Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Econometrics and Management Science

Continual learning using the Variational Soft Random Forest

Naim Achahboun (529911)

Supervisor:	Eoghan O'Neill
Second assessor:	Richard Paap
Date final version:	26th April 2024

Abstract

In this paper, the use of variational inference to estimate decision tree ensembles suitable for continual learning is studied. Building on the work by Cinquin et al. (2023) and Salazar (2024), who introduced variational regression trees, the Variational Soft Random Forest (VSRF) is introduced. The VSRF is an ensemble model of variational regression trees. To allow for increased flexibility, the leaf nodes within a variational regression tree are extended to accommodate both splines and linear forms. Additionally an online Bayesian update is proposed, to allow for efficient re-estimation of the VSRF model. The VSRF is evaluated on a variety of datasets and a main application to a limited-pull contextual multi-armed bandit problem is considered. It is shown that the VSRF model can reduce the predictive variance of a single variational regression tree effectively. For some evaluation datasets, the VSRF can match the out-of-the-box performance of state-of-the-art ensemble methods. However for other evaluation datasets, the VSRF model performs worse. In particular the VSRF is not deemed suitable for use in settings with high-dimensional data and a low signal. As for the proposed online Bayesian update, it can reduce computation time by a factor of 11 relative to offline estimation, but the quality of the resulting predictions is highly dependent on the underlying data.

Contents

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful. In this section, I want to express my gratitude to several people who aided me during my research. Firstly, I would like to thank my supervisor Dr. Eoghan O'Neill for his feedback and insightful suggestions. I also want to thank my family and friends for providing me with encouragement and support.

Chapter 1

Introduction

Ensemble models composed of decision trees are renowned for their strong predictive performance. Among these, Bayesian ensembles such as BART (Chipman et al., 2010) and its variants (Hill et al., 2020) excel in prediction tasks while also offering valuable confidence intervals for predictions.

However, these benefits come at a price. The computational demands of the underlying Markov Chain Monte Carlo (MCMC) algorithms used to estimate the posterior distribution of BART make it hard to scale to large datasets or models with many parameters. For the Bayesian CART algorithm used by BART, the mixing time of a Markov chain grows exponentially with the number of observations (Kim & Rockova, 2023; Ronen et al., 2022). This complicates achieving convergence on large datasets. Additionally there are limited gains from parallel computation given that estimation procedures like back-fitting, require draws in a specific order. There have been recent advances that successfully improve the estimation time of BART (He et al., 2019; O'Neill, 2021). Mainly by improving the posterior inference algorithms used.

In this paper, a different approach is taken by exploring new Bayesian decision tree ensemble models that can be estimated using variational inference (VI). Contrary to algorithms based on MCMC, variational inference formulates the problem of finding the posterior as an optimization problem, which means that it can be scaled effectively using advances in distributed stochastic optimization. My main research question is:

> How can variational regression trees be combined effectively into a forest model, that is suitable for continual learning?

This paper is built on the foundations laid by Cinquin et al. (2023) and Salazar (2024), who introduced variational soft regression trees. A variational soft regression tree is a decision tree with oblique splits and soft decision boundaries that is estimated using variational inference. Because variational soft trees can be estimated relatively quickly, have strong predictive performance, and can produce confidence intervals, they are strong candidates for usage in online settings.

I investigate how a forest of variational soft trees can be used for continual learning. Continual learning refers to the ability of a model to adapt and learn from new information in an efficient way. Models capable of continual learning are desirable in settings where data distribution shifts occur frequently, and models need to be updated quickly to retain their predictive

performance. Continual learning is natural within a Bayesian context as it can be seen as a posterior update of the model given some newly observed evidence.

My research is relevant to online platforms and retailers, as they rely heavily on personalized user experiences to enhance engagement and drive sales. Given the dynamic nature of user behavior, the ability to swiftly update predictive models with new information is paramount to avoid providing irrelevant suggestions. My research is also relevant for large-scale prediction problems in general where there is a need for models that can produce approximate confidence intervals and allow for estimation in a scalable way.

To answer my research question, I propose the Variational Soft Random Forest (VSRF), which is an ensemble model of several variational soft regression trees. The type of ensemble used for the VSRF is similar to that of the Random Forest model (Breiman, 2001). To facilitate continual learning with the VSRF, an online Bayesian update is derived based on the Streaming Variational Bayes (SVB) framework as described by Chérief-Abdellatif et al. (2019). This update allows efficient re-estimation of the VSRF on micro-batches of new data, without having to re-estimate the entire model from scratch.

The VSRF is compared against several state-of-the-art ensemble models on a variety of datasets. Furthermore, the Scarce Promotion Problem (SPP) is considered as a main application. The Scarce Promotion Problem involves personalized promotion selection in the context of limited promotion availability. This application is motivated by existing literature on scarce offers (Barton et al., 2022), which underscores its relevance for online retailers. The SPP was selected as a main application because it can illustrate the benefits of models capable of continual learning. The SPP is addressed by framing it as a limited-pull contextual multi-armed bandit problem. Which is solved using an adjusted Thompson sampling algorithm (Thompson, 1933) that utilizes draws from the posterior of the VSRF.

Empirical evidence shows that the VSRF model reduces the predictive variance compared to a single variational regression tree. Also the VSRF model appears competitive with state-of-the-art ensemble models for several benchmark datasets. However, there were also benchmark datasets for which the performance of the VSRF is clearly worse relative to the other ensemble models. In some cases, the VSRF is unable to improve over the predictions of a single variational regression tree. The VSRF model is not deemed suitable for use in settings with high-dimensional data and a low signal, as the VSRF breaks down when a large number of covariates is used. Hence I conclude that the VSRF is a valuable model, but that it must not be used blindly but rather after careful evaluation has taken place of several candidate models for the prediction problem at hand.

Relative to offline estimation, the proposed online Bayesian update yields an 11-fold reduction in computation time. This drop in computation time is associated with a significant loss in predictive performance for some benchmark datasets, but can also yield improvements for other benchmark datasets considered.

As for the main application to the Scarce Promotion Problem the Thompson sampler using the VSRF model performs well relative to other benchmarks. However, due to the high standard deviations observed in the simulation, I am hesitant to consider this finding as robust.

The paper is structured as follows. In Chapter 2 the relevant literature for my work is

discussed. Chapter 3 contains information on the data used for performance evaluation and my main application. In Chapter 4 the VSRF model and its estimation method are introduced.

In Chapter 5 the predictive performance of the VSRF on the main application and a variety of datasets is evaluated. Finally, in Chapter 6, I summarize the findings of my research and suggest areas for future exploration.

1.1 Contributions

To the best of my knowledge, the original contributions of my work are:

- Extension of the linear leaf nodes of a Variational Regression Tree to splines, to make the model more flexible.
- Proposal and evaluation of a new Variational Soft Random Forest (VSRF) model which combines variational soft trees into a forest.
- Specification of an online Bayesian update procedure for the VSRF based on the work of Chérief-Abdellatif et al. (2019) to facilitate continual learning.

Chapter 2

Related work

In this section, several relevant papers for my work are discussed. First research on Bayesian tree ensembles is reviewed. Then online variational estimation techniques are discussed and afterwards, variational decision trees are covered. I conclude with the relevant literature on the constrained multi-armed bandits problem, which is relevant for my main application of the VSRF model to the Scarce Promotion Problem.

2.1 Bayesian tree ensembles

Ensembles of decision trees are known to be strong predictors for many use cases. Examples from models in this family are: RandomForest (Breiman, 2001), Xgboost (Chen et al., 2015), and Catboost (Dorogush et al., 2018). When uncertainty estimates of the predictions are desirable, Bayesian tree ensemble models provide a natural framework to obtain these.

The most popular Bayesian tree ensemble model is the Bayesian Additive Regression Trees (BART) model introduced by Chipman et al. (2010). A BART model is comprised of a sum of decision trees where each tree is kept relatively small by the choice of an appropriate prior. Although the predictive performance of the BART model is excellent, it is hard to scale effectively to large datasets. Convergence takes a long time, given that the mixing time of the Bayesian CART algorithm grows exponentially in the number of observations (Kim & Rockova, 2023; Ronen et al., 2022). Additionally the iterative back-fitting MCMC algorithm (Chipman et al., 2010) has dependencies between the iterations, limiting the possibilities for parallel computation.

In the literature review on BART by Hill et al. (2020), it is mentioned that the scalability issues of BART models are not due to inefficient implementations, but rather because the underlying algorithms for posterior inference need to be improved.

To deal with this issue and explore other possibilities, many variations and improvements of BART have been proposed by subsequent studies. For example the work by He et al. (2019) who propose the XBART model which is estimated using a stochastic hill-climbing algorithm that is much faster and less memory intensive compared to the back-fitting MCMC algorithm.

Or the work by O'Neill (2021) who develops an alternative implementation of BART that uses importance sampling (BART-IS). Its advantages are that it is fast and embarrassingly parallel because it makes independent draws of many sums of tree models. Another novel sampling algorithm for BART has also been proposed by Lakshminarayanan et al. (2015). They propose

using a sampler based on a top-down particle filtering algorithm (Lakshminarayanan et al., 2013) combined with a Particle Gibbs algorithm (Andrieu et al., 2010). Experiments show that their sampler outperforms the back-fitting algorithm of Chipman et al. (2010) in most cases.

There have also been developments trying to facilitate efficient distributed estimation of BART models, for example using the Consensus Monte Carlo algorithm proposed by Scott et al. (2022). The Consensus Monte Carlo algorithm uses weighted averages of Monte Carlo draws produced in parallel on multiple machines to approximate draws from a single machine. The core focus of the algorithm is limiting the communication between machines as this is expensive in a distributed context.

In my work, the focus is also on efficient estimation of Bayesian ensemble models but I do not explore ways to improve posterior inference for BART models. Rather a new Bayesian ensemble model is proposed that uses variational inference for scalable estimation. The VSRF model introduced is not a variant of BART, but it bears similarity to several variants of BART. The first is the SoftBart model introduced by Linero (2022). The SoftBart model similar to the VSRF replaces the decision trees within the model with soft decision trees (Irsoy et al., 2012). The motivation of Linero (2022) is mainly to improve predictive performance by obtaining smooth predictions. He makes use of a Bayesian back-fitting algorithm to estimate SoftBart. However, given the soft structure of SoftBart, it can also be estimated using variational inference techniques if a suitable variational family can be found.

A second variant of BART that is similar to the VSRF was proposed by Cochrane et al. (2023). They use Hamiltonian Monte Carlo (HMC) methods to sample the posterior of a BART model with soft oblique splits. The authors are motivated to use a HMC approach, because they want to improve posterior inference by utilizing information on the curvature of the likelihood when sampling candidate draws. Soft oblique splits are introduced because HMC approaches require gradient information, and gradients are not well defined with respect to the discrete parameters present in a hard tree. This motivation to soften the tree is identical for models estimated using variational inference, although second-order gradient information is not required for variational inference.

2.2 Online variational inference

Variational inference (VI) is an optimization technique that can be used to approximate complex posterior densities of Bayesian models. The goal of variational inference techniques is to approximate the true posterior by a surrogate posterior chosen from a family of distributions that are easy to sample from. Once a family is chosen, an optimization problem is solved by choosing the parameters of the surrogate posterior in such a way that it is most similar to the true posterior. Where the most widely used method in VI to quantify the similarity between the surrogate and true posterior is the Kullback-Leibler (KL) divergence. Other ways to measure similarity between the true posterior and surrogate posterior have also been proposed, for example, the Wasserstein distance (Ambrogioni et al., 2018).

Variational inference methods can be applied in offline settings where the amount of observations is fixed, but VI can also be applied in online settings. This is crucial for many modern systems where data streams of new observations are continuously observed. Applications of

online variational inference are numerous. For example Wang et al. (2011) develop an online variational inference algorithm for a Hierarchical Dirichlet Process. Their algorithm can be used to cluster documents using only a single pass over all documents, which makes it ideal for usage on large corpora of text.

Often the difficulty when proposing a variational inference algorithm is that for each model a new set of variational update equations must be derived. Hence Ranganath et al. (2014) propose a black-box variational inference algorithm using Monte Carlo sampling and gradient descent with few requirements on the underlying model. In my work I build on top of their black-box algorithm for the offline estimation of my VSRF model, only making use of the model-specific reparameterizations proposed by Salazar (2024), to reduce variance during optimization.

My optimization approach is similar to the approach taken by the Automatic Differentiation Variational Inference (ADVI) method introduced by Kucukelbir et al. (2017). ADVI allows for easy estimation of a broad class of Bayesian models using VI, requiring the researcher to only define the model in Stan (Gelman et al., 2015) and provide a dataset. The similarity with my approach is the usage of automatic differentiation to obtain gradients during the optimization. However because the VSRF is implemented using Pytorch (Paszke et al., 2019), I do not evaluate whether the VSRF can be estimated using the ADVI method.

My online estimation algorithm is based on the work by Broderick et al. (2013), who introduce Streaming Variational Bayes (SVB) a framework for performing online posterior updates. They consider an application of their work to a Latent Dirichlet Allocation model, and compare their SVB-based algorithm, against an offline stochastic variational inference algorithm, and show its advantages. The update rule of SVB has been proposed by other works as well, for example, the study by Zeno et al. (2018) who estimate Bayesian neural networks and show that the obtained posterior distributions over the weights make the network more robust.

There has also been theoretical work on the quality of the SVB by Chérief-Abdellatif et al. (2019), who study generalization bounds for online variational inference algorithms and derive a generalization bound for a special variant of SVB. My online variational inference algorithm for the VSRF model is based on the special variant analyzed by Chérief-Abdellatif et al. (2019). However, I do not study whether the VSRF model satisfies all assumptions required for the proposed generalization bound to hold.

2.3 Variational soft decision trees

The foundations for variational decision trees were laid by Irsoy et al. (2012), who introduced soft decision trees. Soft decision trees use soft gating functions instead of hard decision boundaries in the nodes of a tree. When a hard decision boundary is used, an observation always visits only one of the child nodes. On the other hand, a soft tree uses an oblique soft split by calculating the probability for each observation to visit a child node. This increased flexibility can improve the ability of a soft tree to capture complex patterns. This is supported by the study of Frosst and Hinton (2017), who effectively distill a neural network into a soft tree to allow for more interpretable predictions.

Variational inference is suitable for estimating soft trees due to the absence of rigid decision boundaries, allowing for gradient computation of the Evidence Lower Bound (ELBO). To my

knowledge Cinquin et al. (2023) are the first to use variational techniques to estimate a soft decision tree. They build a boosted model of fixed-depth soft decision trees (VBST). Where each individual soft tree is estimated using variational inference. They evaluate the performance of their VBST on several datasets against XBART (He et al., 2019), SGLB (Ustimenko & Prokhorenkova, 2021), and XGboost (Chen et al., 2015) and conclude that their model is competitive. My work also considers an ensemble of variational trees, but I estimate all trees jointly instead of applying a boosting step after the individual soft trees have been estimated.

Another crucial paper for my work is the paper on variational regression trees (VaRT) by Salazar (2024). Contrary to the approach taken by Cinquin et al. (2023), Salazar (2024) utilizes a non-parametric Bayesian prior, enabling tree growth up to a predefined maximum depth rather than using a fixed-depth tree. He shows that his VaRT model is competitive on a large variety of datasets with Catboost (Dorogush et al., 2018), Xgboost (Chen et al., 2015), and RandomForest (Breiman, 2001). My work can be viewed as a direct extension of his method by generalizing it to a forest specification, introducing splines in the leaf nodes, and introducing an online estimation technique.

2.4 Contextual multi-armed bandits

As an application for the VSRF model, the Scarce Promotion Problem (SPP) is studied. The Scarce Promotion Problem involves an agent allocating a set of scarce promotional offers to a set of customers which arrive sequentially. The SPP can be cast as a constrained contextual multi-armed bandit problem. Because multi-armed bandit problems are studied in a large variety of disciplines, there has been a lot of research in this area. For a thorough introduction see Slivkins et al. (2019).

In general, a multi-armed bandit problem is a problem where a decision maker, referred to as a bandit, has to choose one of several alternatives, commonly called arms in the literature. The decision maker wants to maximize some reward. However he has no knowledge of the exact reward of each arm, hence he has to learn by exploring the arms while exploiting the most rewarding arms at the same time. In the contextual variant of the multi-armed bandit problem, the decision maker receives some information signal called a context before he pulls an arm. This context can be general or specific to each arm. Finally, when the decision maker receives information signals and has to take into account one or several constraints while choosing arms, the problem becomes a constrained contextual multi-armed bandit.

In my work, I focus on a specific type of constrained contextual multi-armed bandit problem where there is a budget constraint imposed on all arms except one. My problem can be formulated as a variant of the bandits with knapsacks problem formalized by Badanidiyuru et al. (2018). They introduce two algorithms to solve bandits with knapsacks problems, the first is a high-level framework called BalancedExploration which makes use of an LP relaxation and iteratively updates a set of potentially LP-perfect distributions over the arms. These distributions are used to pick the arm in a round. The second algorithm is a primal-dual greedy algorithm that aims to pick the action with the highest reward per unit of resource consumption.

Although the algorithms introduced by Badanidiyuru et al. (2018) have favorable regret bounds in this work, I propose another algorithm attempting to exploit the specific structure

of the SPP based on Thompson sampling (Thompson, 1933). My main motivation to use a Thompson sampler combined with the VSRF, is to allow for explainable decisions. The idea to combine a Thompson Sampler with a Bayesian predictive model has been proposed before and is actively used in the industry. See for example the study by Collier and Llorens (2018), who combine a Thompson sampler with a Bayesian neural network to solve a contextual multi-armed bandit problem at Hubspot.

My work also bears similarity to the approach taken by Féraud et al. (2016). They consider an unconstrained contextual multi-armed bandit but do consider a non-linear reward function and propose a similar method compared to my VSRF, which is a random forest that can be updated sequentially in an online setting called a bandit forest. They study its performance relative to baseline methods like a LinUCB algorithm and Multilayer Perceptron (MLP). They find that their bandit forest beats the baselines for most datasets considered.

My work differs from the paper by Féraud et al. (2016) in two crucial points, the first point is that in this work estimation takes place within a Bayesian context and the uncertainty estimates of the posterior are used when deciding to take an action. The second difference is that in this work a constrained version of the contextual multi-armed bandit problem is studied, which is known to be a more difficult problem to solve effectively.

Chapter 3

Data

Table 3.1: The names, number of features p and the number of observations n of the evaluation datasets.

Dataset	p	n
autompg	7	392
energy	8	768
forest	12	517
stock	11	536
concrete	8	1030
solar	10	1066
airfoil	5	1503
housing	13	506

To perform a comprehensive evaluation of the predictive performance of the VSRF, a selection of datasets from the UCI machine learning repository (Asuncion & Newman, 2007) is used. The datasets selected are from several domains to asses how the predictive performance of the VSRF varies across different prediction tasks. The names, number of features p, and the number of observations n of the datasets can be seen in Table 3.1.

A short description is given for the prediction task of each dataset, starting with the autompg dataset where the prediction task involves predicting the miles per gallon (mpg) based on the characteristics of a vehicle. For the energy dataset, the task is to predict the energy efficiency of a building given properties of the building. The forest dataset is used to predict the total area that was burned during a forest fire in Portugal, given meteorological and geographical data. The stock dataset is used to predict the returns of the Istanbul Stock Exchange. The concrete dataset is used to predict the compressive strength of concrete given its age and ingredients. For the solar dataset, the task is to predict the number of solar flares in a 24-hour period. The airfoil dataset was produced by NASA, the task is to predict the sound pressure level from data obtained during experiments in a wind turbine. Finally, for the housing dataset, the task is to predict the median value of homes within a suburb.

For my main application to the Scarce Promotion Problem, a simulation using the Hillstrom dataset (Hillstrom, 2008) is constructed. The Hillstrom dataset contains data on 64000 customers who made a purchase at an online store and were involved in a large-scale randomized experiment. Each customer was randomly allocated into one of three groups. The first group

received an email promoting men's merchandise, the second group received an email promoting women's merchandise and the last group did not receive any email. During a follow-up period of two weeks, each person was monitored to determine whether the e-mail marketing campaign led to new purchases in the online store. The context of e-mail marketing is not identical to an online shopping context but given limited availability of datasets in this area, it will serve as a meaningful proxy.

The simulation study has an observational component and a simulated component. The observational component is the set of customers in the Hillstrom dataset and their past purchase behaviour. The simulated component is the present purchase value given a promotion, which is generated from a known data generating process. This way of constructing the simulation allows for evaluation on covariates of a real dataset, while still being able to evaluate the performance of the Thompson sampler effectively.

Chapter 4

Methods

The methodology is split into several parts. First, the structure of the VSRF model is introduced. Then offline variational inference for the VSRF is discussed. Afterwards, an online Bayesian update is specified for the VSRF. Then the Scarce Promotion Problem is formalized and the heuristic based on Thompson sampling is introduced.

4.1 Structure of the VSRF model

The VSRF model can be used to solve regression problems. It builds on top of the work done by Salazar (2024). In essence, the VSRF model is a mixture of several variational soft decision trees. The soft trees use gating functions to mimic the hard decision boundaries found in regular decision trees. In the context of a regression problem with B pairs $(y_1, x_1), .., (y_B, x_B)$ of the observed target variable and covariates. The prediction $h(x_i)$ of a VSRF comprised of T trees, where each tree t has L_t leaf nodes is given by:

$$h(x_i) = \sum_{t=1}^{T} w_t \sum_{l \in L_t} p(l|x_i) f_l(x_i)$$

Where x_i is a p dimensional vector of covariates. The weight given to a tree t is equal to w_t, and $p(l|x_i)$ is the probability that an observation x_i ends up in leaf l. The prediction of a single leaf l is defined by $f_l(x_i)$. Given that the weights of the trees w_t for $t = 1, .., T$, are chosen to form a convex combination, they can be interpreted as probabilities for an observation x_i to traverse through a tree t. By virtue of this, the prediction of a VSRF can be interpreted as a convex combination of the predictions of the leaves of all trees within the forest.

In order to perform variational inference the likelihood of the VSRF model has to be specified. For a single observation, the likelihood can be specified as:

$$p(y_i|x_i, \Phi, \Psi) = \sum_{t=1}^{T} w_t \sum_{l \in L_t} p(l|x_i, \Phi_t) p(y_i|x_i, \Psi_{tl})$$

Where Φ contains the interior node parameters for all trees, and Ψ contains the parameters for the leaf nodes of all trees. The interior node parameters of a single tree t are denoted by Φ_t. The parameters within Φ_t that are specific to an interior node n are given by Φ_{tn}, additionally

the leaf parameters of a single leaf l in a tree t are captured by Ψ_{tl}. The likelihood of a single variational soft tree is equal to the weighted likelihood contribution of each leaf. Where the weight is given by the probability that an observation x_i ends up in leaf l. This probability is defined as:

$$p(l|x, \Phi_t) = \prod_{n \in P_l} g_n(x_i, \Phi_{tn})^{\mathbb{I}[l \in Q_n^R]} (1 - g_n(x_i, \Phi_{tn}))^{\mathbb{I}[l \in Q_n^L]}$$

Where the set of all nodes on the path from the root of a tree to a leaf l is denoted by P_l. The set of nodes in the right subtree of a node n is given by Q_n^R, and the set of nodes in the left subtree of a node n is given by Q_n^L. The gating function for a node n, $g_n(x, \Phi_{tn})$ is chosen to be a logistic function given by:

$$g_n(x_i, \Phi_{tn}) = \frac{e^{\beta_n^T x_i}}{1 + e^{\beta_n^T x_i}}$$

Where β_n are parameters of the node n. For this choice of gating function, it follows that $\Phi_t = \{\beta_n\}_{n \in N_t}$. Where N_t denotes the number of interior nodes in tree t. Other gating functions could also be considered like a Gaussian CDF. I have decided to use the logistic gating function as both Cinquin et al. (2023) and Salazar (2024) find good empirical performance using the logistic form.

4.1.1 Normalization of observations

All covariates and the target variable are normalized to the unit interval, using min-max normalization. If covariates are used that are defined on different intervals, then the covariates with the largest values will dominate when determining the soft splits within an interior node in a tree. This domination is not desirable as there is no reason to believe that covariates with a larger magnitude are more relevant predictors. Normalization also aids with numerical stability when performing variational inference because there are no large differences in the magnitude of the features.

4.1.2 Tree weights

In this section several weighting methods are proposed for the VSRF. For a weighting method, it is desirable that the weights form a convex combination of the predictions of the trees, as this allows for predictions that are easier to interpret. The first weighting method considered is that of a Random Forest (Breiman, 2001), where each tree is given equal weight:

$$w_t = \frac{1}{T} \quad t = 1, .., T$$

The second weighting method is called inverse depth weighting. The idea behind inverse depth weighting is to give the most flexible trees a relatively smaller contribution to the predictions of the VSRF, with the aim to reduce overall variance of the model. Let d_t denote the maximum

depth of a tree t. Then the inverse depth weight is defined as:

$$w_t = \frac{\frac{1}{d_t}}{\sum_{i=1}^{T} \frac{1}{d_i}} \quad t = 1, .., T$$

The decision to use weighting methods instead of estimating the weights was driven by the results of several initial experiments where the weights were estimated as parameters. These experiments revealed that the performance of the VSRF deteriorates. Mainly because when the weights are learned, a single tree often receives almost all weight, and the other trees have almost no weight at all.

Further research into this tree selection problem revealed two potential causes. The first cause is that once a single tree gets a relatively high weight, its parameters have a larger impact on the gradient updates in future iterations of the variational inference algorithm. Because the parameters in low-weight trees have smaller contributions, their gradient updates become relatively small. This is similar to the vanishing gradient problem observed in deep neural networks. The result in case of the VSRF is that a single tree ends up dominating the forest.

The second likely cause is that in the experiments the weights were drawn from a low-variance Multivariate Gaussian prior. Subsequently these weights were transformed by taking squares and normalizing them, to obtain a convex combination. This normalization can also be a reason for tree selection to occur. Since squaring relatively small numbers, will diminish their contributions even more. A potential way to allow for weight estimation to take place without a normalization step would be to use a Dirichlet prior, which directly produces a convex set of weights. However because Dirichlet samples are drawn using acceptance-rejection sampling, special reparameterizations are required to obtain a low-variance gradient (Naesseth et al., 2017).

4.1.3 Functional form of leaf nodes

For the likelihood contribution of a leaf node, two forms are considered. The first form is used by Salazar (2024), which corresponds to a linear regression model with homoskedastic variance in each leaf. For a given linear leaf l the functional form is given by $f_l(x_i) = \kappa_{tl}^T x_i$. This gives rise to a likelihood contribution of:

$$p(y_i|x_i, \Psi_{tl}) = \phi(y_i; f_l(x_i), \sigma_{tl}^2)$$

Where $\phi(y; \mu, \sigma^2)$ denotes the PDF of a Gaussian distribution. and κ_{tl} is a vector of feature parameters. For this choice of leaf likelihood it follows that $\Psi_t = \{\kappa_{tl}, \sigma_{tl}^2\}_{l \in L_t}$.

For the second form, I propose using a linear spline within each leaf, to account for possible non-linearity in the leaf nodes. Let the basis function of the linear spline with d knots be given by:

$$h_1(x_i) = 1, \quad h_2(x_i) = x_i, \quad h_j(x_i) = \max(x_i - k_j, 0) \quad j = 3, ..d + 2$$

Where k_j are vectors of knots. The functional form for a spline leaf node with d knots is given

by $f_l(x_i) = \sum_{j=1}^{d+2} \kappa_{tlj}^T h_j(x_i)$. Which gives rise to a likelihood contribution of:

$$p(y_i|x_i, \Psi_{tl}) = \phi(y_i; f_l(x_i), \sigma_{tl}^2)$$

For this choice of leaf likelihood it follows that $\Psi_t = \{\kappa_{tl1}, .., \kappa_{tld+2}, \sigma_{tl}^2\}_{l \in L_t}$.

Preliminary experiments revealed that estimating the knots k_j as parameters in the model causes issues with convergence. In particular, because knots tend to overlap. Hence to mitigate this problem, I have decided to fix the knots k_j to be evenly spaced along the quantiles of the distribution of the covariates within the estimation sample.

4.1.4 Sampling a VSRF model

To sample a VSRF model, I extend the non-parametric prior of Salazar (2024) in a straightforward way from a single tree to a forest. A Variational Soft Random Forest with T trees, where each tree t has maximum depth d_t can be sampled by:

1. For each tree $t \in \{1, 2, ..., T\}$
2. Sample $s_{tj} \sim Bernoulli(\gamma_{tj})$, for $j \in \{1, ..., 2^{d_t - 1} - 1\}$
   (a) For each node $i \in \{1, 2, ..., 2^{d_t-1} - 1\}$ of tree t
      i. If $\prod_{k=1}^{floor(\frac{i}{2})} s_{tk} = 1$, and $s_{ti} = 1$ then sample $\beta_{ti} \sim \mathcal{N}(\mu_{ti}, S_{ti})$
      ii. If $\prod_{k=1}^{floor(\frac{i}{2})} s_{tk} = 1$, and $s_{ti} = 0$ then sample $\kappa_{ti} \sim \mathcal{N}(\zeta_{ti}, Z_{ti})$
   (b) For each final leaf node $i \in \{2^{d_t-1}, ..., 2^{d_t} - 1\}$ of tree t
      i. If $\prod_{k=1}^{floor(\frac{i}{2})} s_{tk} = 1$, then sample $\kappa_{ti} \sim \mathcal{N}(\zeta_{ti}, Z_{ti})$

Where $\{\gamma_{ti}\}_{i=1}^{2^{d_t-1}-1} \in [0, 1]^{2^{d_t-1}-1}$ denotes the probability for an interior node i in tree t to spawn two child nodes. The expected feature weights of an interior node i in a tree t are given by $\{\mu_{ti}\}_{i=1}^{2^{d_t-1}-1}$, and the expected feature weights within a leaf node i are given by $\{\zeta_{ti}\}_{i=2^{d_t-1}}^{2^{d_t}-1}$. The diagonal covariance matrices of the coefficients of the interior and leaf nodes are given by S_{ti} and Z_{ti}.

The sampling procedure starts by drawing the node splitting indicators s_{tj}. These indicator variables are used to determine the structure of the forest as it grows. Once the splitting indicators of a tree have been drawn, the interior node parameters and leaf node parameters are sampled. A condition is imposed based on the splitting indicators which corresponds to only drawing those leaves and interior nodes that are currently part of the tree. This can be understood by looking at Figure 4.1, where an example VSRF model is displayed. In Figure 4.1, the splitting indicator of the second tree s_{23} is equal to zero, indicating that the tree has not yet grown to include the nodes with numbers 6 and 7, hence no leaf parameters must be sampled for these nodes. But for the node with number 3 in the right tree, leaf parameters must be sampled instead.

Gaussian priors are imposed on both the feature weights of the leaf nodes and interior nodes. This is mainly motivated by the simplicity of working with a Normal distribution and the strong empirical performance found by Salazar (2024) using these priors.

In the VSRF model, the parameters of the prior distributions are not based on any initial beliefs, but are drawn randomly. For $\{\mu_{ti}\}_{i=1}^{2^{d_t-1}-1}$ and $\{\zeta_{ti}\}_{i=2^{d_t-1}}^{2^{d_t}-1}$ draws from a standard normal distribution are used. For S_{ti} and Z_{ti} the draws must be positive hence the exponent is taken of draws of a standard normal distribution. Finally, for the $\{\gamma_{ti}\}_{i=1}^{2^{d_t-1}-1}$, a logistic transformation is used on draws from a standard normal distribution, this is done to ensure that the values are on the interval between $[0, 1]$. In this work I do not explore other ways to initialize the parameters of the priors, but it is likely that in certain scenario's like for example high-dimensional data careful prior construction instead of random initialization can improve the VSRF model.

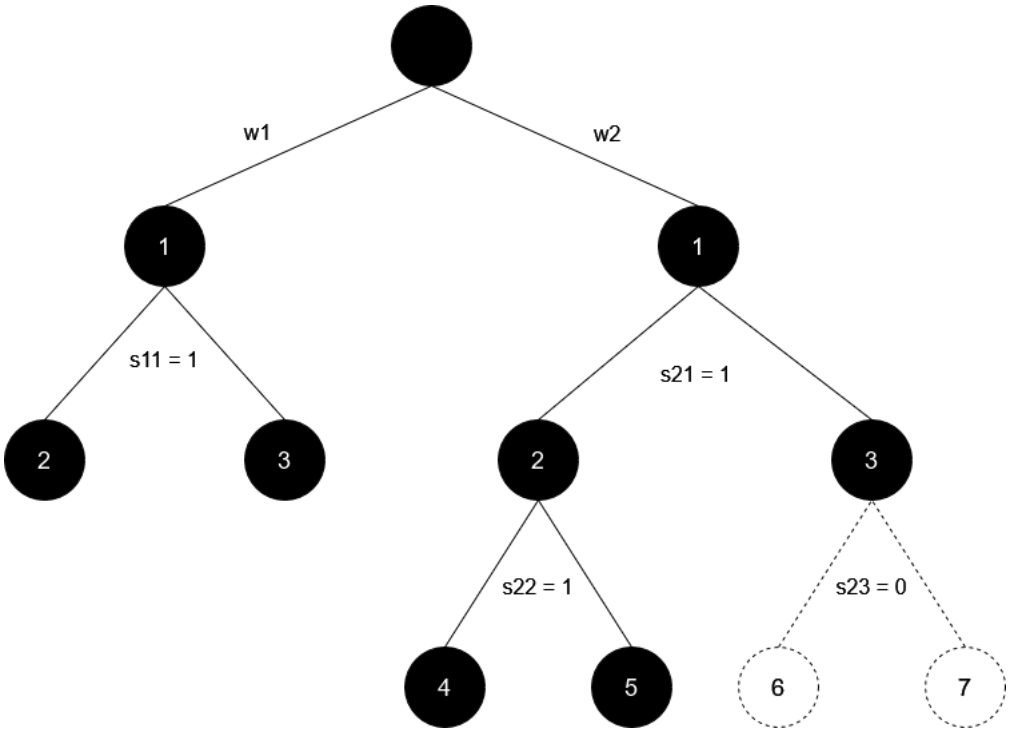

Figure 4.1: Illustration of the structure of a VSRF model with two trees. The left tree has one interior node and two leaf nodes. While the right tree has two interior nodes and three leaf nodes.

4.1.5 Bagging to reduce variance of the VSRF

I study the impact bagging has on the VSRF model. Bagging is introduced by using a different bootstrapped sample of the data for each individual regression tree within a VSRF. Afterwards, the predictions over all trees are averaged with the aim of decreasing the overall predictive variance of the VSRF model. The effectiveness of bagging has been demonstrated by earlier studies, for example in the Random Forest model (Breiman, 2001).

Another benefit of bagging in the context of the VSRF model is that it is likely to encourage structural variation between the trees that are growing in the VSRF. So the trees will have different shapes catered to their own bootstrapped sample. Intuitively if all trees in the VSRF model grow to be similar in structure, then there is little potential for variance reduction since the predictions are likely to be similar as well. I compare the performance of a VSRF model with bagging against a VSRF without bagging to assess whether there are any benefits.

4.2 Offline variational inference

The essential idea in variational inference is to use a surrogate posterior function that approximates the true posterior. Let $p(\Phi, \Psi)$ denote the prior of a VSRF model. Let $q(\Phi, \Psi)$ denote the surrogate posterior. I make use of the mean-field approximation and decompose the posterior into parts. For each part of the approximation, the surrogate distribution is chosen to be equivalent to the prior. The posterior parameters are distinguished from the prior parameters with a tilde $(\tilde{\gamma_{ti}}, \tilde{\mu_{ti}}, \tilde{\zeta_{ti}}, \tilde{S_{ti}}, \tilde{Z_{ti}})$. For convenience of notation, let all parameters of the surrogate posterior be denoted by the vector ω. To find the optimal surrogate posterior, the KL-divergence between the true and surrogate posterior will have to be minimized. However because the true posterior is not known, the true posterior is decomposed using Bayes' theorem. Which gives rise to the evidence lower bound (ELBO) specified as:

$$ELBO(\omega) = \mathbb{E}_{q_\omega}[log(p(y|X, \Phi, \Psi))] - KL(q_\omega(\Phi, \Psi)||p(\Phi, \Psi)) \tag{4.1}$$

The idea behind maximizing the ELBO, is that it is the same as minimizing the KL-divergence between the true and surrogate posterior. The only difference lies in the log-marginal likelihood of the data which is a constant value. Using the definition of the KL-divergence the ELBO can also be written as:

$$ELBO(\omega) = \mathbb{E}_{q_\omega}[log(p(y|X, \Phi, \Psi))] - \mathbb{E}_{q_\omega}[log(q_\omega(\Phi, \Psi))] + \mathbb{E}_{q_\omega}[log(p(\Phi, \Psi))] \tag{4.2}$$

The terms within the ELBO for a model with linear leaf nodes can be expressed as:

$$log(p(y|X, \Phi, \Psi)) = \sum_{j=1}^{B} log(p(y_j|x_j, \Phi, \Psi)) \tag{4.3}$$

$$\begin{aligned} log(p(\Phi, \Psi)) = &\sum_{t=1}^{T} \sum_{i \in N_t} log(\gamma_{ti}) + log(\phi(\beta_{ti}; \mu_{ti}, S_{ti})) \\ &+ \sum_{i \in L_t^q} log(1 - \gamma_{ti}) + \sum_{i \in L_t} log(\phi(\kappa_{ti}; \zeta_{ti}, Z_{ti})) \end{aligned} \tag{4.4}$$

$$\begin{aligned} log(q_\omega(\Phi, \Psi)) = &\sum_{t=1}^{T} \sum_{i \in N_t} log(\tilde{\gamma_{ti}}) + log(\phi(\beta_{ti}; \tilde{\mu_{ti}}, \tilde{S_{ti}})) \\ &+ \sum_{i \in L_t^q} log(1 - \tilde{\gamma_{ti}}) + \sum_{i \in L_t} log(\phi(\kappa_{ti}; \tilde{\zeta_{ti}}, \tilde{Z_{ti}})) \end{aligned} \tag{4.5}$$

Where $\phi(x; \mu, \Sigma)$ denotes the multivariate PDF of a normal distribution. Equation 4.3 displays the joint likelihood over B observations. The interpretation of Equation 4.4 and Equation 4.5 follows from the tree structure of a VSRF model. For each tree t, there is a set of interior nodes N_t, a set of leaf nodes not at the maximum depth L_t^q , and a set comprised of all leaf nodes L_t. Given these definitions it follows that $L_t^q \subset L_t$. For the interior nodes in a tree, there

must be child nodes, hence the contribution of splitting as a node is present in the equations. Furthermore the contribution of the parameters within the node with a Gaussian distribution are also present. For the leaf nodes not at the maximum level, the contribution of not splitting as a node is captured. Had these nodes split they would have been interior nodes. Finally for all leaf nodes, the contribution of the linear model within the leaf is accounted for.

Naturally since the sets (L_t, L_t^q, N_t) determine the structure of a variational tree they have to be expressed in terms of the node splitting indicators s_{ti} for optimization to take place. I make use of the method introduced by Salazar (2024), where the sums are expanded over all nodes and indicator variables based on products of the node-splitting indicators are used to only select those elements within a certain set.

The decision to use the node-splitting variables in this way makes it possible to keep the sizes of all parameters in the VSRF model fixed, instead of having dynamic parameter vectors. In the implementation of the VSRF, all parameters are initialized as if each tree will grow to reach its maximum depth. However, the node-splitting indicators determine whether a parameter has a contribution to the ELBO and hence whether it will receive gradient updates. Even though this choice requires the use of additional memory to store parameters that might never be part of the VSRF, it makes it easier to vectorize the implementation.

In order to optimize the ELBO, its gradient must be computed with respect to the parameters of the surrogate posterior. The gradient of the ELBO is obtained using Monte Carlo sampling. However, given that the regular Monte Carlo estimator has a high variance (Ranganath et al., 2014), I reparameterize the variables in the same way as Salazar (2024) to shrink the variance. The reparamterization essentially shifts the source of the randomness allowing the gradient and expectation to be exchanged. To sample from a Bernoulli distribution, the Softmax-Gumbel approximation (Jang, Gu & Poole, 2016) is used.

4.2.1 Reparameterizations

The reparameterization trick is often used in variational inference to reduce the variance of the Monte Carlo estimator. The idea is relatively simple. The goal is to express a variational parameter in such a way that the random element of the parameter is no longer internal to the parameter itself.

For a Gaussian distribution, its reparameterization follows directly from the properties of the Gaussian. In general let $\delta \sim \mathcal{N}(\mu, \Sigma)$, Denote the cholesky decomposition of Σ as $\Sigma^{0.5}(\Sigma^{0.5})^T$ and introduce a white noise variable $\epsilon \sim \mathcal{N}(0, I)$. Then the reparameterization of δ is given by:

$$\delta = \mu + \Sigma^{0.5}\epsilon$$

This reparameterization can be applied to the parameters of the interior nodes and the parameters of the leaf nodes which are drawn from a Gaussian distribution.

Using the work of Jang et al. (2016), Bernoulli random variables can be reparameterized using the Gumbel distribution. In general let $\delta \sim Bernoulli(\alpha)$, let g_1, g_2 denote 2 independent samples from a Gumbel distribution. Then δ can be expressed as

$$\delta = (1,0)^T one\_hot(argmax(g_1 + \log(\alpha), g_2 + \log(1-\alpha)))$$

Where the argmax function returns the argument of the largest element and the one_hot function applies a one-hot encoding. To make the expression differentiable, the one-hot encoded argmax operator is approximated using the limiting distribution of the softmax.

$$\delta = (1,0)^T \lim_{\tau \to 0^+} softmax(\frac{g_1 + \log(\alpha)}{\tau}, \frac{g_2 + \log(1-\alpha)}{\tau})$$

Where the τ parameter is slowly shrunk to zero over time during the estimation procedure. This reparamterization can be applied to the node splitting indicators s_{ti}.

4.2.2 Estimation algorithm for a VSRF

To maximize the ELBO, I use the Adam optimizer developed by (Kingma & Ba, 2014). The pseudocode for the estimation of a VSRF is given by Algorithm 1. It can be seen that every epoch

Algorithm 1 Estimation of VSRF model

for $i = 1, ..., D$ **do**
  Sample node splitting indicators for all trees.
  Sample leaf and interior node parameters.
  Compute ELBO(ω_i) for all trees using sampled parameters.
  Use automatic differentiation to obtain $\nabla_\omega ELBO(\omega_i)$.
  Compute ω_{i+1} using Adam optimizer given $\nabla_\omega ELBO(\omega_i)$ and ELBO(ω_i)
end for

starts by sampling the splitting indicators, the parameters of the leaf nodes, and the parameters of the interior nodes. Then the ELBO is approximated using the sampled parameters. The use of a single sample greatly reduces computational cost to approximate the ELBO, while there is minimal increase in variance relative to using multiple samples, given the reparamterization trick. Once the ELBO has been computed, the computation graph is used to perform automatic differentiation and obtain gradient estimates of the ELBO with respect to all parameters of the surrogate posterior. Finally the ELBO and gradient information are used by the Adam optimizer to update the parameters of the surrogate posterior. Preliminary experiments have shown that good a balance between convergence and estimation time, can be achieved for the VSRF when Algorithm 1 is given 500 epochs. Furthermore for the learning rates of the Adam optimizer, the same configuration as Salazar (2024) is used, initially the learning rate is chosen to be 0.1, and a decay is used until a minimum learning rate of $\frac{0.01}{D}$ is reached. This corresponds to larger gradient updates at the start of the optimization while gradually shrinking the magnitude of the updates at later iterations.

4.2.3 Generating predictions from a VSRF

Given a new observation x_{B+1}, a prediction from the VSRF can be generated. Let $D = \{(y_B, x_B), .., (y_1, x_1)\}$ be the set of all previously observed samples. Then the approximate posterior predictive distribution of y_{B+1} is given by:

$$p(y_{B+1}|x_{B+1}, D) \approx \sum_{t=1}^{T} w_t \int p(y_{B+1}|x_{B+1}, D, \Phi_t, \Psi_t) q_\omega(\Phi_t, \Psi_t)\, d(\Phi_t, \Psi_t) \tag{4.6}$$

The distribution is an approximation because the surrogate posterior $q_\omega(\Phi_t, \Psi_t)$ is used instead of the true posterior. When generating point predictions, the expected value of the posterior predictive distribution is used. Let $\Phi_t^{(i)}, \Psi_t^{(i)}$ be Monte Carlo draws from the surrogate posterior $q_\omega(\Phi_t, \Psi_t)$ for $i = 1, .., M$. Then the expected value can be approximated as follows:

$$\mathbb{E}_{q_\omega}[y_{B+1}|x_{B+1}] \approx \sum_{t=1}^{T} \frac{w_t}{M} \sum_{i=1}^{M} \sum_{l \in L_t} p(l|x_{B+1}, \Phi_t^{(i)}) f_l(x_{B+1}|\Psi_{tl}^{(i)}) \tag{4.7}$$

Where the functional form of $f_l(x_{B+1}|\Psi_{tl}^{(i)})$ depends on whether the leaf uses a spline or linear form.

4.3 Online variational inference

To derive the online Bayesian update for VSRF, the Sequential Variational Bayes (SVB) as detailed by Chérief-Abdellatif et al. (2019) is used. Let ω denote the vector of all parameters of the surrogate posterior of the VSRF. Suppose a stream of i observations has been observed in the past. When a new mini-batch of observations $\{i+1, i+2, ...i+k\}$ arrives, the prior is set to a previously estimated surrogate posterior $p(\Phi, \Psi) = q_{\omega_i}(\Phi, \Psi)$. For a single new observation the update rule is given by:

$$\omega_{i+1} = \underset{\omega \in \mathcal{M}}{\arg\max} \quad \mathbb{E}_{q_\omega}[log(p(y_i|x_i, \Phi, \Psi))] - KL(q_\omega(\Phi, \Psi)||q_{\omega_i}(\Phi, \Psi)) \tag{4.8}$$

To perform an update on a mini-batch, one could consider performing k repeated updates using Equation 4.8, but this has an additional cost in estimation time. Hence I make use of all observations in the mini-batch to get a noisy estimate of the log-likelihood. For a mini-batch of k observations, the expression for the updated parameters ω_{i+k} of the surrogate posterior is given by Equation 4.9.

$$\omega_{i+k} = \underset{\omega \in \mathcal{M}}{\arg\max} \sum_{j=i+1}^{i+k} \mathbb{E}_{q_\omega}[log(p(y_j|x_j, \Phi, \Psi))] - KL(q_\omega(\Phi, \Psi)||q_{\omega_i}(\Phi, \Psi)) \tag{4.9}$$

Where the set $\mathcal{M}$ is the set of all potential parameter values. Chérief-Abdellatif et al. (2019) suggest using a linear approximation $\omega^T \nabla_\omega \mathbb{E}_{q_{\omega_i}}[log(p(y_j|x_j, \Phi, \Psi))]$ for $\mathbb{E}_{q_\omega}[log(p(y_j|x_j, \Phi, \Psi))]$. This approximation is used to avoid the re-computation of gradients over past observations for the log-likelihood since it is evaluated using the previously obtained parameter values. However

in the expectation in the specification above the log-likelihood is only evaluated on a new mini-batch, and can be obtained for the VSRF relatively cheaply. Hence there is no need to use a linear approximation in this case.

The mini-batch update rule can be linked to the idea of stochastic gradient descent (Robbins & Monro, 1951). When stochastic gradient descent is used random draws of the data are used to approximate the gradient. However this randomness need not be introduced by a researcher. That is if the stream of incoming observations is assumed to be drawn at random from the data-generating process, then using only the newly observed samples might provide sufficient information for the online update. Mainly because the information provided by the previously observed observations is already captured by the updated prior.

4.4 Implementation

The VSRF model is written in Python using Pytorch (Paszke et al., 2019), by extending the implementation of a single variational regression tree provided by Salazar (2024). Pytorch has support for automatic differentiation and several optimization methods, this makes it a good fit for the VSRF. The VSRF implementation allows for estimation on GPUs with CUDA support, yielding large gains in computation time in particular for larger datasets.

To allow for easy adoption and facilitate comparisons with other models, the VSRF model is published as an open-source package on Pypi and Github. For more details on my implementation of the Variational Soft Random Forest see https://github.com/achasol/vsrf-model.

4.5 Application

In this section, the Scarce Promotion Problem is introduced. Let there be a retailer who observes a single customer per time period for a total of T periods. In each period, the retailer can choose one of K promotions. At each point in time the retailer observes a p-dimensional context $x_t \in \mathbb{R}^p$ with information about the customer. After observing this information the retailer chooses a promotion $a_t \in \{1, .., K\}$ to offer to the customer. I make use of Rubins causal model (Angrist et al., 1996) to represent the profit at time $t = 1, .., T$ for a given promotion $k \in 1, .., K$ as Y_{tk}. The event that the retailer does not make any promotion is fixed to $k = 1$. Then the causal effect commonly referred to as the uplift for promotion k is $Y_{tk} - Y_{t1}$.

I assume the stable unit treatment value assumption (SUTVA), unconfoundedness given the observed context x_t, and non-linear realizability of the profit of a promotion k which translates to:

$$\mathbb{E}[Y_{tk}|x_t, (x_{t-1}, a_{t-1}), ..., (x_1, a_1)] = h(x_t, k)$$

Where h is an unknown function to the retailer. The restriction is imposed that each promotion can only be used at most C_k times $\forall k \in \{2, .., K\}$. Then the scarce promotion problem can be formulated as:

$$
\begin{aligned}
&\text{Max}_{z_{tk}} \sum_{t=1}^{T}\sum_{k=1}^{K} h(x_t,k)\cdot z_{tk} \\
\text{subject to:}\quad &\sum_{k=1}^{K} z_{tk} = 1, \quad \forall t \in \{1,...,T\} \\
&\sum_{t=1}^{T} z_{tk} \leq C_k, \quad \forall k \in \{1,...,K\} \\
&z_{tk} \in \{0,1\}, \quad \forall t \in \{1,...,T\}, \forall k \in \{1,...,K\}
\end{aligned}
$$

Where the decision variable z_{tk} is equal to $\mathbb{I}[a_t = k]$, meaning a binary indicator that a given promotion k was chosen. Note that if the retailer was able to know which customers were going to visit ahead of time, then the above problem is equivalent to a multidimensional multiple-choice knapsack problem which belongs to the class of NP-complete problems.

However, in the online setting, an additional layer of complexity is added to the problem since the retailer does not know which customers will visit his shop. Hence my main focus is on finding a heuristic for the sequential problem with good performance. Where good performance is measured in terms of regret. To this end let $a_t^* = argmax_k h(x_t,k)$ denote the optimal promotion at time t.[1]

Then the regret at time t can be defined as:

$$r_t = h(x_t, a_t^*) - h(x_t, a_t)$$

The regret captures how much profit was lost by choosing a certain promotion compared to the optimal promotion. The aim of the sequential decision problem is to pick promotions such that profit is maximized or equivalently such that $R_T = \sum_{t=1}^{T} r_t$ is minimized and the scarcity constraints are satisfied.

A fundamental challenge faced by the retailer is to try and predict the profit given the context x_t and the chosen action a_t. In this work using the VSRF model to approximate the profit function h is proposed to try and predict the profit.

4.5.1 Thompson sampling with scarcity correction

Once the reward approximation model has been estimated, it can be used in a Thompson sampling (Thompson, 1933) algorithm. I slightly adjust the original Thompson sampling algorithm by introducing a scarcity correction specific to my constrained problem. To this end define the remaining capacity of a promotion k at time t to be $n_{tk} = C_k - \sum_{j=1}^{t} \mathbb{I}[a_t = k]$. Then define the relative capacity at time t to be $w_{tk} = \frac{n_{tk}}{\sum_{i=1}^{K} n_{ti}}$. Let ω denote the parameters of the surrogate posterior. Let the surrogate posterior of a VSRF be denoted by $q_\omega(\Phi, \Psi)$. Then the Thompson sampling algorithm with scarcity correction is shown in Algorithm 2.

It can be seen that whenever a customer arrives 100 draws are made from the VSRF model

[1]Note that this quantity cannot be obtained by the retailer since only a single promotion can be evaluated, but it is known in the context of a simulation.

Algorithm 2 Thompson sampling with scarcity correction

for $t = 1, ..., T$ **do**
  for $k = 1, ..., K$ **do**
    for $i = 1, ..., 100$ **do**
      Sample $\Phi^{(i)}, \Psi^{(i)}$ from $q_\omega(\Phi, \Psi)$
      Compute predicted reward $\hat{h}^{(i)}(X_t, k)$ using $\Phi^{(i)}, \Psi^{(i)}$
    end for
    Uniformly draw one prediction $\hat{\theta}_{tk}$ from $[\hat{h}^{(1)}(x_t, k), ..., \hat{h}^{(100)}(x_t, k)]$
  end for
  Choose promotion $a_t = \arg\max_k w_{tk}\hat{\theta}_{tk}$
  Observe Y_{t,a_t}
  Update $q_\omega(\Phi, \Psi)$ using (Y_{t,a_t}, x_t, a_t)
end for

for the predicted rewards of all promotions. Then for each promotion a random draw of the predicted reward is chosen. Afterwards a scarcity correction is performed on the reward. And the promotion is chosen with the highest corrected reward.

The idea behind the scarcity correction is that by biasing the reward estimates slightly to take the remaining capacity of the promotion relative to the other promotions into account, the performance of the algorithm can be improved. After the promotion is shown to the customer and the true reward is observed, the VSRF model is updated. The model is updated in a mini-batch fashion, given that a single sample has negligible impact on the posterior. The elegance of the Thompson sampler is that it deals with the exploitation-exploration trade-off in an intuitive way. That is by taking draws from the posterior of the VSRF there is always some non-zero probability for the VSRF to pick a promotion that does not have the highest expected value.

4.6 Evaluation

4.6.1 Evaluation metric

I have decided to utilize the Root Mean Squared Error (RMSE) to evaluate the performance of the VSRF model. The main reason is that this metric is widely used, and it allows for easy comparison with the earlier work by Salazar (2024), who also makes use of the RMSE. Let y_i be the true outcome variable, and let $\hat{y}_i$ denote some estimate of y_i for $i = 1, ..., N$. Then the RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.10}$$

4.6.2 Predictive performance of the VSRF

Predictive capability of VSRF

To evaluate the predictive capabilities of the VSRF model, a comparison is performed with BART (Chipman et al., 2010), RandomForest (Breiman, 2001), Catboost (Dorogush et al., 2018), and Xgboost (Chen et al., 2015), which are well-known decision tree ensemble methods

with strong predictive performance. Out-of-sample RMSE values are obtained from 5-fold cross-validation on a variety of datasets from the UCI machine learning repository (Asuncion & Newman, 2007).

4.6.3 Friedman's five dimensional test function

To evaluate how the VSRF model performs in a scenario with high-dimensional data with only a small number of relevant covariates, the simulation scheme introduced by Friedman (1991) is used. The scheme starts by making n independent draws of p uniform variables:

$$x_1, ..., x_p \sim Uniform(0, 1)$$

Using the draws, the target variable y is computed as follows:

$$y = 10sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon$$

Where $\epsilon \sim \mathcal{N}(0, 1)$. As can be seen by the specification of the target variable y it only depends on the first five covariates, all other covariates are not relevant. This specification of the target makes it possible to study how the VSRF performs when only a small subset of the predictors is relevant. Similarly to Chipman et al. (2010), a simulation is performed using $n = 100$ and varying the number of covariates $p \in \{10, 100, 1000\}$. Additionally another simulation is performed using $n = 1000$ while varying the covariates in the same way, to see the impact of increasing the number of observations. Performance is evaluated using out-of-sample RMSE scores that are averaged over five folds for several variants of the VSRF and a VART model.

Quality of the online variational approximation

To evaluate the quality of the online Bayesian update, 5-fold cross-validation is used on four datasets. For each fold, the VSRF is estimated in an offline fashion on the entire estimation sample, and the VSRF is also estimated in a streaming fashion on the estimation sample. The performance of both estimation procedures is compared using the RMSE computed on the holdout sample. The procedure is repeated for several datasets.

4.6.4 Application to Scarce Promotion Problem

Simulation based on Hillstrom

My simulation based on the Hillstrom dataset (Hillstrom, 2008) has three main components. First, the arrival process determines which customers arrive at the shop. Second the purchase decision, which determines whether a customer will make a purchase or not, and third the reward-generating process which determines the latent reward/profit for all possible promotions that the retailer can offer.

In the simulation, customers are shown one of two promotions or no promotion at all, indicated by $a_t \in \{1, 2, 3\}$. Where $a_t = 1$ corresponds to the case where no promotion is shown.[2]

[2]For convenience of notation without loss of any generality, I assume that each customer arrives at a distinct point in time t.

For the arrival process, a uniformly random draw is made of a customer from the customers present in the Hillstrom dataset.

To model the purchase decision, a draw from a Bernoulli random variable is used with success probability $\delta_0 + \delta_1 \mathbb{I}[a_t = 2] + \delta_2 \mathbb{I}[a_t = 3]$. Where δ_0 is the baseline purchase probability, and δ_1, δ_2 impact the purchase probability when a promotion is shown. The assumption is imposed that $\delta_0, \delta_1, \delta_2 \geq 0$. Which means that that showing a promotion never has an adverse effect on the baseline purchase probability. At the start of the simulation, a draw is made for the values of $\delta_0, \delta_1, \delta_2 \sim Dirichlet(1, 1.2, 1.6)$. The specification chosen for the Dirichlet distribution allows for an increase in purchase probability when a promotion is shown. Where the second promotion with a parameter value of 1.6 is slightly more effective compared to the first promotion with a value of 1.2.

When the purchase decision is negative, a consumer does not make a purchase and hence a reward of zero is observed. When a consumer does make a purchase The reward of each promotion Y_{tk} is simulated using two forms. The first form is a linear model given by Equation 4.11.

$$Y_{tk} = \theta_0^T \tilde{X}_t + \epsilon_t \tag{4.11}$$

The second form is a non-linear model given by Equation 4.12.

$$Y_{tk} = \sum_{i=1}^{p+2} 2cos(\theta_{0,i} \tilde{X}_{ti}) + \theta_{1,i} \tilde{X}_{ti} + \theta_{2,i} \tilde{X}_{ti}^2 + \epsilon_t \tag{4.12}$$

Where $\tilde{X}_t = (X_t, \mathbb{I}[k = 2], \mathbb{I}[k = 3])$ and $\epsilon_t \sim \mathcal{N}(0, 10)$ is a noise term. In all simulation runs for both the linear and non-linear forms, the initial values of $\theta_0, \theta_1, \theta_2$ are drawn from a half-normal distribution with mean zero and a variance of one. The decision to use both a linear and non-linear model is to evaluate how the Thompson sampler with the VSRF performs in both settings.

Benchmark algorithms

The performance of the scarcity-corrected Thompson sampler using the VSRF is compared against several benchmarks. The first benchmark is the Naive benchmark, it never offers a promotion to any customer that arrives. The second benchmark is the Random benchmark, it chooses a random promotion that still has capacity left and offers it to a customer. The third benchmark is an implementation of the Epsilon-Greedy algorithm (Sutton & Barto, 2018). The Epsilon-Greedy algorithm creates an internal representation of the reward of each promotion and then recommends the promotion with the highest reward with a probability $1 - \epsilon$, otherwise with probability ϵ it chooses a random promotion. The final benchmark model is a Thompson sampler using draws from the posterior of a VSRF without scarcity correction, so that the impact of performing a scarcity correction can be analyzed.

Chapter 5

Results

5.1 Predictive performance of VSRF

In this section, the predictive performance of the Variational Soft Random Forest is evaluated. First a comparison is performed on eight datasets, against several state-of-the-art decision tree ensemble methods. Afterward, a deep dive into the different parameter choices of the VSRF is performed using four evaluation datasets. Then the performance of the VSRF on high-dimensional data is studied using Friedman's five dimensional test function. All experiments were performed on a machine with 32 GB of RAM, an Intel Core I9 processor, and a Nvidia Geforce RTX 4070 GPU.

5.1.1 General performance

In Table 5.1 the predictive performance of the VSRF is compared against XGBoost (Chen et al., 2015), BART (Chipman et al., 2010), RandomForest (Breiman, 2001), Catboost (Dorogush et al., 2018) and a VART (Salazar, 2024). The RMSE values in the table are averaged over five folds of each dataset to increase the robustness of the findings. The lowest RMSE value within a row is made bold.

All benchmark models are evaluated with the default parameters and fifty regression trees. For the VART and VSRF models, two trees with a maximum tree depth of three are used. It is evident from Table 5.1, that on average the Catboost model has the lowest RMSE values, given the use of default parameters. This finding is likely caused by the wide adoption of Catboost, which means that the default hyperparameters are chosen in such a way that they have strong out-of-the-box performance.

For some datasets, both the VART and VSRF with linear leaf nodes seem to have relatively high RMSE values compared to the non-variational benchmarks. Where the spline variant of the VSRF appears to be most competitive with the other benchmark models considered. A potential explanation could be that the linear VSRF and VART simply lack the flexibility to capture the relationship observed in some datasets.

Notably for the forest dataset where all models have relatively high RMSE values, the VART and BART models achieve the lowest RMSE amongst all models. The linear VSRF model also performs well on the forest dataset. On the other hand for the energy dataset, the performance

of the VART and VSRF is relatively bad compared to XGBoost and CatBoost.

When comparing the performance of the VSRF models relative to the VART using Table 5.1, the results are not clear as to the added value of the VSRF model relative to a single VART. For most datasets, there seems to be little variation in RMSE values. Except for an outlier of the airfoil dataset, where the spline variant of the VSRF achieves almost half the RMSE of the VART model. In the subsequent section the added value of the VSRF relative to the VART will be further explored.

Table 5.1: Out-of-sample RMSE values for several benchmark models and a linear and spline variant of the VSRF.

Dataset	**XGBoost**	**BART**	**RandomForest**	**CatBoost**
airfoil	**0.042**	0.127	0.046	0.05
autompg	0.079	0.088	0.071	**0.068**
concrete	0.049	0.103	0.056	**0.047**
energy	**0.009**	0.081	0.013	**0.009**
forest	0.22	**0.193**	0.198	0.215
housing	0.065	0.1	0.062	**0.057**
solar	0.103	0.1	0.1	0.099
stock	0.044	0.081	0.042	0.045
Dataset	**VART** $(d = 3)$	**spline-VSRF**$(t = 2, d = 3)$	**linear-VSRF** $(t = 2, d = 3)$	
airfoil	0.251	0.138	0.268	
autompg	0.106	0.083	0.109	
concrete	0.126	0.133	0.13	
energy	0.091	0.127	0.086	
forest	**0.193**	0.215	0.196	
housing	0.105	0.102	0.112	
solar	**0.092**	0.095	0.093	
stock	0.04	0.092	**0.039**	

5.1.2 A single tree or a forest?

In this section, a single VART model is compared against two variants of the VSRF to see whether the VSRF is able to improve over a single VART. In Figure 5.1, four boxplots are displayed for 4 selected datasets. These datasets will be used during the evaluation in all upcoming sections and were selected based on the RMSE values in Table 5.1. In particular, for the forest and stock datasets, the VSRF variants had relatively low RMSE values, while for the energy and solar dataset RMSE was relatively high. The goal of selecting these four datasets is to provide a nuanced picture of the predictive performance of the VSRF.

Each candle in the boxplots of Figure 5.1 corresponds to a model and displays the distribution of 100 RMSE values generated and averaged over 5 folds. For each dataset the VART, VSRF with linear leaf nodes (VSRF-linear), and the VSRF with splines in the leaf nodes (VSRF-spline) are compared. The maximum tree depth of all models is chosen to be three, and the number of trees in the VSRF is chosen to be two.

The results of 5.1 show a more nuanced picture of the effect of the VSRF relative to Table 5.1. It can be seen that by introducing splines into the VSRF the variance of the predictions is increased significantly for all datasets. This finding might be caused by the large increase in parameters relative to the linear variant of the VSRF.

For the energy dataset, one can observe that the linear VSRF model achieves lower RMSE values compared to a single VART. For the forest and solar datasets, the RMSE distribution of

the linear VSRF model is rather similar to that of a single VART. Only for the stock dataset the 75th quantile of the observed RMSE values is higher for the linear and spline VSRF. This finding illustrates that for the stock dataset, making the model more flexible comes at a large cost in an increased variance of the predictions.

It is clear from the evaluation of Figure 5.1 that the VSRF can improve on a single VART, However choosing a VSRF model over a single VART is not always a beneficial decision. Depending on the problem and dataset at hand the models need to be evaluated to determine which model can achieve the best predictive performance. In the next section, an analysis is performed to assess the impact of increasing the number of trees in a VSRF model.

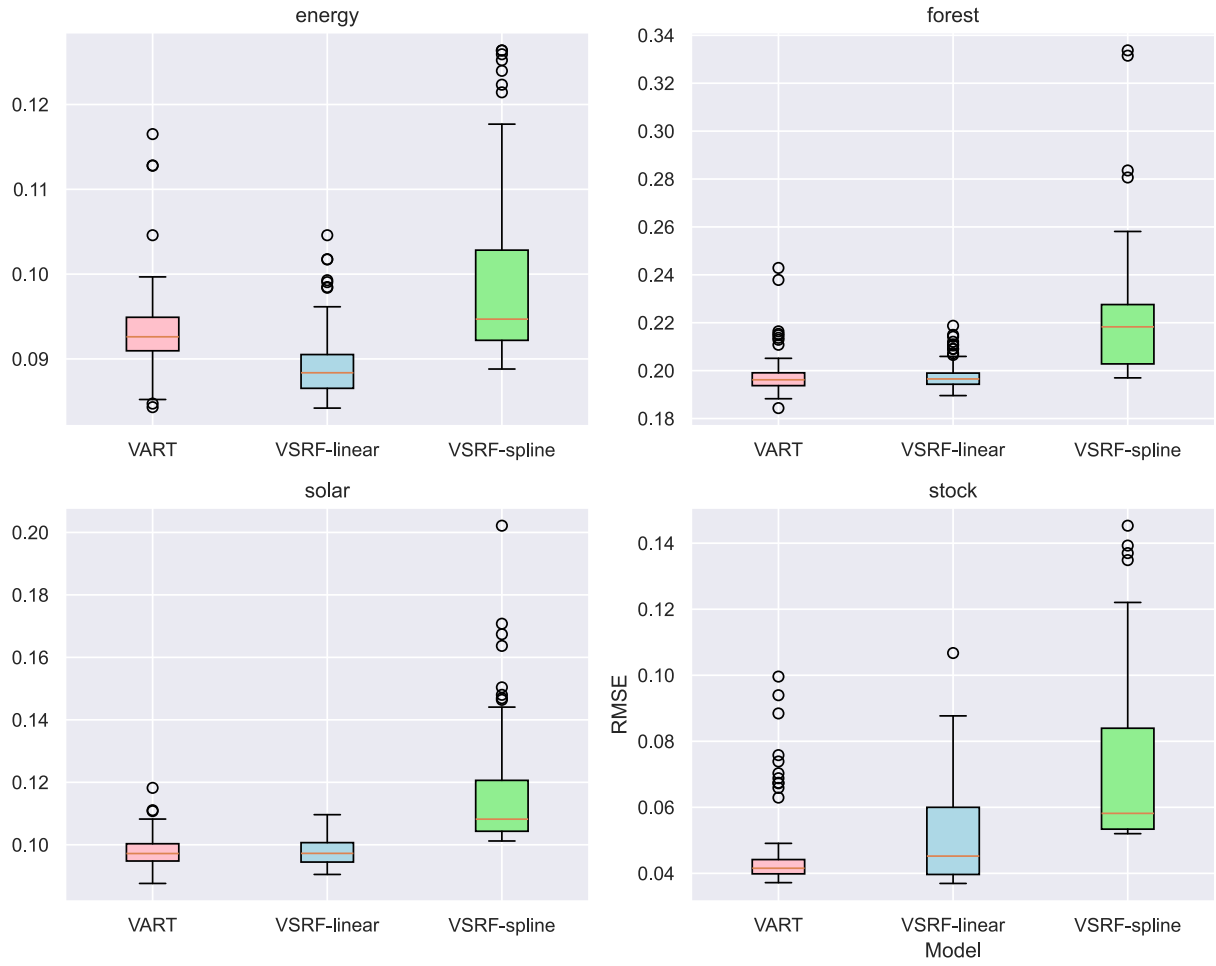

Figure 5.1: Boxplots of the RMSE of a VART, a linear VSRF and a spline VSRF for four different evaluation datasets.

5.1.3 Dense or shallow forest?

In this section, the impact of increasing the number of trees within a VSRF model is analyzed. The maximum depth of the trees is kept constant, and only the number of trees included in the forest is changed. In Figure 5.2, four models can be seen evaluated on the four chosen evaluation datasets. The first model is a single regression tree (VART), the second model is a VSRF with 5 trees (VSRF-5), the third model is a VSRF with 10 trees (VSRF-10) and the last

model is a VSRF with 25 trees. All VSRF models in the figure use linear leaf nodes, no bagging, inverse-depth weighting, and have a maximum tree depth of four.

As can be seen in Figure 5.2, there is a clear reduction in the variance of the RMSE values when using a forest model over a single tree. However, it does seem to be the case that the RMSE distribution is shifted upwards when 25 trees are used. It is likely that this is caused by the fact that the model with 25 trees has too many parameters to estimate relative to the size of the datasets considered. However the upward shift is not as pronounced for the solar and forest datasets.

The reduction in variance seems to be most clearly visible for the forest and solar datasets. For these datasets there also seems to be a consistent downward shift in the distribution of RMSE values. For the energy dataset, there is a smaller reduction visible and for the stock dataset, there does not seem to be a reduction at all. The reason that there is little to no variance reduction for the stock dataset is likely because the variance of a single VART model is already quite low.

Although when comparing the stock dataset in Figure 5.2 with the stock dataset in Figure 5.1, the variance of the linear VSRF variant seems to have decreased, indicating that using a forest with five trees instead of two provides better predictions. This pattern persists as a forest with 10 trees has a similar median RMSE when compared to a VART, but the observed RMSE values are much more concentrated as seen by the lower interquantile range in Figure 5.2.

5.1.4 Linear or spline leafs?

In this section, the decision to use linear leaf nodes or spline leaf nodes is analyzed. Four boxplots for four different datasets are used. The models evaluated in Figure 5.3, are a linear VSRF model (VSRF), a VSRF model with a single knot in each spline in the leaf nodes (VSRF-spline-1), and a VSRF model with 2 knots in the splines of the leaf nodes (VSRF-spline-2). The maximum tree depth is set to three and the VSRF models have two trees.

As can be seen in Figure 5.3, the VSRF model with linear leaf nodes consistently achieves the lowest variance in RMSE. This can be explained by the lower flexibility of the linear VSRF model compared to the spline variants, and hence the variance component of the RMSE is likely to be smaller. The RMSE for a spline model could still be lower if the bias decreases more relative to the variance increase, but this is not observed in Figure 5.3.

For the energy dataset specifically an interesting pattern is observed where the spline with a single knot performs much worse compared to the linear VSRF and the spline VSRF with two knots. This finding suggests that the choice of the number of knots can have a large impact on the performance of the spline variant, which indicates that tuning might be desirable.

For the other datasets, a consistent increase in the variance is observed across all spline variants. Notably, the maximum observed RMSE values for the spline variants of the VSRF appear to be much higher, which is not a desirable property.

Looking at the results of all datasets it appears that by choosing to use spline nodes the variance of the predictions is increased significantly. For the datasets considered there does not appear to be a sufficiently large enough reduction in RMSE values to warrant using splines in leaf nodes. Future research is desirable to determine whether these patterns generalize to other

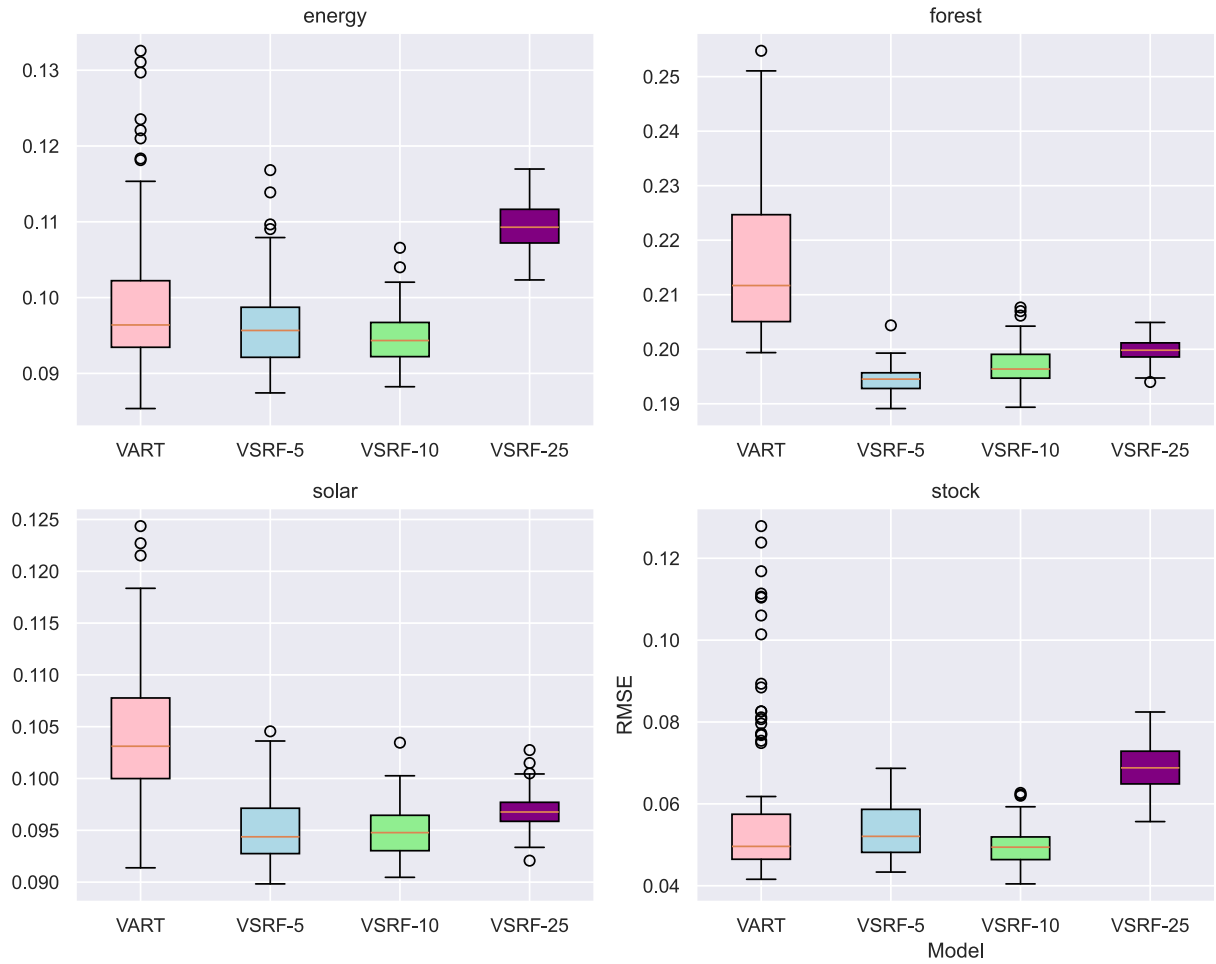

Figure 5.2: Boxplots of the RMSE of a VART and three VSRF models with increasing number of trees, for four different evaluation datasets.

datasets and parameter configurations of the VSRF model.

5.1.5 To bag or not to bag?

In this section, the impact of bagging is studied for the VSRF model. When bagging is used each regression tree within the VSRF model is estimated on a bootstrapped sample of the data. In Figure 5.4, four datasets can be observed with four different linear VSRF models. A bagged VSRF model with two trees (bag-VSRF-2), a VSRF model without bagging with two trees (VSRF-2), a bagged VSRF model with ten trees (bag-VSRF-10), and a VSRF model without bagging with ten trees (VSRF-10).

In general, it can be seen that with or without bagging, increasing the number of trees in the VSRF model decreases the variance of the observed RMSE values, which matches the findings observed in Figure 5.2. This finding is especially pronounced in the forest dataset, and is common to ensemble methods in general, as the number of learners is increased, the individual contribution of a single learner to the final prediction becomes smaller. This tends to have a positive effect on the variance of the predictions.

The impact of bagging on the VSRF seems to differ per dataset. For the forest dataset,

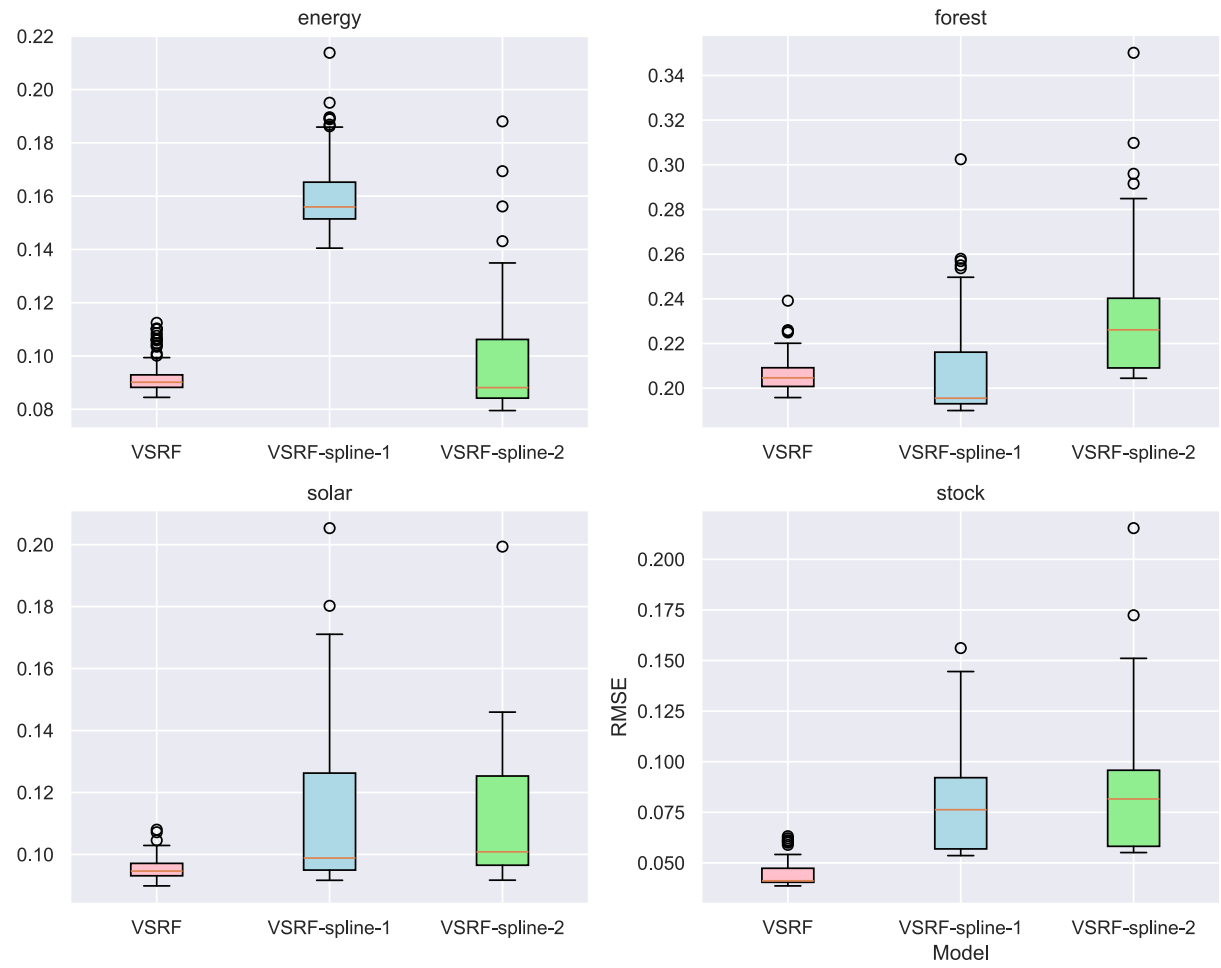

Figure 5.3: Boxplots of the RMSE of a linear VSRF model, a spline VSRF with one knot, and a spline VSRF with two knots for four evaluation datasets.

bagging reduces the overall RMSE, but for the solar dataset, bagging increases the RMSE in the VSRF variant with two trees. For the stock dataset, bagging seems to have a positive effect in reducing the maximum RMSE values observed, while for the energy dataset, bagging does not seem to have affected the RMSE distribution. Based on the observed results, I conclude that bagging is worth exploring for some datasets, and can provide benefits in some instances.

5.1.6 Uniform or inverse-depth weighting?

In this section, two weighting schemes for the VSRF are evaluated. The evaluation takes place on the same four datasets as the previous subsections. In Figure 5.5 four different models are considered, a linear VSRF model with inverse depth and uniform weighting (VSRF-id & VSRF-u), a VSRF with spline nodes with both inverse-depth and uniform weighting (VSRF-spline-id & VSRF-spline-u). Each variant of the VSRF has a maximum tree depth of three and is comprised of two trees.

The difference between the weighting schemes appears to be rather subtle in Figure 5.5. For some datasets like the energy dataset, the uniform weighting method seems to outperform the inverse depth weighting method slightly. For the stock dataset, however, the inverse depth

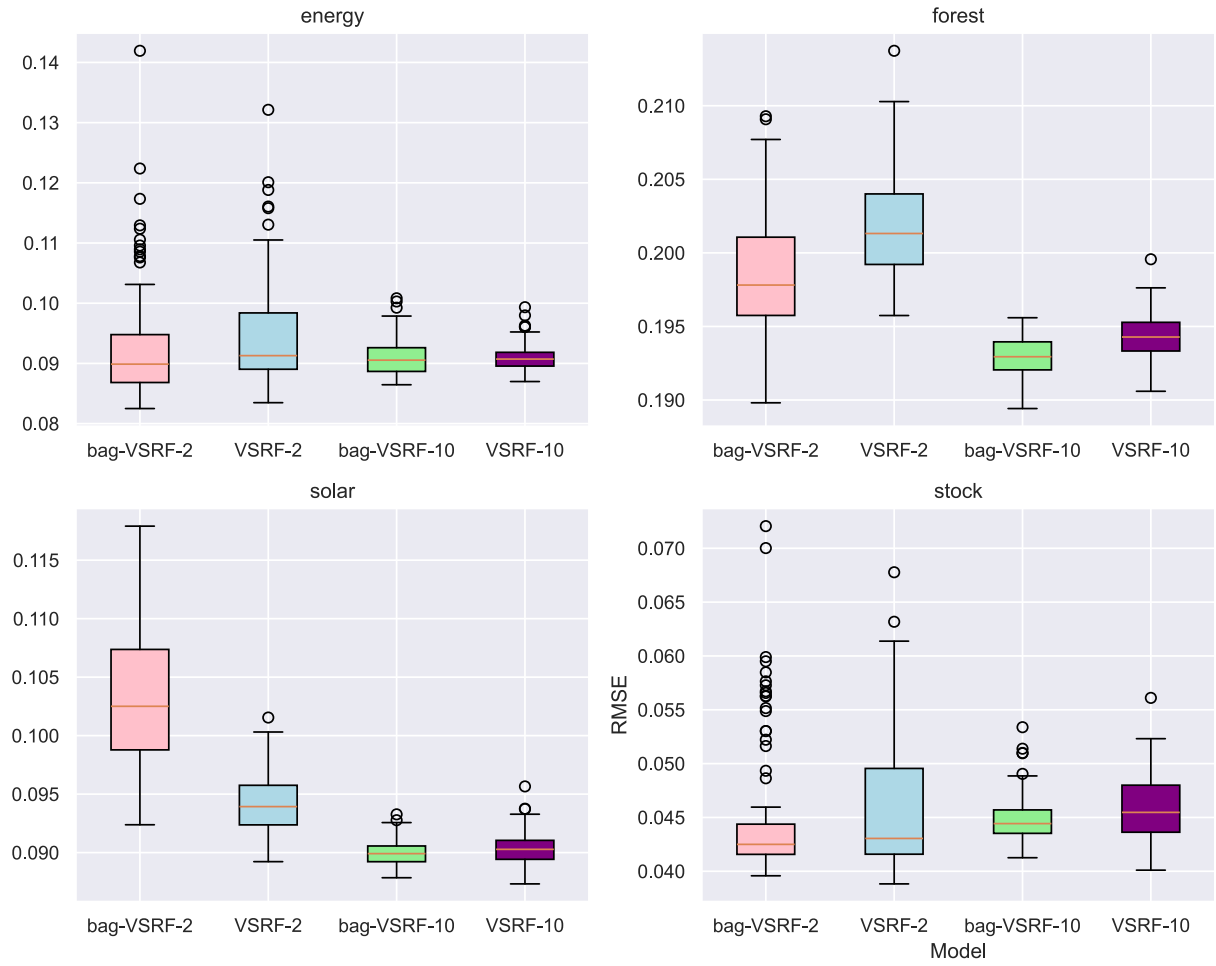

Figure 5.4: Boxplots of the RMSE of two VSRF models, with and without bagging for four evaluation datasets.

weighting method seems to have considerably lower median RMSE for the spline variant.

The choice of weighting function does not seem to alter the upper quantiles of the RMSE distribution observed in Figure 5.5 much. This is contrary to my expectation as I expected the dispersion of the predictions to decrease when using inverse-depth weighting. A possible explanation for this finding could be that the relationship between the dispersion of the predictions and the depth of a tree are not well approximated by a linear form in which case a different specification of the weighting function is more appropriate.

Based on the observed results I conclude that there are minor differences in performance when adopting either inverse depth or uniform weighting on the evaluation datasets. Hence the decision can be made to adopt a single weighting scheme or explore which weighting scheme suits a particular use-case best.

5.1.7 Overall impression

When combining all findings discussed above, the picture of the VSRF emerges as a model that can improve over a single variational regression tree for some datasets. However, the model must be properly evaluated, and if possible hyperparameter tuning should take place to extract

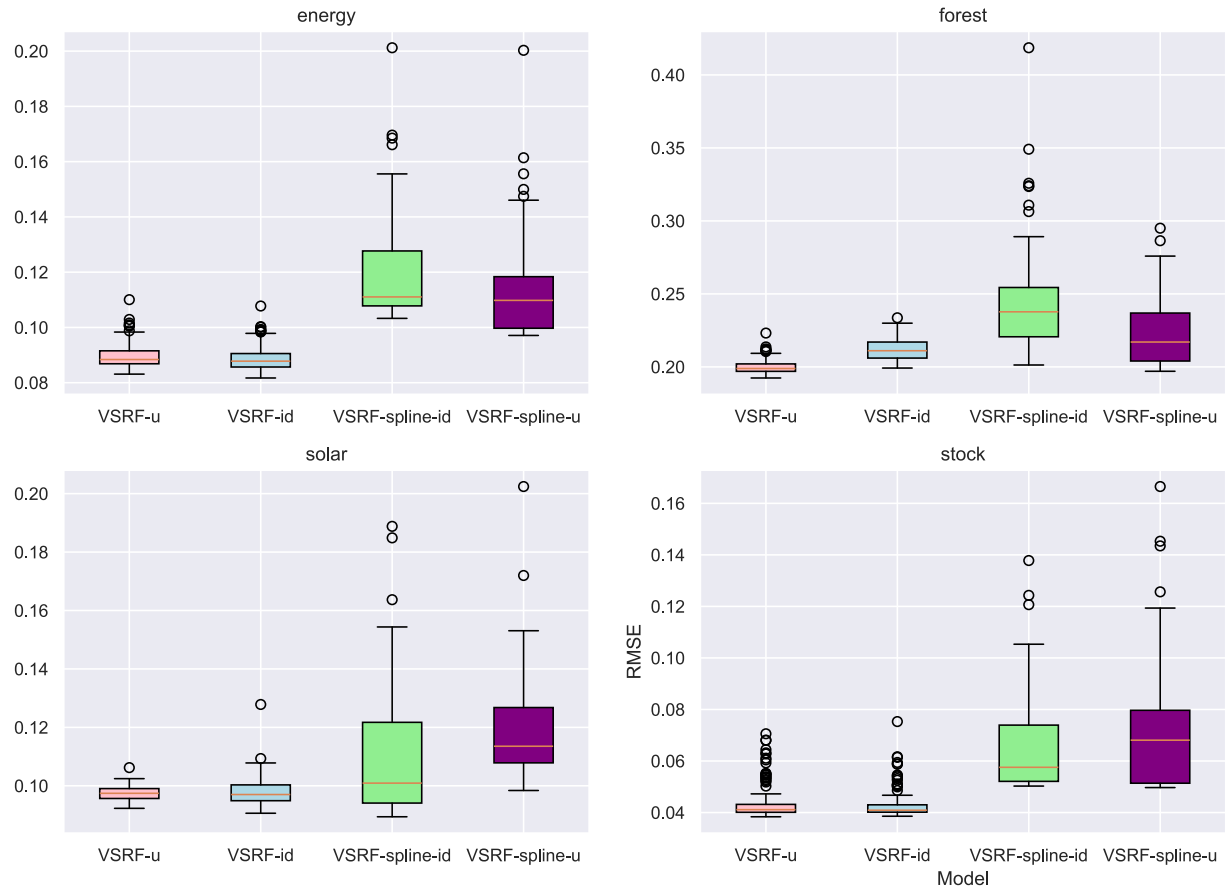

Figure 5.5: Boxplots of the RMSE of a linear and spline VSRF model, with uniform and inverse-depth weighting for four evaluation datasets.

maximum benefit. Including more trees in the VSRF model seems to reduce overall predictive variance but does come at a cost in required computation time. Using spline nodes in the VSRF has a negative influence on the variance of the predictions in many cases and hence the decision must be made carefully. Bagging can have a positive influence on the predictive performance of the VSRF. There did not seem to be much impact from the different weighting schemes on the VSRF suggesting that both options are viable.

It is important to realize that the findings above are meant to illustrate the trade-offs that the VSRF model faces. There is no guarantee that specific findings generalize to other datasets. Hence only empirical evaluation and proper hyperparameter tuning can give a conclusive answer which configurations of the VSRF are desirable for a specific use case.

5.2 Friedman's five dimensional test function

In this section, the performance of the Variational Soft Random Forest is evaluated in settings with high dimensional data. In Table 5.2 and Table 5.3, the out-of-sample quantiles of the RMSE can be seen, using data generated from Friedman's five-dimensional test function.

Four different models are evaluated. Which are a single variational regression tree (VART), a VSRF model with linear leaf nodes and two trees, a VSRF model with linear leaf nodes and ten trees and a VSRF model with spline leaf nodes and four trees. All models have a maximum tree depth of four. Looking at the results for $p = 100$ and $p = 1000$ covariates in Table 5.2, one can observe that both the VSRF model and VART model have very poor out-of-sample performance, given that the target variable has been normalized to the unit range. When looking at Table 5.3 it is evident that increasing the number of observations decreases the RMSE, but performance remains very poor, in particular for $p = 1000$.

The bad performance in high-dimensional settings of the VSRF can be explained by looking at a single VART model. I think there are several factors as to why the VART model breaks down in the case of high-dimensional data. The first is that for each interior node, it tries to find an oblique split over all covariates. Geometrically trying to find an oblique split is like trying to find a hyperplane that partitions the covariate space effectively, because the volume of the covariate space grows exponentially in the number of parameters, the samples become sparse. This sparsity complicates the process of finding a good split and is a clear illustration of the curse of dimensionality. Another factor is the large difference between the parameters and available observations, a single node in a VART already has at least p parameters, and given that only a small number of samples is used ($n = 100, 1000$), very strong regularization would be required on the node parameters to make estimation feasible.

An interesting finding in Table 5.2 and Table 5.3, is that the linear variants of the VSRF are able to improve over a single VART model in the case of high-dimensional data. For example, the median RMSE appears to be roughly 3x times smaller for the linear VSRF compared to a single VART when $p = 1000$. However, this finding is not observed for the spline variant of the VSRF which performs worse relative to a single VART.

In the case of $p = 10$, the results seem to align with the experiments on the UCI datasets in Table 5.1. The only notable difference is that in Table 5.2, the spline variant of the VSRF is the worst-performing model when $n = 100$, while it can be seen in Table 5.3, the spline becomes the best performing model when $n = 1000$. This finding is in stark contrast with the performance of the spline variant of the VSRF when $p = 1000$, where the performance seems to deteriorate when the number of observations is increased.

Overall, I conclude given the performance observed in Table 5.2 and Table 5.3, that both the VSRF and VART model are not suitable for usage in high-dimensional settings. Although having a large number of observations is likely to improve performance of some variants of the VSRF. The most promising avenue to explore to realize improvements while keeping the sample size fixed, would be to adjust the oblique splits to take a random subset of the covariates instead of all covariates, similar to the random covariate selection used when finding a split in a RandomForest (Breiman, 2001).

Table 5.2: Out-of-sample (25%, 50%, 75%) quantiles of the RMSE values for $p = 10$, $p = 100$, $p = 1000$ covariates of a VART and three VSRF variants evaluated on the Friedman simulations using $n = 100$ observations.

Model	**p=10**	**p=100**	**p=1000**
VART($d = 4$)	(0.16, 0.17, 0.2)	(2.79, 3.16, 3.61)	(63.04, 67.78, 72.7)
Linear-VSRF($t = 2, d = 4$)	(0.17, 0.17, 0.18)	(1.24, 1.38, 1.64)	(19.06, 21.74, 24.38)
Linear-VSRF($t = 10, d = 4$)	(0.19, 0.2, 0.2)	(2.24, 2.33, 2.45)	(29.36, 30.89, 32.18)
Spline-VSRF($t = 4, d = 4$)	(0.29, 0.31, 0.35)	(6.31, 7.04, 7.67)	(89.88, 96.17, 102.78)

Table 5.3: Out-of-sample (25%, 50%, 75%) quantiles of the RMSE values for $p = 10$, $p = 100$, $p = 1000$ covariates of a VART and three VSRF variants evaluated on the Friedman simulations using $n = 1000$ observations.

Model	**p=10**	**p=100**	**p=1000**
VART($d = 4$)	(0.1, 0.1, 0.11)	(0.44, 0.62, 0.87)	(70.98, 74.49, 78.67)
Linear-VSRF($t = 2, d = 4$)	(0.1, 0.1, 0.1)	(0.23, 0.30, 0.41)	(26.00, 29.00, 31.53)
Linear-VSRF($t = 10, d = 4$)	(0.11, 0.11, 0.11)	(0.95, 1.04, 1.11)	(22.47, 24.34, 25.90)
Spline-VSRF($t = 4, d = 4$)	(0.07, 0.08, 0.1)	(3.06, 3.4, 3.87)	(153.77, 159.58, 165.61)

5.3 Online variational approximation

The results of the online estimation procedure relative to an offline estimation procedure on the four benchmark datasets can be seen in Figure 5.6. The graphs for the regular offline estimation model are formed by re-estimating the model on a growing portion of the estimation sample in steps of 10%. After each re-estimation step the model is evaluated on a 10% hold-out sample to obtain an out-of-sample RMSE. This process is repeated over five different folds and the estimation samples are randomly shuffled. The averaged RMSE values are displayed by the blue lines in Figure 5.6. For the streaming algorithm, a posterior update is performed using the new observations without full re-estimation of the model. Similarly to the regular estimation procedure, a hold-out sample is used to obtain an RMSE value. The RMSE values are averaged over five folds and are denoted by the orange lines in Figure 5.6.

As can be seen in Figure 5.6, the quality of the streaming estimation procedure tends to differ per dataset. Surprisingly for the solar and forest datasets, the streaming procedure yields an improvement over offline estimation. However, for the stock and energy dataset, the reverse is observed as there is a loss in predictive ability by estimating the model in a streaming fashion.

The findings above combined suggest that the quality of the online estimation procedure is highly dependent on the underlying dataset. A possible explanation could be the variation in the underlying observations, causing the mini-batch approximation of the log-likelihood to no longer be representative of the holdout sample.

Another important point to mention is that from Figure 5.6 it can be seen that repeated estimation in a streaming way does not seem to accumulate approximation error. On the contrary, the difference between the blue and orange lines seems to decrease over time, indicating that consecutive streaming updates over a longer time horizon do not appear to negatively affect predictive power. For the energy, forest and solar datasets there does not appear to be a meaningfull difference in RMSE once the entire estimation sample has been observed.

The key benefit of the online variational approximation is that relative to the offline estimation algorithm the total computation time is decreased by a factor of 11. This reduction is sufficiently large to make the online variational approximation useful in practice. For some datasets a reduction in RMSE can also be achieved. In conclusion, it is clear that the quality of the online Bayesian update depends on the dataset considered and for some datasets requires a trade-off to be made between faster estimation and overall predictive power of the model.

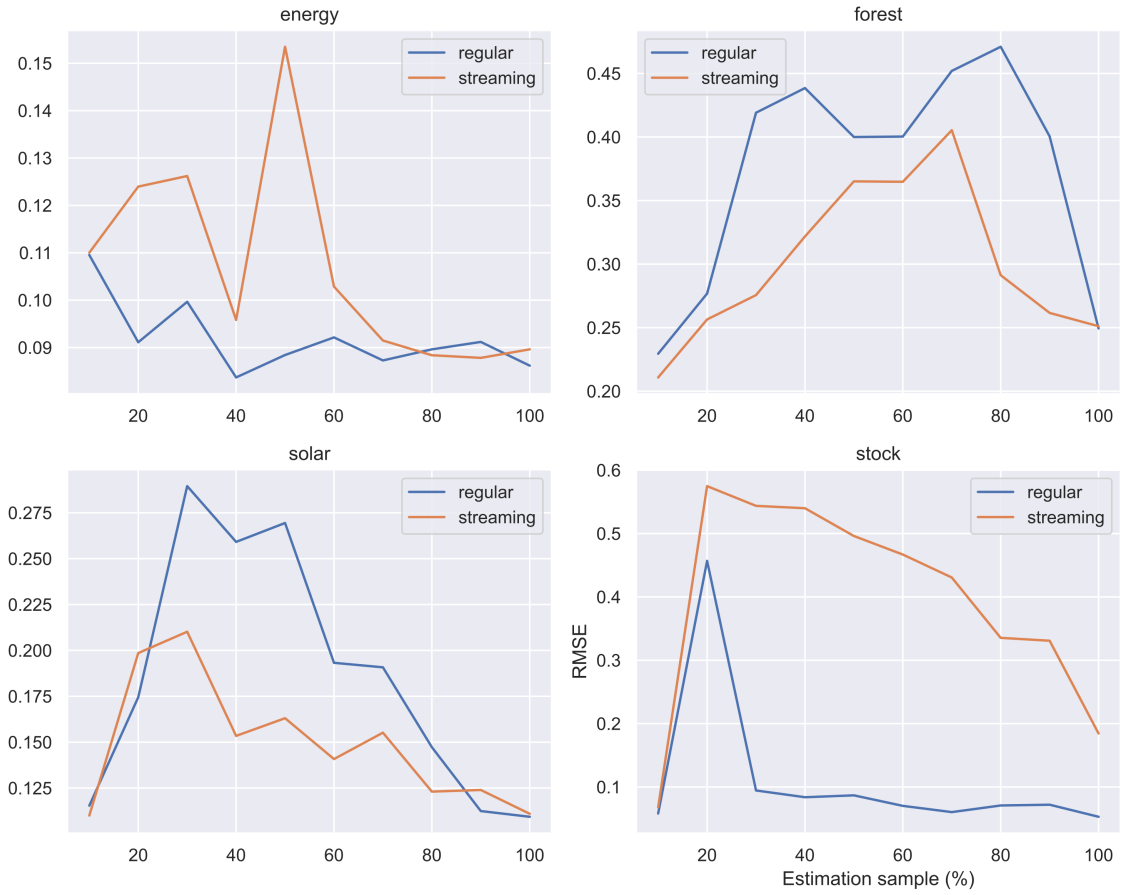

Figure 5.6: Plots of the RMSE of a VSRF model updated in a streaming fashion compared to regular offline estimation for four evaluation datasets.

5.4 Application: Scarce Promotion Problem

In this section, the results for two simulations of the Scarce Promotion Problem are discussed. The first simulation has a linear reward specification. While in the second simulation, a non-linear reward is introduced. The VSRF model is a more sensible choice given a non-linear reward function. In each simulation run 500 customers visit an online shop. The simulation is repeated 50 times and averaged results are reported in the plots and tables below.

5.4.1 Scarce Promotion Problem with linear reward

The trajectory of the cumulative regret of the scarcity corrected Thompson sampler and other benchmark models can be seen in Figure 5.7. In general, the regret of all methods considered appears to be linear over time. It can be seen that the worst-performing benchmarks are the Naive algorithm, which never offers a promotion and the Epsilon-greedy algorithm.

Interestingly the Random algorithm which offers a random promotion to each customer performs the best out of all algorithms. This finding does cast doubt on the added value of the VSRF model. It is likely a consequence of the design of the simulation since there are no large deviations in reward between the different promotions. In Table 5.4, the final cumulative regret values can be seen with their standard deviations. It is clear that the algorithms based on Thompson sampling outperform the Naive and Epsilon-greedy algorithms.

As for the scarcity correction of the Thompson Sampler, a reduction in the cumulative regret can be observed. Hence performing a scarcity correction appears to be beneficial. However given the large standard deviations observed in Table 5.4, the findings discussed above are relatively unstable and are unlikely to hold in general.

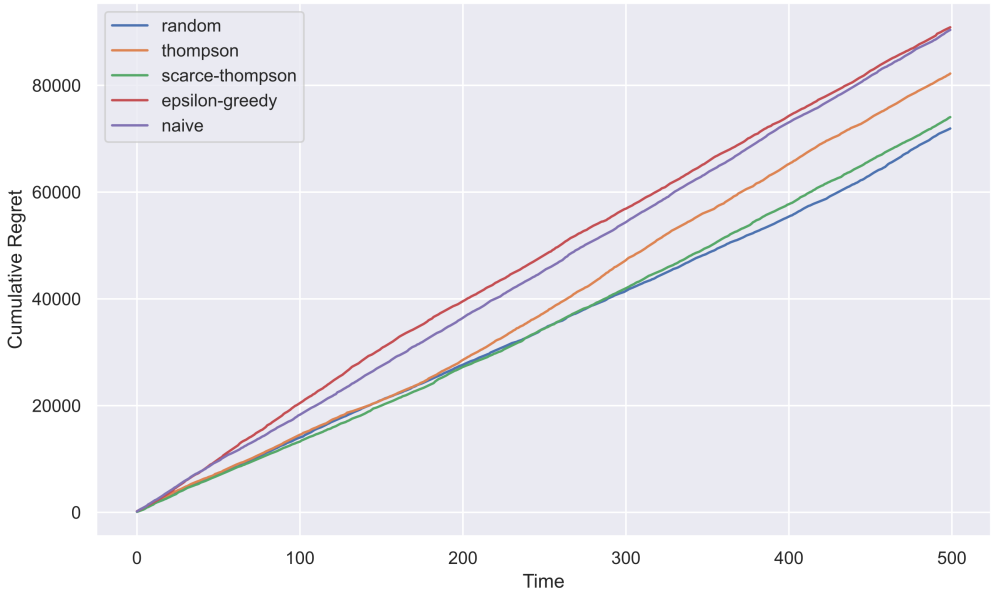

Figure 5.7: Trajectory of average cumulative regret for the scarce Thompson sampler and benchmark models on 50 simulations with linear reward.

Table 5.4: Average cumulative regret over 50 simulation runs of the Scarce Thompson sampler and benchmarks with linear reward.

Algorithm	**Cumulative Regret**
Random	**71919 (33885)**
Naive	90391 (48710)
Epsilon-greedy	90878 (48612)
Thompson sampler	82215 (42571)
Scarce Thompson sampler	74051 (36729)

5.4.2 Scarce Promotion Problem with non-linear reward

In this subsection, the results of the simulation with non-linear reward function are discussed. The results for the scarce Thompson sampler and the other benchmark algorithms can be seen in Figure 5.8. Similarly to the simulation with a linear reward model from Figure 5.7, the Naive and Epsilon-greedy algorithms have the highest cumulative regret. For the Naive algorithm, this finding is in line with my expectation. However, for the Epsilon-greedy algorithm, this finding was not what I expected. I think given the non-linearity of the reward, storing the valuation of a promotion as an average is the reason the Epsilon-greedy algorithm performs poorly.

In the case of a non-linear reward model, the Thompson sampler with scarcity correction outperforms the other methods. This again confirms that a scarcity correction is beneficial with respect to the exploration exploitation trade-off faced. We also see that the Thompson samplers are able to improve over a random assignment in this case, although not with a large margin. However similarly as in the simulation with linear reward specification, due to the large standard deviations observed in Table 5.5, it is likely these findings do not generalize well.

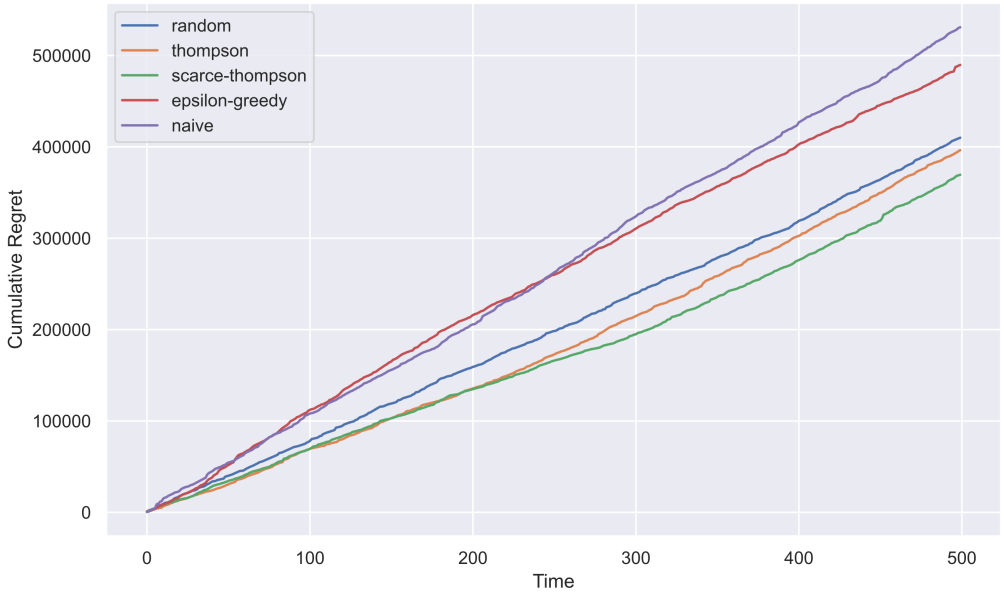

Figure 5.8: Trajectory of the average cumulative regret for the scarce Thompson sampler and benchmark models on 50 simulations with non-linear reward.

Table 5.5: Average cumulative regret over 50 simulation runs of the Scarce Thompson sampler and benchmarks with non-linear reward.

Algorithm	**Cumulative Regret**
Random	410052 (170543)
Naive	530865 (317725)
Epsilon-greedy	489662 (238809)
Thompson sampler	396392 (209239)
Scarce Thompson sampler	**369474 (179167)**

Chapter 6

Conclusion

In this paper, a new way to estimate Bayesian ensembles of decision trees has been evaluated. I have attempted to answer the research question

> How can variational regression trees be combined effectively into a forest model, that is suitable for continual learning?

The proposed variational soft random forest (VSRF) which jointly estimates several variational regression trees has been evaluated comprehensively. It has been shown that the predictive performance of the VSRF is highly dependent on the prediction task at hand, but that it has the potential for variance reduction over a single variational regression tree. The VSRF model is competitive with state-of-the-art ensemble models for some prediction tasks considered, but for others, it produces results that are considerably worse. In particular, in settings with high-dimensional data, the VSRF model is not a viable option. I conclude that the VSRF model is a valuable contribution to the available set of models, but further research is required into why and for which regression tasks the performance of the VSRF deteriorates.

With regards to the online Bayesian update to facilitate continual learning, it has been shown that the updating scheme can reduce computation time by a factor of 11, but this comes at a significant cost in lost predictive power for some evaluation datasets considered. Hence I conclude that the VSRF model is suitable for continual learning, but due care must be taken to establish that the online estimation algorithm works well for the task at hand.

Finally in the main application of the VSRF to the Scarce Promotion Problem, it has been shown that the Thompson sampler using draws from a VSRF model is competitive relative to several benchmark models. However given the large standard deviations of the simulation runs, I am not confident that these findings generalize well to other settings. Further research is desirable to assess how the proposed Thompson sampler using the VSRF compares against more advanced algorithms for limited-pull contextual multi-armed bandits.

6.1 Limitations

My work has several limitations that must be taken into account when evaluating my results and conclusions. The first limitation was the restriction in available computational power available during my experiments. This is why the decision was made to evaluate models with the default hyper-parameters instead of performing tuning for each model to give a more faithful representation of the predictive performance of the models.

A second limitation driven by my lack of computational power was the choice to evaluate the VSRF model on relatively small datasets. Performance on large datasets is of interest in particular for the streaming estimation algorithm.

A third limitation of my work is the focus on the empirical performance of the VSRF, it would have been beneficial if theoretical results were derived on for example the conditions under which a variance reduction can be achieved using a VSRF model relative to a VART.

A fourth limitation of my work is that I use random draws to initialize the hyperparameters of the prior distribution. Although this choice allows for a coherent comparison with Salazar (2024), it is likely that careful construction of the priors will improve the performance of the VSRF.

A final limitation of my work pertains to the simulation used to study the Scarce Promotion Problem. The simulation has several flaws that affect the external validity of the results obtained. A more fruitful evaluation of the Scarce Promotion Problem would be to study it using a field experiment in a real online shop.

6.2 Future work

I conclude with several directions for future work which I think are valuable contributions to the literature. The most straightforward yet potentially promising direction of future work is to consider a boosted model of variational regression trees similar to Cinquin et al. (2023), but estimating the trees jointly instead of applying a boosting step after each tree has been estimated. This differs from my approach since every tree in the VSRF is estimated using either a fixed or bootstrapped estimation sample. However when using a boosting algorithm like AdaBoost (Freund & Schapire, 1997), the samples will be weighted based on the performance of the previous trees. This could increase predictive power but will introduce dependence between the trees which complicates parallel estimation.

A second direction for future work is to extend the spline formulation of the VSRF model from linear splines to higher-order polynomials and assessing its performance. It will be necessary to find ways to reduce the variance of the spline variant of the VSRF. This can likely be achieved using either hard parameter restrictions or experimenting with different priors.

A third direction for future work lies in the improvement of online variational approximations by for example considering adaptive regularization of the ELBO based on the sample size of the micro-batches. In this way the contribution of the prior during the optimization problem is varied based on how much new evidence is observed.

A fourth direction for future work lies in adjusting the VSRF model to be suitable for use in settings with high-dimensional data. I expect that this can be achieved by selecting only a random subset of the covariates for each interior node. While for the leaf nodes variable selection using either a Laplace prior or a horseshoe prior (Carvalho et al., 2009) can be considered.

A fifth direction for future work could be trying to find ways to get interpretable predictions from a VSRF model. Possibly by extending the approach of Campbell et al. (2022), who develops a procedure to find exact Shapley values for decision tree ensembles.

A final area for future work lies in the empirical evaluation of models like the VSRF to see how the model performs in practice and based on that enhance simulations for future studies to be more realistic.

References

Ambrogioni, L., Güçlü, U., Güçlütürk, Y., Hinne, M., van Gerven, M. A. & Maris, E. (2018). Wasserstein variational inference. *Advances in Neural Information Processing Systems*, *31*.

Andrieu, C., Doucet, A. & Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *72*(3), 269–342.

Angrist, J. D., Imbens, G. W. & Rubin, D. B. (1996). Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, *91*(434), 444–455.

Asuncion, A. & Newman, D. (2007). *Uci machine learning repository.* Irvine, CA, USA.

Badanidiyuru, A., Kleinberg, R. & Slivkins, A. (2018). Bandits with knapsacks. *Journal of the ACM (JACM)*, *65*(3), 1–55.

Barton, B., Zlatevska, N. & Oppewal, H. (2022). Scarcity tactics in marketing: A meta-analysis of product scarcity effects on consumer purchase intentions. *Journal of Retailing*, *98*(4), 741–758.

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

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C. & Jordan, M. I. (2013). Streaming variational bayes. *Advances in neural information processing systems*, *26*.

Campbell, T. W., Roder, H., Georgantas III, R. W. & Roder, J. (2022). Exact shapley values for local and model-true explanations of decision tree ensembles. *Machine Learning with Applications*, *9*, 100345.

Carvalho, C. M., Polson, N. G. & Scott, J. G. (2009). Handling sparsity via the horseshoe. In *Artificial intelligence and statistics* (pp. 73–80).

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... others (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, *1*(4), 1–4.

Chérief-Abdellatif, B.-E., Alquier, P. & Khan, M. E. (2019). A generalization bound for online variational inference. In *Proceedings of machine learning research* (Vol. 101, p. 662-677).

Chipman, H. A., George, E. I. & McCulloch, R. E. (2010). Bart: Bayesian additive regression trees.

Cinquin, T., Rukat, T., Schmidt, P., Wistuba, M. & Bekasov, A. (2023). Variational boosted soft trees. In *Aistats 2023.*

Cochrane, J. A., Wills, A. G. & Johnson, S. J. (2023). Rjhmc-tree for exploration of the bayesian decision tree posterior. *arXiv preprint arXiv:2312.01577*.

Collier, M. & Llorens, H. U. (2018). Deep contextual multi-armed bandits. *arXiv preprint arXiv:1807.09809*.

Dorogush, A. V., Ershov, V. & Gulin, A. (2018). Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Féraud, R., Allesiardo, R., Urvoy, T. & Clérot, F. (2016). Random forest for the contextual bandit problem. In *Artificial intelligence and statistics* (pp. 93–101).

Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119–139.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, *19*(1), 1–67.

Frosst, N. & Hinton, G. (2017). Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*.

Gelman, A., Lee, D. & Guo, J. (2015). Stan: A probabilistic programming language for bayesian inference and optimization. *Journal of Educational and Behavioral Statistics*, *40*(5), 530–543.

He, J., Yalov, S. & Hahn, P. R. (2019). Xbart: Accelerated bayesian additive regression trees. In *The 22nd international conference on artificial intelligence and statistics* (pp. 1130–1138).

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

Hillstrom, K. (2008). The minethatdata e-mail analytics and data mining challenge.

Irsoy, O., Yıldız, O. T. & Alpaydın, E. (2012). Soft decision trees. In *Proceedings of the 21st international conference on pattern recognition (icpr2012)* (pp. 1819–1822).

Jang, E., Gu, S. & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Kim, J. & Rockova, V. (2023). On mixing rates for bayesian cart. *arXiv preprint arXiv:2306.00126*.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. & Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of machine learning research*, *18*(14), 1–45.

Lakshminarayanan, B., Roy, D. & Teh, Y. W. (2013). Top-down particle filtering for bayesian decision trees. In *International conference on machine learning* (pp. 280–288).

Lakshminarayanan, B., Roy, D. & Teh, Y. W. (2015). Particle gibbs for bayesian additive regression trees. In *Artificial intelligence and statistics* (pp. 553–561).

Linero, A. R. (2022). Softbart: Soft bayesian additive regression trees. *arXiv preprint arXiv:2210.16375*.

Naesseth, C., Ruiz, F., Linderman, S. & Blei, D. (2017). Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial intelligence and statistics* (pp. 489–498).

O'Neill, E. (2021). *Essays on tree-based methods for prediction and causal inference* (Unpublished doctoral dissertation). University of Cambridge.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox & R. Garnett (Eds.), *Advances in*

neural information processing systems 32 (pp. 8024–8035). Curran Associates, Inc.

Ranganath, R., Gerrish, S. & Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics* (pp. 814–822).

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Ronen, O., Saarinen, T., Tan, Y. S., Duncan, J. & Yu, B. (2022). A mixing time lower bound for a simplified version of bart. *arXiv preprint arXiv:2210.09352*.

Salazar, S. (2024). Vart: Variational regression trees. *Advances in Neural Information Processing Systems*, *36*.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I. & McCulloch, R. E. (2022). Bayes and big data: The consensus monte carlo algorithm. In *Big data and information theory* (pp. 8–18). Routledge.

Slivkins, A. et al. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, *12*(1-2), 1–286.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3-4), 285–294.

Ustimenko, A. & Prokhorenkova, L. (2021). Sglb: Stochastic gradient langevin boosting. In *International conference on machine learning* (pp. 10487–10496).

Wang, C., Paisley, J. & Blei, D. M. (2011). Online variational inference for the hierarchical dirichlet process. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 752–760).

Zeno, C., Golan, I., Hoffer, E. & Soudry, D. (2018). Bayesian gradient descent: Online variational bayes learning with increased robustness to catastrophic forgetting and weight pruning. *arXiv preprint arXiv:1803.10123*.