251–278. [4]
- Hooker, G. (2007), "Generalized Functional ANOVA Diagnostics for High-Dimensional Functions of Dependent Variables," *Journal of Computational and Graphical Statistics*, 16, 709–732. [1]
- Horel, E., and Giesecke, K. (2019), "Towards Explainable AI: Significance Tests for Neural Networks," arXiv:1902.06021. [1,5]
- Ishwaran, H. (2007), "Variable Importance in Binary Regression Trees and Forests," *Electronic Journal of Statistics*, 1, 519–537. [1]
- Javanmard, A., Montanari, A. (2018), "Online Rules for Control of False Discovery Rate and False Discovery Exceedance," *The Annals of Statistics*, 46, 526–554. [13]
- Johnson, V. E., and Rossell, D. (2012), "Bayesian Model Selection in High-Dimensional Settings," *Journal of the American Statistical Association*, 107, 649–660. [2]
- Kazemtabar, J., Amini, A., Bloniarz, A., and Talwalkar, A. S. (2017), "Variable Importance Using Decision Trees," in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA. [1]
- 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. [3,6]
- Kveton, B., Wen, Z., Ashkan, A., Eydgahi, H., and Eriksson, B. (2014), "Matroid Bandits: Fast Combinatorial Optimization With Learning," in *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pp. 420–429. [6]
- Kveton, B., Wen, Z., Ashkan, A., and Szepesvari, C. (2015), "Combinatorial Cascading Bandits," *Advances in Neural Information Processing Systems*, 28, 1450–1458 [3]
- Lafferty, J., and Wasserman, L. (2008), "RODEO: Sparse, Greedy Nonparametric Regression," *The Annals of Statistics*, 6, 28–63. [2]
- Lai, T., and Robbins, H. (1985), "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Mathematics*, 6, 4–22. [2,7]
- Lei, J., GSell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018), "Distribution-Free Predictive Inference for Regression," *Journal of the American Statistical Association*, 113, 1094–1111. [1]
- Leike, J., Lattimore, T., Orseau, L., and Hutter, M. (2016), "Thompson Sampling is Asymptotically Optimal in General Environments," in *Conference on Uncertainty in Artificial Intelligence*. Virginia, United States: AUAI Press. [6]
- Li, R., Zhong, W., and Zhu, L. (2012), "Feature Screening Via Distance Correlation Learning," *Journal of the American Statistical Association*, 107, 1129–1139. [12]
- Liang, F., Li, Q., and Zhou, L. (2018), "Bayesian Neural Networks for Selection of Drug Sensitive Genes," *Journal of the American Statistical Association*, 113, 955–972. [9,11,14]
- Lin, Y., and Zhang, H. H. (2006), "Component Selection and Smoothing in Multivariate Nonparametric Regression," *The Annals of Statistics*, 34, 2272–2297. [2]
- Linero, A. R. (2018), "Bayesian Regression Trees for High-Dimensional Prediction and Variable Selection," *Journal of the American Statistical Association*, 113(522), 626–636. [9,11]
- Linero, A. R., and Yang, Y. (2018), "Bayesian Regression Tree Ensembles That Adapt to Smoothness and Sparsity," *Journal of the Royal Statistical Society, Series B*, 80, 1087–1110. [9,11]
- Liu, Y., Ročková, V., and Wang, Y. (2018), "ABC Variable Selection With Bayesian Forests," arXiv:1806.02304. [16]
- Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013), "Understanding Variable Importance in Forests of Randomized Trees," *Advances in Neural Information Processing Systems*, 1, 431–439. [5]
- Lu, Y., Fan, Y., Lv, J., and Noble, W. S. (2018), "DeepPINK: Reproducible Feature Selection in Deep Neural Networks," *Proceedings of the 32nd International Conference in Neural Information Processing Systems*, pp. 8690–8700. [1]
- Mase, M., Owen, A. B., and Seiler, B. (2019), "Explaining Black Box Decisions by Shapley Cohort Refinement," arXiv:1911.00467. [1]
- Mitchell, T. J., and Beauchamp, J. J. (1988), "Bayesian Variable Selection in Linear Regression," *Journal of the American Statistical Association*, 83, 1023–1032. [2]
- Narisetty, N. N., He, X. (2014), "Bayesian Variable Selection With Shrinking and Diffusing Priors," *The Annals of Statistics*, 42, 789–817. [2,3]
- Ni, J., Neslin, S. A., and Sun, B. (2012), "Database submission—The ISMS durable goods data sets," *Marketing Science*, 31(6), 1008–1013. [14]
- Olden, J. D., and Jackson, D. A. (2002), "Illuminating the Lack Box: A Randomization Approach for Understanding Variable Contributions in Artificial Neural Networks," *Ecological Modelling*, 154, 135–150. [1]

- Owen, A. B., and Prieur, C. (2017), “On Shapley Value for Measuring Importance of Dependent Inputs,” *SIAM/ASA Journal on Uncertainty Quantification*, 5, 986–1002. [1,4]
- Pandey, S., Chakrabarti, D., and Agarwal, D. (2007), “Multi-Armed Bandit Problems With Dependent Arms,” in International Conference on Machine Learning, Corvallis, OR. [7]
- Patterson, E., and Sesia, M. (2018), *Knockoff: The Knockoff Filter for Controlled Variable Selection*, Statistics Department, Stanford University. R package version 0.3.2. [14]
- Radchenko, P., and James, G. M. (2010), “Variable Selection Using Adaptive Nonlinear Interaction Structures in High Dimensions,” *Journal of the American Statistical Association*, 105, 1541–1553. [2]
- Ramdas, A., Yang, F., Wainwright, M. J., and Jordan, M. I. (2017), “Online Control of the False Discovery Rate With Decaying Memory,” in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, eds. I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. [13]
- Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. (2009), “Sparse Additive Models,” *Journal of the Royal Statistical Society, Series B*, 71, 1009–1030. [2]
- Rhee, S.-Y., Fessel, W. J., Zolopa, A. R., Hurley, L., Liu, T., Taylor, J., Nguyen, D. P., Slome, S., Klein, D., Horberg, M., Flamm, J., Follansbee, S., Schapiro, J. M., and Shafer, R. W. (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, 456–465. [14]
- Rhee, S.-Y., Taylor, J., Wadhera, G., Ben-Hur, A., Brutlag, D. L., and Shafer, R. W. (2006), “Genotypic Predictors of Human Immunodeficiency Virus Type 1 Drug Resistance,” *Proceedings of the National Academy of Sciences*, 103, 17355–17360. [14]
- Ročková, V., and George, E. I. (2014), “EMVS: The EM Approach to Bayesian Variable Selection,” *Journal of the American Statistical Association*, 109, 828–846. [2]
- (2018), “The Spike-and-Slab LASSO,” *Journal of the American Statistical Association*, 113, 431–444. [2,3,12,14]
- Rossell, D., and Telesca, D. (2017). Nonlocal priors for high-dimensional estimation. *Journal of the American Statistical Association*, 112, 254–265. [2]
- Russo, D. (2016), “Simple Bayesian Algorithms for Best Arm Identification,” in Conference on Learning Theory, New York, USA. [7,16]
- Scheipl, F. (2011), “spikeslabgam: Bayesian Variable Selection, Model Choice and Regularization for Generalized Additive Mixed Models in r,” arXiv:1105.5253. [2]
- Shapley, L. S. (1953), “A Value for n-Person Games,” *Contributions to the Theory of Games*, 2, 307–317. [4]
- Thompson, W. R. (1933), “On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Sample,” *Biometrika*, 25, 285–294. [2]
- Tibshirani, R. (2011), “Regression Shrinkage and Selection Via the LASSO: A Retrospective,” *Journal of the Royal Statistical Society*, 73, 273–282. [12]
- van der Pas, S., Scott, J., Chakraborty, A., and Bhattacharya, A. (2019), *horseshoe: Implementation of the Horseshoe Prior* (R package version 0.2.0). [12]
- van der Pas, S., Szabó, B., van der Vaart, A. (2017), “Uncertainty Quantification for the Horseshoe” (with discussion), *Bayesian Analysis*, 12, 1221–1274. [12]
- Vannucci, M., and Stingo, F. C. (2010), “Bayesian Models for Variable Selection That Incorporate Biological Information,” *Bayesian Statistics*, 9, 1–20. [2]
- Wang, J., Shen, J., and Li, P. (2018), “Provable Variable Selection for Streaming Features,” in International Conference on Machine Learning, Stockholm, Sweden. [13]
- Wang, S., and Chen, W. (2018), “Thompson Sampling for Combinatorial Semi-Bandits,” in *International Conference on Machine Learning*, pp. 5114–5122. [3,5,6,7]
- Ye, M., and Sun, Y. (2018), “Variable Selection Via Penalized Neural Network: A Drop-Out-One Loss Approach,” in International Conference on Machine Learning, Stockholm, Sweden. [1]
- Zhang, T., Ge, S. S., and Hang, C. C. (2000), “Adaptive Neural Network Control for Strict-Feedback Nonlinear Systems Using Backstepping Design,” *Automatica*, 36, 1835–1846. [1]
- Zhou, J., Foster, D. P., Stine, R. A., and Ungar, L. H. (2006), “Streamwise Feature Selection,” *Journal of Machine Learning Research*, 7, 1861–1885. [4,5,13]