Bayesian Nonlinear Modeling with Many Predictors

Master Thesis Business Analytics and Quantitative Marketing

Student: Thijs Wijnberg
Student ID: 385754

Supervisor: Prof. dr. D. Fok
Second assessor: Prof. dr. R. Paap

Abstract

By using smooth effect for predictors, a researcher is relieved of the burden of assuming a specific functional form for how a predictor influences the response variable. For a data set with many predictors, estimation of smooth effects might become difficult computationally, and overfitting might occur. Our research overcomes these issues by using the Spike and Slab Generalized Additive Model (SSGAM) proposed by Scheipl et al. (2012). This Bayesian method estimates smooth effects, including interaction effects, and shrinks small effects to prevent overfitting. The contribution of our paper is to make this methodology feasible for a data set with many predictors. We propose to apply a first step of variable selection with DART proposed by Linero (2018), which performs variable selection with a Bayesian modification of a decision tree ensemble. In this way, we can estimate smooth effects for a data sets with many predictors. Our proposed methodology is used to model the choice of viewing a premiere of a new TV series on prime time TV in the US. We visualize the estimated smooth effects to provide new insights into how advertising, demographic variables and TV viewing behavior influence consumer behavior. The predictive performance only drops slightly compared to a competitive benchmark, while interpretation is greatly improved.

Keywords: Generalized Additive Model, Bayesian inference, Function selection

October 17, 2018

1 Introduction

Since the start of the use of scanning equipment in the packaged good industry, the focus of marketing research has shifted. The availability of consumer behavior data on the individual level allows marketing researchers to investigate how consumers react to promotions and price reductions. The explosion in the amount and variety of data greatly influenced statistical research in marketing (Rossi and Allenby, 2000). Demonstrated by the highly cited pioneering research with scanning equipment of Guadagni and Little (1983), this area of marketing research has been of great value, both academically and practically.

In the current digital landscape, there are plenty of methods to obtain and store data of consumer behavior on the individual level. For businesses, insight into individual customer behavior became an essential part of marketing strategy. Uncovering the preferences of individuals by using marketing analytics allows for tailoring products and services to consumers demand. This ultimately increases the equilibrium profit of a firm (Iyer et al., 2005).

However, the value of complex marketing analytics models should not be overstated. Throughout the academic marketing literature, complex parametric models barely outperform simpler ones on data sets of small to moderate size. Often, the introduced complexity causes an unfavorable bias-variance trade-off. Complex parametric models support a richer representation of the data-generating mechanism. However, this might increase the variance of the parameters at the same time, which may ultimately cause the model to over-fit the data (Wedel and Kannan, 2016). Thus, it seems challenging to develop complex models which have appealing properties in terms of interpretation, while also providing good predictive accuracy.

To successfully create a model on a large data set, the researcher typically needs to apply methods which are (*i*) able to handle a huge amount of observations, (*ii*) select relevant predictors among a large set and (*iii*) estimate nonlinear relations between the response variable and selected predictors. Luckily, the collaborations between statisticians and computer scientists have resulted in a vast range of methods which are able to carry out these tasks (Varian, 2014).

Our paper focuses on several techniques in data science that are able to carry out these tasks. Specifically, we focus on the Spike and Slab Generalized Additive Model (SSGAM), introduced by Scheipl et al. (2012). With this method, we are able to estimate and visualize the nonlinear relations that are found between the predictors and the response variable, including interactions of the predictors. In this way, we do not have to make simplifying assumptions on how a predictor affects the response variable. This method decomposes the smooth effect of each predictor into separate orthogonal components with a clear interpretation. Next, function selection is applied to every component, such that an effect is only estimated if sufficient evidence for that effect is found in the data.

However, when we use this model on a data set with many predictors, the computation time rises such that it no longer is feasible to use this method. We propose a solution for this issue by using the SSGAM in combination with a suitable variable selection method. As the SSGAM methodology models smooth effects, it is important that we use a variable selection method which also does not assume specific relations of a specific shape between predictors and the response variable a priori. We address this by using DART variable selection proposed by Linero (2018) to select a subset of predictors that are most likely to exert the largest influence on the response vari-

able. With this method, we can carry out model-free variable selection.In addition, we apply a pragmatic solution to find which pairs of the variables selected by DART are likely to exert an interaction effect.

In a large application on measuring the effectiveness of TV advertising, we show the insights that can be gained by using nonlinear modeling. We use an individual-level data set on TV viewing behavior with 6582 observations and 203 predictors. The largest data set on which SSGAM has previously been used with has only 500 observations and 150 predictors (Scheipl, 2010). By combining this method with the DART variable selection, we open up the possibility to apply the SSGAM on larger marketing applications. With the obtained nonlinear functions, we can gain new insight into how consumer traits affect behavior.

In this application, we model the probability of viewing the premiere episode of a new TV series broadcast on prime time TV in the US. We pay additional attention to advertising effectiveness. In the literature, the shape of the response function to advertising is often studied (Schmidt and Eisend, 2015). Next to this, previous research on marketing effectiveness uncovered that the effect of advertising is dependent on consumer characteristics and advertising types (Vakratsas and Ambler, 1999). Thus, we use our proposed methodology to shed a light on the shape of the advertising response function and investigate how the effect of advertising differs across individuals.

Nonlinear modeling allows a different set of research questions to be answered, compared to an approach where the functional form of a predictor is chosen a priori. The exact questions differ for each application, but our methodology allows us to investigate the following general research question in more detail:

What relationships are found between predictors and the response variable in an application with many predictors?

In Section 2, we describe the relevant literature for this research. In Section 3, we describe the used methodology in more detail. Section 4 contains a description of the data set and other models that are used for comparison. In Section 5, we visualize the results of the models and compare the performance to other models. Section 6 contains a discussion on the used methods and topics for future work.

2 Literature Study

In this section, we first discuss the publications concerned with estimating nonlinear effects of predictors. Second, we discuss the methods to carry out variable selection without making restricting model assumption. Fourth, we review the influential publications on individual responses to advertising. Fifth, we review the literature on live TV program choice.

2.1 Nonlinear modeling

In order to use a smooth effect for estimating the effect of a predictor, we use a model with the structure of a Generalized Additive Model (GAM) introduced by Hastie and Tibshirani (1986). Specifically, we use the Spike and Slab Generalized Additive Model (SSGAM) introduced by Scheipl et al. (2012). With this model, the probability of watching a live TV program is modeled with a sum of estimated smooth effects for the predictors in the model. The smooth effect of each predictors is decomposed in multiple orthogonal components, which have a clear interpretation. Function selection is applied, such the effect of a component is shrunken to zero if the effect on the response variable is only has a small. We can inspect the estimated functional form of the effect to obtain more insight into how a predictor influences the response variable. Splines are used to model the smooth effects of the predictors. Thus, before describing the literature on the SSGAM, we review the methodology on using splines in modeling.

Often, a cubic spline is used to model the nonlinear relation between a predictor and the response variable (Hastie et al., 2009, Ch. 5.2). A problem associated with estimating cubic splines is the selection and placement of knots, which determine how flexible the shape of the spline is allowed to be. Eilers and Marx (1996) propose the penalized B-spline methodology (renamed as P-spline), which uses a large number of equidistant knots and puts difference penalties on the spline coefficients. This method circumvents the problem of having to select an appropriate set of knots for a spline. Instead of the selection of knots, the spline can be tuned with a single smoothness parameter, which governs how much we allow the spline to oscillate. In practice, P-splines have been very popular for modeling smooth effects in additive models (Eilers et al., 2015).

A Bayesian adaptation of the P-spline methodology is introduced by Lang and Brezger (2004). This method allows for Bayesian inference of the model and the use of different penalty structures for the spline coefficients via their prior distribution. This method also allows for the estimation of two dimensional splines, which are smooth surfaces to describe the joint effect of two numerical predictors. This is useful when two predictors are expected to interact with each other. For example, these smooth surfaces allow us to investigate the interaction effects between demographic variables and advertising effectiveness.

Brezger and Lang (2006) introduce the generalized Structured Additive Regression (STAR) model, which is the generalization of the Bayesian P-spline model (2004) that can be used for any response variable from an exponential family, such as a binary response variable. The estimation of this model is difficult when a large number of predictors is added to the model, as multiple parameters are required for each spline.

In order to make the STAR methodology feasible with a larger set of predictors, Kneib et al. (2011) propose the regularized STAR. For a small number of variables

for which the nonlinear relation with the response variable is of interest, Bayesian P-splines are used to model the functional form. For a large set of categorical variables, sparse linear regression with the Bayesian lasso (Park and Casella, 2008) is applied. This allows the STAR methodology to be applied on high dimensional data with a large amount of categorical predictors and a small amount of numerical predictors for which splines are estimated. Unfortunately, this method does not alleviate the computational burden of estimating splines for the numeric variables. Many parameters need to be estimated for each spline of a numeric variable. Therefore using many numeric variables make the method demanding computationally.

Scheipl et al. (2012) extend the STAR methodology such that function selection can be applied on the smooth effects. In their proposed Spike and Slab Generalized Additive Model (SSGAM), each basis of a univariate splines is separated into two orthogonal components. These components can be interpreted in terms of the linear trend and the nonlinear part of the smooth effect. The design matrices of the orthogonal components contain fewer columns, such that the number of parameters needed to estimate a smooth effect is reduced. Next to that, we can use the design matrices of these components to obtain a decomposed version of a bivariate spline.

For the Bayesian P-splines in the STAR, the linear trend of the spline was implicitly unpenalized in the model. This might be undesirable, as we often do not know whether a linear trend is suitable to model the effect of a predictor. By decomposing each spline onto multiple orthogonal components with the SSGAM, we can apply function selection separately on all components. Then, the effect of a component is only added in the model if it exerts enough influence on the response variable. In order to select which components are relevant, the SSGAM carries out the function selection by using a variant of the spike-and-slab priors introduced by Mitchell and Beauchamp (1988). This method makes sure that a parsimonious model is obtained where all irrelevant effects are shrunk to zero.

Next to the application in medical research to study the spread of diseases (e.g. Lai et al. (2015) and Chammartin et al. (2013)), the SSGAM has also been used in marketing research to find customer characteristics that should be used for targeting potential new customers (Tillmanns et al., 2017).

Until now, the largest data sets that were used to model a binary variable with the SSGAM consist of roughly 3000 observation and 45 predictors and 500 observation and 125 predictors. In these models, no interaction effects were added. The computation times for SSGAM are much better compared to the STAR approach, but the computation time still increases to unfeasible levels when a larger number of smooth effects is estimated. This is especially the case when many interactions effects are estimated, as roughly 40 parameters needs to be sampled for each smooth surface in the model (Scheipl, 2010).

This makes it difficult to apply this method on applications with many individuals and predictors. The contribution of our paper is to combine a suitable variable selection method with the SSGAM in order to make this method feasible for larger data sets. Ideally, we would like to detect all variables for which an effect of a reasonable size could be estimated in the SSGAM. For that purpose, we use a variable selection technique that does not assume relations of a specific shape between the response variable and predictors a priori. The method we use to carry out variable selection is discussed next.

2.2 Variable Selection

Bayesian Additive Regression Trees (BART) proposed by Chipman et al. (2010) can be used to carry out variable selection without having to specify relations of a specific shape between predictors and response variable a priori. Tree structures are suitable for this task, as they implicitly take nonlinearities and variable interactions into account. BART identifies which variables exert an influence on the response variable by using a sum of Bayesian Regression Trees (Chipman et al., 1998). The influence of each individual tree in the ensemble is kept small by imposing a prior on the tree components. Using a sum of trees causes each tree to explain a small, but different portion of the response variable. The trees are fitted using a tailored version of Bayesian backfitting (Hastie and Tibshirani, 2000). This method was initially developed to have good predictive performance, but can also be used for variable selection. The BART method produces a subset a variables which are used to split on in the sum-of-trees model for each posterior draw. The proportion of draws where a variable is included in the BART model can be used for the purpose of variable selection. However, these inclusion proportions are found to be very sensitive to the hyperparameter settings, which makes it more difficult to perform variable selection with BART.

Bleich et al. (2014) proposed a permutation test which allows to determine whether the influence that a predictor exerts on the response variable is likely to be real, regardless of the chosen hyperparameter setting. This test generates an interval of variable inclusion proportions for randomly permuted data. With this, we can determine whether the found proportion of draws that include a certain variable is large enough, such that the predictor is likely to have a real effect on the response variable. They concluded that their method performs competitively to a wide range of variable selection methods in various simulations and applications. However, this permutation-based approach requires refitting the BART many times, which makes the variable selection demanding computationally.

Linero (2018) extended the BART framework by using a sparsity-inducing Dirichlet prior on the probability that a predictor is chosen for a split in a tree. This modification removes the sensitivity to the hyperparameters, so that we can carry out variable selection without having to use a computationally demanding permutation test. In addition, this method generates a posterior distribution of variable inclusion probabilities, which gives us insights into the relative importance of the variables. This Dirichlet prior BART model is named DART by the authors.

As the estimation of interaction effects in the SSGAM is demanding, we only estimate an interaction effect of two variables if they are likely to have a real effect on the response variable. We use a Friedman's H-statistic (Friedman and Popescu, 2008) to find interaction effects between variables that are likely to exert influence on the response variable. With this statistic, we are able quantify the relative importance of all two-way interactions with the advertising variables. On the basis of this, we add a small number of interactions which are most likely to have an effect. The calculation of the Friedman's H-statistics requires making many predictions. As DART is a Bayesian model, a prediction is made for every draw in the posterior. Therefore, it is too demanding computationally to calculate the Friedman H-statistic with the DART model. Instead, we use a Random Forest (Breiman, 2001) on subset of predictors selected by DART. For a Random Forest, the Friedman H-statistics can be calculated quickly, with a good performance in general (Friedman and Popescu, 2008).

2.3 Individual Response to Advertising

Each year, TV advertisers in the US spend 70 billion dollars on advertising. Regardless of the emergence of other popular media platforms, TV advertising still is one of the largest advertising platforms, and is predicted to remain substantial[1]. Much research on advertising effectiveness has been done, from which we use the findings to select what variables should be created in our application.

In an attempt to bring together insights from both academics and industry professionals, Pechmann and Stewart (1988) carried out a qualitative literature synthesis on advertising effectiveness. They compared the research on how individuals react to advertising in an experimental setting to the research on how individuals react in practice. In an experimental setting, desired results of advertising such as brand recall, brand attitude and purchase intentions peak at roughly three ad exposures. After more exposures, brand recall stops increasing and brand attitude and purchase intentions even start to decline. Thus, these studies concluded that consumers should be exposed to three ads for maximum effect. This conclusion echoes the much followed advice from Krugman (1972), which is often used as a rule of thumb by industry practitioners.

When casually watching television, it is much less likely that undivided attention is being paid to advertising, which causes the results to be different in practice. In general, the effect of seeing an additional ad is found to be diminishing. However, the effect continues to be positive even after a consumer has been exposed to an ad multiple times. Additionally, the magnitude of the effect varies across different ad types, such as mostly verbal ads, or ads that contain emotion evoking images. At the time of writing, Pechmann and Stewart (1988) encouraged researchers to measure the magnitude of how this effect changes, which has been extensively studied since then.

Schmidt and Eisend (2015) summarize the found answers on the questions posed by Pechmann and Stewart. In their meta-analysis, they describe which variables are found to have an influence on advertising effectiveness. They conclude that the time between exposures and the amount of personal involvement with an ad are the most important variables that affect advertising effectiveness. Thus, to correctly calculate the total effect of advertising on an individual, the ad types and the time between the exposures to each ad should be taken into account.

Next to characteristics of the ad, the characteristics of an individual, and thus how advertising is experienced, are also found to influence the effect of advertising. (Vakratsas and Ambler, 1999; Campbell and Keller, 2003). To take account of this observable heterogeneity in consumer behavior, we incorporate both individual and ad specific information when modeling the effect of advertising. We allow the effect of advertising to vary across individuals by adding interactions between the advertising variables and individual characteristics. Consequently, the estimated effects for these interactions inform an advertiser on which ad types have been most effective, and which individuals responded to advertising most clearly.

[1] `https://www.emarketer.com/content/us-tv-ad-spending-to-fall-in-2018`

2.4 Live TV Program Choice

The research on the effect of advertising is mostly focused on the consumer goods industry. However, TV ads are also often aimed at increasing viewership rates for a program aired in the near future. The literature on this topic guides us on how we should model live TV viewing choice and what variables unrelated to advertising should be created and added to the model in our application.

In a meta-analysis on live TV program choice modeling, Webster and Wakshlag (1983) conclude that live TV program choice mostly depends on the availability of a viewer and not on program content. First, the viewer decides whether to watch TV or not. Second, a choice for a specific program is made. They conclude that it is unlikely that viewers are drawn towards watching live TV solely by program content.

With the introduction of streaming services such as Netflix and Hulu, it becomes even less likely that viewers are attracted to live TV viewing by program content. These streaming services give TV viewers more control over TV program choice when consuming media. Now, TV viewers can watch their preferred TV series at any given moment if it is available on a streaming service (Schweidel and Moe, 2016). With this broader supply of different channels to consume media from, it is even harder to predict whether a consumer will choose to watch a new TV series on live TV. Therefore, we decide to focus on the individuals that already chose to watch TV on the evening of the modeled TV series. In this way, we model the live TV program choice, conditional on their availability for viewing.

The method developed in Rust and Alpert (1984) is often used as benchmark for predicting individual TV program choice. The novelty of this publication was to investigate the hypothesis that many viewers are not inclined to switch channels after a TV show ends. They conclude that many viewers do not switch channels and recommend to always include a variable that denotes viewership of the previous program when modeling live TV choice. In an extension of this model, Rust et al. (1992) concluded that the program preferences of individual closely match with the a priori genre categorization of the TV program content by the Nielsen TV panel[2]. This confirms that we can use the total viewing hours of TV series of a certain genre categorization to describe viewing preference of an individual.

Shachar and Emerson (2000) and Danaher and Dagger (2012) conclude that the performance of TV rating models is substantially improved by including series specific random-effects in the choice model. The importance of these random-effects show that there is unobserved heterogeneity in the popularity of different TV series. Thus, when modeling program choice for multiple TV series in one model, unobserved heterogeneity should be taken into account. As the estimation of unobserved heterogeneity is out of the scope of our research, we choose to model only one TV series to circumvent this issue.

The mentioned TV choice models often assume that the individual is aware of all aired programs. In reality, individuals are not fully aware of the available TV content when starting to view. Advertising helps to make the viewer aware of a TV content, and might cause the viewer to watch the program (Webster and Wakshlag, 1983). Therefore, adding the dimension of advertising to a model of live TV program choice seems a useful addition.

Lovett and Staelin (2016) make a distinction between three different ad types for TV content: paid advertising such as launched TV ads, owned advertising such as

[2] http://www.nielsen.com/us/en/solutions/measurement/television.html

a website of a TV series and earned advertising such as the word of mouth effect of TV viewers that discuss events of their favorite TV series on social media. They conclude that paid advertising has the largest effect on increasing TV program choice probability, compared to the other media channels they investigated. The main role of paid advertising is to remind the viewer of the series its existence. Next to that, advertising informs the viewer on how well the series matches its taste.

This paper identifies that TV ads have an important effect on TV program choice. However, paid advertising is broadcast on different channels and the content of the ads usually differ. For these different types of TV ads, advertising effectiveness might also differ, which is not investigated in this paper. Also, the total exposure to advertising is added in the choice models linearly, which might not be a suitable specification. Thus, a valuable addition of our research is that we use a more flexible function to estimate the effects of exposure to different types of advertising.

3 Methodology

In this section, the applied methodology is described in detail. The methods are explained in the order of how we apply them. First, we describe the DART variable selection. Second, we describe the method to detect interactions. Third, we briefly review the general form of a P-spline, before moving on to the methodology of the SSGAM. In order to limit the number of different symbols used, some of the notation is used multiple times across the sections. However, there is no relation between the duplicated notation, except when we refer to it explicitly.

3.1 DART Variable Selection

In order to obtain a subset of predictors which are likely to exert influence on the binary response variable, we use the DART methodology proposed by Linero (2018). This method is an extension to the BART methodology proposed by Chipman et al. (2010). The method takes input $\boldsymbol{X}$, a matrix of size $N \times K$, to predict response variable $\boldsymbol{y}$ by using a sum of binary decision trees. In this sum-of-trees model, each tree explains a small, but different part of the response variable y. This differs from model averaging approaches where each tree makes a prediction for y, which are averaged to obtain the final prediction. When a model averaging approach is used like is done in Chipman et al. (1998), all trees in the ensemble tend to gravitate to a single large tree. Chipman et al. (2010) show that BART models are computationally inexpensive and perform competitively in various experiments and applications in terms of predictive power.

In our research, the DART is used to model a binary response variable. A probit model is used such that

$$P[y_i = 1|\boldsymbol{X}_i] = \hat{y}_i = \Phi(f(\boldsymbol{X}_i) + \mu_0) \tag{1}$$

where $\Phi(\cdot)$ denotes the standard normal CDF, $f(\boldsymbol{X}_i)$ denotes the sum of binary decision trees and $\boldsymbol{X}_i$ denotes row i of $\boldsymbol{X}$. μ_0 denotes a scalar offset, which is described below. The goal of this method is to obtain $f(\boldsymbol{X}_i) = \sum_{t=1}^{T} g(\boldsymbol{X}_i; \mathcal{T}_t, \boldsymbol{\mu}_t)$, where $g(\boldsymbol{X}_i; \mathcal{T}_t, \boldsymbol{\mu}_t)$ corresponds to the part of y_i that is predicted by tree t. Here, $\mathcal{T}_t$ denotes the topology and splitting rules of tree t, and $\boldsymbol{\mu}_t = (\mu_{t1}, \ldots, \mu_{tL_t(\mathcal{T}_t)})'$ denotes the vector of length $L_t(\mathcal{T}_t)$ of prediction values at the terminal nodes of tree t.

The topology and splitting rules $\mathcal{T}_t$ consists of a sequence of decision rules with corresponding splitting values. These rules determine how many terminal nodes $L_t(\mathcal{T}_t)$ tree t has, and which terminal node is reached with input $\boldsymbol{X}_i$. The path to one of the terminal nodes is determined by decision rules in the form of $[X_{ij} > c_{ts}]$, where X_{ij} denotes the value of predictor j for individual i, and c_{ts} denotes the sampled splitting value for split s in tree t. These rules determine what direction is taken at a split in a tree. After a sequence of splits, a terminal node is reached and the value from $\boldsymbol{\mu}_t$ which corresponds to that terminal node is used as the prediction of the tree. Thus, an individual tree predicts $g(\boldsymbol{X}_i; \mathcal{T}_t, \boldsymbol{\mu}_t) = \mu_{tl}$ if the topology and splitting rules $\mathcal{T}_t$ lead to terminal node l with input $\boldsymbol{X}_i$, where μ_{tl} denotes the prediction value for the l^{th} terminal node.

In order to obtain samples of $\mathcal{T}_t$ and $\boldsymbol{\mu}_t$ for $t = 1, \ldots, T$, we use a modification of Bayesian backfitting (Hastie and Tibshirani, 2000). First, we describe the used prior distribution for the parameters that govern the tree structures. Second, we describe

the method to obtain samples of the posterior distribution $p((\mathcal{T}_1, \boldsymbol{\mu}_1), \ldots, (\mathcal{T}_T, \boldsymbol{\mu}_T)|\boldsymbol{y})$. Third, we describe how we use the output of the model.

3.1.1 Priors on the Tree Structures

In order to simplify the prior distributions, independence of the prior distributions is assumed across trees and terminal nodes, such that

$$p((\mathcal{T}_1, \boldsymbol{\mu}_1), \ldots, (\mathcal{T}_T, \boldsymbol{\mu}_T)) = \prod_t^T p(\boldsymbol{\mu}_t|\mathcal{T}_t)p(\mathcal{T}_t) = \prod_t^T \left(p(\mathcal{T}_t) \prod_l^{L_t(\mathcal{T}_t)} p(\mu_{lt}|\mathcal{T}_t) \right).$$

This simplifies the prior distributions such that the tree components $(\mathcal{T}_t, \boldsymbol{\mu}_t)$ are independent across the trees and all elements in $\boldsymbol{\mu}_t$ are independent within each tree.

We independently generate the trees as follows. Each tree structure is initialized at depth $d = 0$. A node is given two child nodes of depth $d+1$ with probability $q(d)$, where $q(d) = \gamma(1+d)^{-\lambda}$ with $\gamma \in (0,1)$ and $\lambda \in [0, \infty)$. Thus, a node is terminal with probability $1 - q(d)$. The tree keeps increasing in depth until all nodes are terminal. Selecting larger values for γ and smaller values for λ result in deeper trees. This probability q(d) induces a prior distribution on the number of terminal nodes in a tree $L_t(\mathcal{T}_t)$. Figure 1 shows the prior distribution of $L_t(\mathcal{T}_t)$ for hyperparameters $\gamma = 0.95$ and $\lambda \in \{1, 2\}$. As can be seen, the priors cause the individual trees to have few terminal nodes, with 31% and 60% of the trees having less than three terminal nodes for $\lambda = 1$ and $\lambda = 2$, respectively. This prior shrinks the size of each tree to a simpler fit, such that we need multiple trees make capture the effect of all variables in $\boldsymbol{X}$ on $\boldsymbol{y}$. With this way, we cause each tree in the ensemble to explain a different, independent part of $\boldsymbol{y}$. This prior causes the conceptual difference between this a sum-of-trees model and a model averaging approach where each tree in the ensemble models $\boldsymbol{y}$ independently.

The degree of interaction effects between the predictors is also governed by the depth of each individual tree. Deep trees have the possibility of splitting on a sequence of different predictors before arriving at the terminal node. In this way, a tree structure adds interaction effects between the predictors. By changing the hyperparameters γ and λ, the prior probability to increase in depth, $q(d)$, is larger for higher d. In this way, DART can be configured to allow for more interaction effects in the trees. This is desirable for our SSGAM described next, as we also want to add

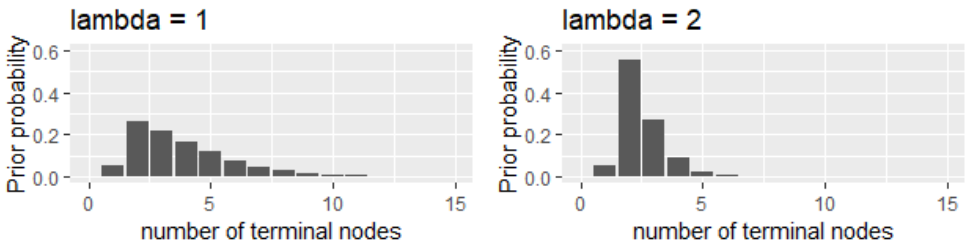

Figure 1: Prior probability for the number of terminal nodes in an individual trees in the ensemble. $\gamma = 0.95$ and $\lambda \in \{1, 2\}$.

interaction affect in the specification. We use two hyperparameter settings $\gamma = 0.95$ and $\lambda \in \{1, 2\}$ in the application to investigate this effect.

The splitting probabilities s_j are defined as the probability of selecting variable $\boldsymbol{X}_j$ to split on at any given node in the trees, where $\boldsymbol{X}_j$ denotes the jth column of $\boldsymbol{X}$. These probabilities are the same for all trees in the ensemble. For $\boldsymbol{s} = (s_1, \ldots, s_K)$, we use the prior distribution $\boldsymbol{s} \sim \mathcal{D}(\alpha/K, \ldots, \alpha/K)$, where $\mathcal{D}$ denotes a Dirichlet distribution. The use of this prior distribution is proposed by Linero (2018) to improve the variable selection performance of the BART methodology when K is large. Compared to BART, which uses $s_j = K^{-1}$ for all variables, this prior favors using a smaller number of predictors when α/K is small. More favorable properties, such as the asymptotic distribution of the number of predictors included in the model for different α, are described in Linero (2018).

By using this sparsity inducing prior for $\boldsymbol{s}$, we allow only a small number of variables to have a high variable inclusion probability s_j. The α plays an important role in determining how many s_j have drawn values larger than zero. Instead of selecting α as a hyperparameter, it is treated as a random variable. This allows us to let the data determine the number of predictors for which the s_j are larger than zero.

For α, we use the prior $\frac{\alpha}{\alpha+\rho} \sim Beta(a_\alpha, b_\alpha)$. Here, a_α and b_α are selected hyperparameters and hyperparameter ρ corresponds to the guess of the researcher on how many of the variables in $\boldsymbol{X}$ exert influence on the response variable. By default, $a_\alpha = 0.5$ and $b_\alpha = 1$ are selected such that the prior gives additional preference to models with few s_j that have probability mass far from zero. If a researcher has a strong belief about the number of predictors that exert influence on the response variable, ρ can be used to alter the prior distribution on α. In the application, we roughly use $\rho \in \{\frac{K}{10}, \frac{K}{4}, K\}$, such that we can evaluate how sensitive the results are to this hyperparameter. Figure 2 shows boxplots of the number elements in $\boldsymbol{s}$ that are larger than $\frac{1}{K}$ for 100 prior draws of α with $a_\alpha = 0.5$ and $b_\alpha = 1$ for different values of ρ with $K = 100$. Here we can see that the number of $s_j > \frac{1}{K}$ is lower for small values of ρ, such that more sparsity is induced.

After drawing a variable to split on, a splitting rule is constructed as $[X_{ij} \leq c_{ts}]$, when $\boldsymbol{X}_j$ is the chosen splitting variable and c_{ts} the value to split on at split s in tree t. In order to determine the splitting value c_{ts}, a value is drawn from a uniform distribution over all observed values of $\boldsymbol{X}_j$. If the drawn splitting value c_{ts} leads to a splitting rule which contradicts a splitting rule used in a higher node, a new c_{ts} is drawn. If these rules do not exist, a splitting value is drawn which contradicts a

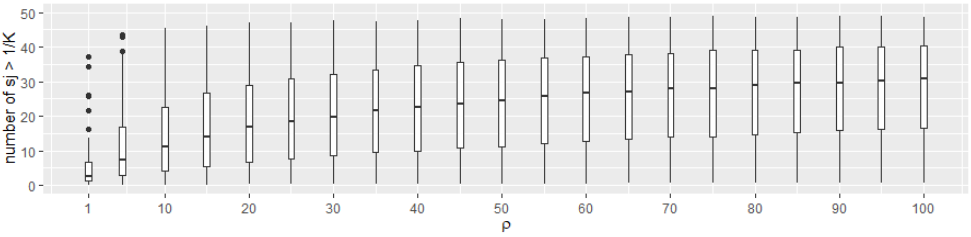

Figure 2: Boxplot of number of $s_j > \frac{1}{K}$ in $\boldsymbol{s}$ for different values of ρ. 100 prior draws of α are taken with $a_\alpha = 0.5$ and $b_\alpha = 1$ and $K = 100$.

previous splitting rule, such that only one of the child nodes can be reached.

Finally, each of the μ_{tl} are drawn with prior distribution $N(0, \sigma_\mu^2)$ where $\sigma_\mu = 3/k\sqrt{T}$ with default choice $k = 2$. Here, the division by T has the effect of limiting the influence of each individual tree in the ensemble, as the μ_{tl} are more shrunk to zero as T increases. This selection of k cause the prediction of the full ensemble $f(\boldsymbol{X}_i)$ to be in the interval of $(-3, 3)$ with a high probability, such that probabilities in the interval of $(0.001, 0.999)$ are obtained.

The prior on μ_{tl} causes $f(\boldsymbol{X}_i)$ to be shrunk towards 0, such that $P[y_i = 1|\boldsymbol{X}_i]$ is shrunk toward 0.5 if there are many trees in the ensemble. To overcome this issue and shrink $P[y_i = 1|\boldsymbol{X}_i]$ to a desired probability p_0, a fixed offset μ_0 in (1) is used with $\mu_0 = \Phi^{-1}(p_0)$ with $p_0 = N^{-1}\sum_{i=1}^{N} y_i$. In this way, the prior on the μ_{tl} parameters shrinks $f(\boldsymbol{X}_i)$ to p_0 instead of 0.5. This data dependent prior is informative, but with $k = 2$, probability mass is obtained for a wide interval of values for $f(\boldsymbol{X}_i)$.

Linero (2018) find that slightly more variables get selected when using a larger number of trees T. We use $T \in \{50, 200\}$ to investigate the impact in our application.

3.1.2 Posterior Sampling with the Backfitting MCMC

To sample the parameters from the posterior density, we use the backfitting MCMC algorithm as described in Chipman et al. (2010). Following the idea of Albert and Chib (1993), latent variables $\boldsymbol{z} = (z_1, \ldots, z_n)$ are independently generated from a truncated normal distribution conditional on y_i as

$$z_i|y_i, \boldsymbol{X}_i, \{\mathcal{T}_t, \boldsymbol{\mu}_t\}_{t=1}^{T} \sim \begin{cases} N(f(\boldsymbol{X}_i) + \mu_0, 1)I[z_i > 0] \text{ if } y_i = 1 \\ N(f(\boldsymbol{X}_i) + \mu_0, 1)I[z_i \leq 0] \text{ if } y_i = 0 \end{cases} \text{ for } i = 1, \ldots, N. \quad (2)$$

such that $z_i > 0$ if $y_i = 1$ and $z_i \leq 0$ if $y_i = 0$. The sum-of-trees model is used to model $z_i = f(\boldsymbol{X}_i) + \mu_0 + \varepsilon_i$, with independently distributed $\varepsilon_i \sim N(0, 1)$ assumed.

A Gibbs sampler is used to obtain successive draws of $(\mathcal{T}_t, \boldsymbol{\mu}_t)$, conditional on $(\mathcal{T}_{(t)}, \boldsymbol{\mu}_{(t)}, \boldsymbol{z}, \boldsymbol{y})$, where $\mathcal{T}_{(t)}$ and $\boldsymbol{\mu}_{(t)}$ denote the draws of all $\mathcal{T}$ and $\boldsymbol{\mu}$, except those for t. Thus, we obtain successive draws from

$$p(\mathcal{T}_t, \boldsymbol{\mu}_t|\mathcal{T}_{(t)}, \boldsymbol{\mu}_{(t)}, \boldsymbol{z}, \boldsymbol{y}) \quad (3)$$

for trees $t = 1, \ldots, T$.

This conditional distribution from (3) only depends on $\mathcal{T}_{(t)}, \boldsymbol{\mu}_{(t)}, \boldsymbol{z}, \boldsymbol{y}$ through partial residuals

$$R_{it} \equiv z_i - \sum_{k \neq t} g(\boldsymbol{X}_i; \mathcal{T}_k, \boldsymbol{\mu}_k) \quad (4)$$

where the partial residuals for all observations is denoted as $\boldsymbol{R}_t = (R_{1t}, \ldots, R_{Nt})$. This $\boldsymbol{R}_t$ corresponds to the fit of the ensemble when tree t is excluded. Thus, the draws from $p(\mathcal{T}_t, \boldsymbol{\mu}_t|\mathcal{T}_{(t)}, \boldsymbol{\mu}_{(t)}, \boldsymbol{z}, \boldsymbol{y})$ are equivalent to the draws from $p(\mathcal{T}_t, \boldsymbol{\mu}_t|\boldsymbol{R}_t)$. This posterior is formally equivalent to the posterior of a single tree model with $R_{it} = g(\boldsymbol{X}_i; \mathcal{T}_t, \boldsymbol{\mu}_t) + u_i$, with $u_i \sim N(0, 1)$ where the effect of all other trees on z_i are captured in R_{it}. The name backfitting MCMC lends its name from the backfitting step of the Bayesian Backfitting method proposed by Hastie and Tibshirani (1986), which is used to fit smooth functions in additive models on partial residuals.

After integrating out on $\boldsymbol{\mu}_t$, we obtain the posterior

$$p(\mathcal{T}_t|\boldsymbol{R}_t) \propto p(\mathcal{T}_t)p(\boldsymbol{R}_t|\mathcal{T}_t) = p(\mathcal{T}_t) \int p(\boldsymbol{R}_t|\boldsymbol{\mu}_t, \mathcal{T}_t)p(\boldsymbol{\mu}_t|\mathcal{T}_t)d\boldsymbol{\mu}_t. \quad (5)$$

With this closed form expression for $\mathcal{T}_t|\boldsymbol{R}_t$, the sampling can be carried out in two steps. This is done by sequentially drawing $\mathcal{T}_t|\boldsymbol{R}_t$ and $\boldsymbol{\mu}_t|\mathcal{T}_t, \boldsymbol{R}_t$ for each tree.

We make a draw for $\mathcal{T}_t|\boldsymbol{R}_t$ by using the Metropolis-Hastings algorithm proposed by Chipman et al. (1998). This algorithm generates a Markov chain of tree structures which converge in distribution to the posterior $p(\mathcal{T}_t|\boldsymbol{R}_t)$ in (5). In the algorithm, a new tree structure is proposed based on the current tree structure using one of four moves, with the probability of each move denoted in parenthesis. When creating the next tree structure in the Markov chain, the next tree grows by

1. Growing a new split with a splitting rule and two terminal nodes (0.25).
2. Removing a split which removes two terminal nodes and a split (0.25).
3. Randomly changing a splitting rule in an internal node in the tree (0.40).
4. Swapping a rule between parent and child node (0.10).

Next, the new tree structure $\mathcal{T}_t^*$ is accepted as the drawn $m+1^{th}$ tree in the chain with probability $\min\left\{\frac{p(\mathcal{T}_t^*)p(\boldsymbol{R}_t|\mathcal{T}_t^*)}{p(\mathcal{T}_t^m)p(\boldsymbol{R}_t|\mathcal{T}_t^m)}, 1\right\}$. Otherwise, the previous draw is used such that $\mathcal{T}_t^{m+1} = \mathcal{T}_t^m$. This stochastic searching algorithm explores the space of different tree structures, without getting stuck in local optima (Chipman et al., 2010).

The draws for $\boldsymbol{\mu}_t|\mathcal{T}_t, \boldsymbol{R}_t$ are independently drawn from a normal distribution using a Gibbs sampler. By using a conjugate prior for $\boldsymbol{\mu}_t$, the sampling for this step is facilitated. With these new draws for the prediction values of this tree, the partial residual for the next tree that is sampled can be calculated.

Thus, the scheme is as follows:

1. Initialize tree structure at depth 0 with one terminal node and draw a corresponding prediction value from the prior $\boldsymbol{\mu}_t \sim N(0, \sigma_\mu)$ for all trees $t = 1, \ldots, T$.
2. Draw $\boldsymbol{z}|\boldsymbol{y}$ with distributions as in (2).
3. Calculate $\boldsymbol{R}_t$ as in (4).
4. Draw $\mathcal{T}_t|\boldsymbol{R}_t$ with the Metropolis-Hastings algorithm from Chipman et al. (1998).
5. Draw $\boldsymbol{\mu}_t|\mathcal{T}_t, \boldsymbol{R}_t$ from a normal distribution using a Gibbs sampler.
6. Repeat steps 3 – 5 for all trees $t = 1, \ldots, T$.
7. Repeat steps 2 – 6 until satisfactory number of samples.

The proposed algorithm allows the structure, size and prediction values for each tree to change at every iteration. Elegantly put by Chipman et al.: "we can imagine the algorithm as analogous to sculpting a complex figure by adding and subtracting small dabs of clay". Chipman et al. (2010) recommend to sample the parameters in one long chain, as mixing problems do not appear to be an issue. Kapelner and Bleich (2016) studied the number of required draws to attain convergence for a various simulations and applications. They conclude that no more than 1000 iterations are required for burn-in. After the burn-in period, an additional 1000 draws are sufficient for inference on the posterior of the $\hat{y}_i$.

More details on the robust performance for various hyperparameter settings can be found in Linero (2018) and Chipman et al. (2010). A more elaborate description of the sampling procedure is described in Kapelner and Bleich (2016). For the estimation of the BART and DART models we use the R-package `BART` (McCulloch et al., 2018).

3.1.3 Using the Output of DART

Following Linero (2018), a predictor $\boldsymbol{X}_j$ is selected if it is used to split on in at least 50% of the draws. This results in the set of selected predictors $\tilde{\boldsymbol{X}}$, a $N \times J$ matrix with $J \leq K$, as a subset of predictors are selected from data set $\boldsymbol{X}$. In order to obtain information on the relative importance of the predictors in $\boldsymbol{X}$, we analyze the draws of splitting variables $\boldsymbol{s}$.

In addition to variable selection, we can also use BART and DART to obtain predictions of the response variable. As prediction value from a posterior sample of M draws, we use $\hat{y}_i = M^{-1} \sum_{m=1}^{M} \Phi(f^m(\boldsymbol{X}_i) + \mu_0)$, where $f^m(\boldsymbol{X}_i)$ corresponds to the m^{th} posterior draw of $f(\boldsymbol{X}_i)$. The predictive performance of these methods is found to be very good in simulations and applications, which make it a strong benchmark for predictive performance (Chipman et al., 2010; Linero, 2018). However, it is harder to visualize how the predictors influence the response variable. We compare the performance of these methods with the performance of SSGAM to determine how much predictive performance is traded in for the possibility of visualizing the effects in SSGAM.

3.2 Interaction Detection

To detect which variables are likely to interact with advertising, we use Friedman's H-statistic (Friedman and Popescu, 2008). For this statistic, a Random Forest (Breiman, 2001) is used to create predictions of $\boldsymbol{y}$ with input data $\boldsymbol{X}$. For each of the variables $\boldsymbol{X}_j$ in the data set, the influence of the variable is measured by using a Partial Dependency (PD) function proposed by (Friedman, 2001)

$$\hat{F}_j(\boldsymbol{X}_j) = N^{-1} \sum_{i=1}^{N} F(\boldsymbol{X}_{i\setminus j}, \boldsymbol{X}_j), \tag{6}$$

where $F(\cdot)$ denotes the predicted value of the random forest and $\boldsymbol{X}_{i\setminus j}$ the i^{th} row of the $\boldsymbol{X}$ matrix, excluding the j^{th} column. In this way $F(\boldsymbol{X}_{i\setminus j}, \boldsymbol{X}_j)$ relates to a prediction of the model, where only the value of $\boldsymbol{X}_j$ is changed. The PD function $\hat{F}_j(\boldsymbol{X}_j)$ is centered around its mean. With this PD function, we can see how the average predicted values $\hat{F}_j(\boldsymbol{X}_j)$ change when the value of $\boldsymbol{X}_j$ is varied for all individuals.

Next, we calculate the PD functions when varying two variables $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$ as

$$\hat{F}_{jk}(\boldsymbol{X}_j, \boldsymbol{X}_k) = N^{-1} \sum_{i=1}^{N} F(\boldsymbol{X}_{i\setminus\{j,k\}}, \boldsymbol{X}_j, \boldsymbol{X}_k), \tag{7}$$

where $F(\boldsymbol{X}_{i\setminus\{j,k\}}, \boldsymbol{X}_j, \boldsymbol{X}_k)$ denotes the prediction value of the random forest when other values are used for $\boldsymbol{X}_j$, $\boldsymbol{X}_k$ while the other variables are not changed. This PD function $\hat{F}_{jk}(\boldsymbol{X}_j, \boldsymbol{X}_k)$ is also centered around its mean.

An interaction between variables $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$ occurs when a change of the predicted value $F(\boldsymbol{X})$ by changing $\boldsymbol{X}_j$ is dependent on the value of $\boldsymbol{X}_k$. Thus, If the variables do not interact in any way, we obtain $\hat{F}_{jk}(\boldsymbol{X}_j, \boldsymbol{X}_k) = \hat{F}_j(\boldsymbol{X}_j) + \hat{F}_k(\boldsymbol{X}_k)$. If the variables interact, this equality does not hold. In that case, the model predicts differently for certain combinations of $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$.

Friedman's H statistic for interaction between variable j and k is defined as

$$H_{jk}^2 = \sum_{i=1}^{N} [\hat{F}_{jk}(\boldsymbol{X}_{ij}, \boldsymbol{X}_{ik}) - \hat{F}_j(\boldsymbol{X}_{ij}) - \hat{F}_k(\boldsymbol{X}_{ik})]^2 / \sum_{i=1}^{N} \hat{F}_{jk}^2(\boldsymbol{X}_{ij}, \boldsymbol{X}_{ik}) \tag{8}$$

where $\boldsymbol{X}_{ij}$ denotes the i^{th} element of $\boldsymbol{X}_j$ and all centered PD functions are used. This statistic measures the proportion of variance of $\hat{F}_{jk}(\boldsymbol{X}_{ij}, \boldsymbol{X}_{ik})$ that is not captured by $\hat{F}_j(\boldsymbol{X}_{ij})$ and $\hat{F}_k(\boldsymbol{X}_{ik})$ for the observed values of $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$. Based on this statistic, we can quantify how much two predictors interact.

As a model to obtain PD functions, other models that include interactions are also suitable to detect variables that interact. Thus, it would be possible to use the DART model that we obtained. However, many predictions need to be made to obtain the PD functions. As DART is a Bayesian method, the prediction takes some time, which make it too demanding computationally to use it to calculate the Friedman H statistic. Instead, we use a Random Forest which is easy to obtain, makes predictions relatively fast and has a good performance for interaction detection in simulations (Friedman and Popescu, 2008). As the model $F(\cdot)$, we use a Random Forest with 250 trees to model $\boldsymbol{y}$ with subset of predictors selected by DART $\tilde{\boldsymbol{X}}$. We use the recommended hyperparameters of the `R`-package `randomForest` (Liaw and Wiener, 2002). For all pairs of predictors in $\tilde{\boldsymbol{X}}$, the Friedman H statistic is calculated as in (8). As we do not want to increase the computational burden too much, we only include the three interactions with the largest H statistic.

3.3 Bayesian P-splines

As the foundation of the SSGAM model, Bayesian P-splines proposed by Lang and Brezger (2004) are used to model the smooth effects of predictors. Both one and two dimensional P-splines are used, which we discuss subsequently.

A spline is a piecewise polynomial for which the parameters of the polynomial are allowed to vary on predefined intervals. In practice, cubic splines are most commonly used (Hastie et al., 2009, Ch.5).A selection of knots is made that govern the intervals on which the spline is can be described by a different cubic polynomial. For each interval between two knots, the parameters that describe the cubic polynomial can be different, while the complete spline is continuous and has a continuous first and second order derivative.

The selection of the number of knots and the placement of knots require more attention. When a small number of knots is used, the nonlinear relation might be captured insufficiently with the spline. However, for a large number of knots, some modification is needed to prevent the spline from overfitting the data. Next to that, the placement of knots has to be such that each interval between knots contains observations, such that a polynomial can be fitted in that interval.

Eilers and Marx (1996) propose a solution to circumvent the problem of having to select the number and positions of knots. They propose to use a large number of equidistant knots and impose restrictions on the parameters that govern the shape of the spline. With this modification, the smoothness of the spline can be governed with a single parameter, which can be tuned to prevent overfitting. Additionally, the extrapolation of the spline is improved by also adding some knots outside the range of observed values of the predictor. This penalized spline is known as the P-spline and has been frequently used since its introduction Eilers et al. (2015).

3.3.1 Basis Representation of a Predictor

When using a P-spline to model the effect of numeric predictor $\boldsymbol{x}$ on response variable $\boldsymbol{y}$, we need to create a basis for $\boldsymbol{x}$. We follow the approach from Eilers and Marx (2010) to create a basis for $\boldsymbol{x}$. First, we calculate the Truncated Power Functions (TPF) for a degree p and a vector of equadistant knots $\boldsymbol{\zeta} = (\zeta_{-p}, \ldots, \zeta_{r+p})$, where r is the selected number of knots used within the range of observed values of $\boldsymbol{x}$. As noted before, $2p$ knots are added outside of the domain to improve extrapolation when using the spline. This gives TPFs

$$g_{ij}(x_i) = (x_i - \zeta_j)^p I[x_i > \zeta_j]$$

for $i = 1, \ldots, N$ and $j = -p, \ldots, r+p$. Next, the basis $\boldsymbol{B}(\boldsymbol{x})$ is constructed by taking the differences of these TPFs as

$$\boldsymbol{B}_j(\boldsymbol{x}) = (h^p p!)^{-1}(-1)^{p+1}\Delta^{p+1} g_j(\boldsymbol{x}), \tag{9}$$

where h denotes the chosen distance between two knots such that $h = (\max\{\boldsymbol{x}\} - \min\{\boldsymbol{x}\})/r$ and Δ^p denotes the p^{th} difference operator which takes the difference of the functions $g_j(\boldsymbol{x})$ as $\Delta^p g_j(\boldsymbol{x}) = \Delta^{p-1}(g_j(\boldsymbol{x}) - g_{j-1}(\boldsymbol{x}))$ where Δ^0 is the identity function. For a cubic spline with $p = 3$, this results in the following difference of TPFs

$$\begin{aligned}
\Delta^4 g_j(\boldsymbol{x}) &= \Delta^3(g_j(\boldsymbol{x}) - g_{j-1}(\boldsymbol{x})) \\
&= \Delta^2(g_j(\boldsymbol{x}) - 2g_{j-1}(\boldsymbol{x}) + g_{j-2}(\boldsymbol{x})) \\
&= \Delta^1(g_j(\boldsymbol{x}) - 3g_{j-1}(\boldsymbol{x}) + 3g_{j-2}(\boldsymbol{x}) - g_{j-3}(\boldsymbol{x})) \\
&= g_j(\boldsymbol{x}) - 4g_{j-1}(\boldsymbol{x}) + 6g_{j-2}(\boldsymbol{x}) - 4g_{j-3}(\boldsymbol{x}) + g_{j-4}(\boldsymbol{x})
\end{aligned}$$

for $j = 2, \ldots, r+p$. The basis representation of $\boldsymbol{x}$ is then described as $\boldsymbol{B}(\boldsymbol{x}) = [\boldsymbol{B}_1(\boldsymbol{x}), \ldots, \boldsymbol{B}_M(\boldsymbol{x})]$, a matrix of size $N \times M$, with $M = r + p - 1$. Thus, each basis is a linear combination of TPFs, which is scaled by the factor $(h^p p!)^{-1}(-1)^{p+1}$, such that $\sum_{j=1}^{M} \boldsymbol{B}_j(\boldsymbol{x}) = 1$ for all elements in $\boldsymbol{x}$. A visual representation for a $N(0,1)$ distributed variable and the basis representation for a cubic spline with 10 knots is shown in Figure 3. For more properties on this basis representation, such as a comparison with other often used basis representations, see Eilers and Marx (2010).

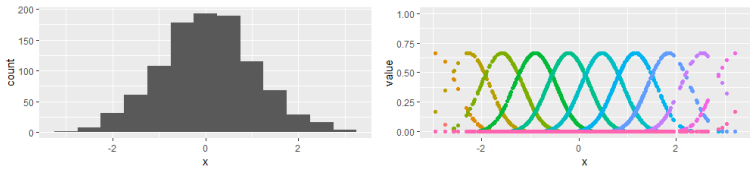

Figure 3: Spline basis representation of $\boldsymbol{x}$ drawn from $N(0,1)$ using a cubic spline with 10 knots. Each color denotes a different $\boldsymbol{B}_m(\boldsymbol{x})$ for $m = 1, \ldots, M$.

3.3.2 Univariate splines

An univariate P-spline as defined in Eilers and Marx (1996) is constructed as

$$f(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{\delta}, \tag{10}$$

where $\boldsymbol{B}(\boldsymbol{x})$ is the basis representation of a cubic spline defined in the previous section. The parameters $\boldsymbol{\delta}$ could be estimated in a linear regression if $\boldsymbol{B}(\boldsymbol{x})$ is of full rank. However, this causes the problem of overfitting described before, which is why the preferred approach is to penalize the $\boldsymbol{\delta}$ parameters. Eilers and Marx (1996) propose to add a penalty in the objective function on the d^{th} order differences of the adjacent spline parameters in $\boldsymbol{\delta}$. This difference penalty causes neighboring parameter estimates in $\boldsymbol{\delta}$ to be drawn towards each other. This results in a smoother spline, which can be seen in Figure 4. Here we see that a more smooth spline is obtained after bringing the elements in $\boldsymbol{\delta}$ closer towards their neighboring elements.

The selection of the difference order determines what effect in the spline is not penalized. With a second order difference penalty, a linear trend of the spline is unpenalized, as all second order differences between the spline parameters are zero if a linear relation exists between a predictor and a response variable (Eilers and Marx, 1996). Higher order difference penalties can also be used to allow for unpenalized estimation of higher order relations between $\boldsymbol{x}$ and y. The second order difference penalty is used in this research to limit the flexibility of the estimated splines.

Lang and Brezger (2004) propose a Bayesian variant of the P-spline by using a specific prior distribution for the spline coefficients $\boldsymbol{\delta}$ to shrink the δ to neighboring elements. Here, the second order difference penalty is replaced by its stochastic analogue. A prior distribution for $\boldsymbol{\delta}$ is used with restrictions in the form of $\delta_k = \frac{4}{6}(\delta_{k-1} + \delta_{k+1}) + \frac{1}{6}(\delta_{k-2} + \delta_{k+2}) + \frac{1}{6}u_k$, with $u_k \sim N(0, \tau^2)$ where τ^2 is a parameter that governs the amount of smoothing.

These restrictions can be denoted more conveniently by using penalty matrix $\boldsymbol{P} = \boldsymbol{\Delta}^{d'}\boldsymbol{\Delta}^{d}$ for a d^{th} order difference penalty, where $\boldsymbol{\Delta}^{d}$ denotes the d^{th} difference operator matrix. Depending on which restriction on neighboring δ_ks the researcher wants to impose, d can be chosen. For example, in the case of $M = 5$ with second order difference penalties ($d = 2$) this results in matrices

$$\boldsymbol{\Delta}^2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{P} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ -2 & 5 & -4 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \\ 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix},$$

such that the prior can be denoted as $\boldsymbol{\delta}|\tau^2 \propto exp\{-(2\tau^2)^{-1}\boldsymbol{\delta}'\boldsymbol{P}\boldsymbol{\delta}\}$, or more compactly

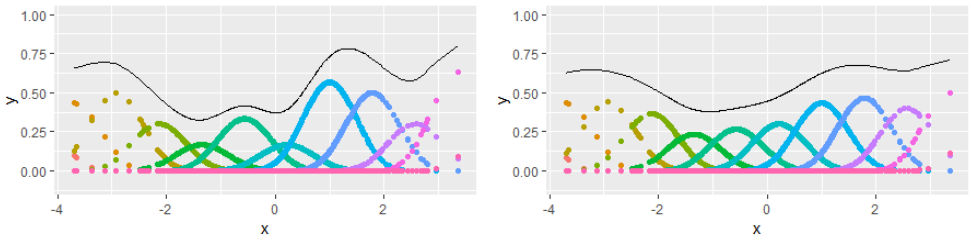

Figure 4: Splines for different degrees of smoothing. Left panel shows cubic spline with 10 knots and chosen $\boldsymbol{\delta}$ parameter. Right panel shows the spline after moving all elements in $\boldsymbol{\delta}$ closer to its neighbors, which result in a spline which is more smooth.

as $\boldsymbol{\delta}|\tau^2 \sim N(0, \tau^2 \boldsymbol{P}^+)$. Here, $\boldsymbol{P}^+$ denotes the generalized inverse of $\boldsymbol{P}$, such as the Moore Penrose inverse (Ben-Israel and Greville, 2005), which needs to be used because the previously described penalty matrix $\boldsymbol{P}$ is not of full rank.

The singular $\boldsymbol{P}$ matrix causes the prior for $\boldsymbol{\delta}$ to be improper. Specifically, the second order difference penalty implies using an improper flat prior on the constant and linear trend of the estimated spline (Lang and Brezger, 2004). Intuitively, this improper prior distribution with a second order difference penalty matrix $\boldsymbol{P}$ has the same "prior density" for $\boldsymbol{\delta} + a\mathbf{1}_M + b[1, \ldots, M]'$ for all values of scalars a and b.

In order to obtain a proper posterior for $\boldsymbol{\delta}$, a non-diffuse prior must be assumed for the smoothing parameter τ^2 (Hobert and Casella, 1996). This is important, because Hobert and Casella (1996) argue that Gibbs sampler draws from improper posterior distributions are ill-behaved in theory. Often, the dispersed prior distribution $\tau^2 \sim IG(a_\tau, b_\tau)$ is used, which results in a proper posterior for $\boldsymbol{\delta}$.

3.3.3 Bivariate splines

This univariate spline framework can also be extended to the bivariate case. A basis for a two-dimension spline is obtained by calculating the Hadamard product of the basis representation of two numeric variables $\boldsymbol{x}$ and $\boldsymbol{z}$ such that

$$f(\boldsymbol{x}, \boldsymbol{z}) = \sum_{m_x=1}^{M_x} \sum_{m_z=1}^{M_z} \delta_{m_x m_z} \boldsymbol{B}_{m_x}(\boldsymbol{x}) \circ \boldsymbol{B}_{m_z}(\boldsymbol{z}), \tag{11}$$

where $\boldsymbol{B}_{m_x}(\boldsymbol{x})$ denotes the m_x^{th} column of the basis representation of $\boldsymbol{x}$ as defined previously, and $\circ$ denotes the operator for the Hadamard product, which performs element-wise multiplication. The basis for a bivariate spline can be conveniently rewritten by as an $N \times (M_x M_z)$ matrix

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{z}) = \Big(& \boldsymbol{B}_1(\boldsymbol{x}) \circ \boldsymbol{B}_1(\boldsymbol{z}), \boldsymbol{B}_1(\boldsymbol{x}) \circ \boldsymbol{B}_2(\boldsymbol{z}), \ldots, \\ & \boldsymbol{B}_{M_x}(\boldsymbol{x}) \circ \boldsymbol{B}_{M_z-1}(\boldsymbol{z}), \boldsymbol{B}_{M_x}(\boldsymbol{x}) \circ \boldsymbol{B}_{M_z}(\boldsymbol{z}) \Big). \end{aligned} \tag{12}$$

The parameter vector can be denoted as $\boldsymbol{\delta}_{xz} = (\delta_{11}, \delta_{12}, \ldots, \delta_{M_x M_z - 1}, \delta_{M_x M_z})$, such that the function in (11) can be denoted as

$$f(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{z}) \boldsymbol{\delta}_{xz}. \tag{13}$$

In order to penalize the parameter vector $\boldsymbol{\delta}_{xz}$, we can use the kronecker product of the two individual penalty matrices can be used $\boldsymbol{P}_{xz} = \boldsymbol{P}_x \otimes \boldsymbol{P}_z$ of size $(M_x M_z) \times (M_x M_z)$. Here $\boldsymbol{P}_x$ denotes the $M_x \times M_x$ penalty matrix that is used for the spline of $\boldsymbol{x}$ and likewise for $\boldsymbol{P}_z$. Now, we have $\boldsymbol{\delta}_{xz}|\tau^2 \sim N(0, \tau^2 \boldsymbol{P}_{xz}^+)$, just as in the one-dimensional case (Lang and Brezger, 2004).

In addition to using a Hadamard product of two numeric predictors, it is also possible to use the Hadamard product of a numerical and categorical variable $\boldsymbol{x}$ and $\boldsymbol{w}$, respectively. In this case

$$f(\boldsymbol{x}, \boldsymbol{w}) = \sum_{m_x=1}^{M_x} \sum_{m_w=1}^{M_w} \delta_{m_x m_w} \boldsymbol{B}_{m_x}(\boldsymbol{x}) \circ I[\boldsymbol{w} = m_w], \tag{14}$$

where $\boldsymbol{w}$ has M_w unique category values. Just like in the previous case, this basis could be written as an $N \times (M_x M_w)$ matrix as

$$\begin{aligned}\boldsymbol{B}(\boldsymbol{x}, \boldsymbol{w}) = \Big(&\boldsymbol{B}_1(\boldsymbol{x}) \circ I[\boldsymbol{w} = 1], \boldsymbol{B}_1(\boldsymbol{x}) \circ I[\boldsymbol{w} = 2], \ldots, \\ &\boldsymbol{B}_{M_x}(\boldsymbol{x}) \circ I[\boldsymbol{w} = M_w - 1], \boldsymbol{B}_{M_x}(\boldsymbol{x}) \circ I[\boldsymbol{v} = M_w]\Big).\end{aligned} \tag{15}$$

Together with parameter vector $\boldsymbol{\delta}_{xw} = (\delta_{11}, \delta_{12} \ldots, \delta_{M_x M_w - 1}, \delta_{M_x M_w})$ of length $M_x M_w$, the function in (14) can be denoted as

$$f(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{w})\boldsymbol{\delta}_{xw}, \tag{16}$$

In this case, no penalty is imposed across categories of $\boldsymbol{w}$ such that $\boldsymbol{\delta}_{xw}|\tau^2 \sim N(0, \tau^2 \boldsymbol{P}_{xw}^+)$ with $\boldsymbol{P}_{xw}^+$ as a $(M_x M_w) \times (M_x M_w)$ block diagonal matrix with M_w matrices $\boldsymbol{P}_x^+$ of size $M_x \times M_x$ on its diagonal. In this way, we can rewrite all considered splines in the form $\boldsymbol{B}\boldsymbol{\delta}$, with $\boldsymbol{\delta}|\tau^2 \sim N(0, \tau^2 \boldsymbol{P}^+)$.

3.4 Spike and Slab Generalized Additive Model

For the estimation of the nonlinear functions with the matrix of predictors $\boldsymbol{X}$ of size $N \times J$ and response variable $\boldsymbol{y}$ of length N, we use the Spike and Slab Generalized Additive model (SSGAM) proposed by Scheipl et al. (2012). An implementation of this method is available in the `SpikeSlabGAM` package in `R` (Scheipl, 2011), which is also used in this research. First, we first describe the general form of a STAR model on which the SSGAM model is based. Second, we describe how the design matrices for the smooth effects in the SSGAM are obtained and what prior distributions are used for the parameters. Third, we give the general model formulation for an SSGAM. Fourth, we describe how we can reparametrize the parameters, such that function selection can be applied. Fifth, we describe how posterior samples for the model are obtained. Sixth, we describe evaluation metrics that we can use to draw conclusions from the output of an SSGAM.

3.4.1 Generalized STAR Model

The SSGAM uses the same foundation as the generalized STAR model proposed by Brezger and Lang (2006). The generalized STAR models a response variable from the exponential family with a sum of penalized splines in a Bayesian way. The generalized STAR model for a binary variable is defined with a logistic regression as

$$E[y_i|\eta_i] = \left(1 + e^{-\eta_i}\right)^{-1} \quad \text{with} \quad \boldsymbol{\eta} = \eta_0 + \sum_{q=1}^{Q} \boldsymbol{\eta}_q. \tag{17}$$

Here, η_0 is the estimated intercept and $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_N]'$ denotes the vector of predictions for all individuals. These predictions are composed of a sum of Q effects $\boldsymbol{\eta}_q$. Each $\boldsymbol{\eta}_q$ relates to a smooth effect that is estimated with an univariate or bivariate spline of one or two of the predictors in $\boldsymbol{X}$. When we use the definitions from (10), (13) and (16) to rewrite the effects $\boldsymbol{\eta}_q$, we obtain

$$\sum_{q=1}^{Q} \boldsymbol{\eta}_q = \sum_{j=1}^{J} f_j\left(\boldsymbol{X}_j\right) + \sum_{u=1}^{J} \sum_{v>u} f_{uv}\left(\boldsymbol{X}_u, \boldsymbol{X}_v\right) \tag{18}$$

where $\boldsymbol{X}_j$ corresponds to the j^{th} column of $\boldsymbol{X}$, $f_j(\boldsymbol{X}_j)$ corresponds to an univariate spline as in (10) and $f_{uv}(\boldsymbol{X}_u, \boldsymbol{X}_v)$ relates to a bivariate spline as defined in (13) or (16), depending on the type of the variables $\boldsymbol{X}_u$ and $\boldsymbol{X}_v$. In this example, we have $Q = J + \frac{J(J-1)}{2}$, such that an interaction is added for every pair of variables.

As was shown in the previous section, we can rewrite each nonlinear function $f_j(\boldsymbol{X}_j)$ or $f_{uv}(\boldsymbol{X}_u, \boldsymbol{X}_v)$ as $\boldsymbol{B\delta}$, where $\boldsymbol{B}$ corresponds to the $N \times M$ basis representation $\boldsymbol{B}(\boldsymbol{X}_j)$ as in (9) or Hadamard product of bases $\boldsymbol{B}(\boldsymbol{X}_u, \boldsymbol{X}_v)$ as in (12) or (15), with corresponding $\boldsymbol{\delta}$ of length M.

As is shown in the previous section, a Bayesian P-spline with a second order difference penalty uses improper prior distribution $\boldsymbol{\delta}|\tau^2 \sim N(0, \tau^2 \boldsymbol{P}^+)$. This penalty structure allows for unpenalized estimation of an intercept and a linear trend of the effect $\boldsymbol{B\delta}$, as we discussed earlier. By centering all effects and adding one intercept in the model, the unpenalized intercept can be removed from the spline. However, the estimation of the linear trend remains unpenalized. As we do not know in what way a predictor exerts influence on the response variable, it is undesirable to estimate the linear trend without penalization. It might as well be that we find a nonlinear effect, but no linear effect. Ideally, we want to be able to penalize the linear trend of $\boldsymbol{B\delta}$ as well, if the data shows little evidence for it.

To achieve this, Scheipl et al. (2012) propose to decompose each univariate spline $\boldsymbol{B\delta}$ into two orthogonal components, on which function selection is applied separately. They propose to split each $\boldsymbol{B\delta}$ into $\boldsymbol{D}_{lin}\beta_{lin} + \boldsymbol{D}_{nl}\boldsymbol{\beta}_{nl}$. Here, $\boldsymbol{D}_{lin}\beta_{lin}$ represents the linear trend of the spline and $\boldsymbol{D}_{nl}\boldsymbol{\beta}_{nl}$ represents the nonlinear part of the spline. Here, we use the design matrices $\boldsymbol{D}_{lin}$ of size $N \times 1$ and $\boldsymbol{D}_{nl}$ of size $N \times S_{nl}$, which are described in more detail in the next section.

By decomposing a univariate spline like this, we can put a prior on both β_{lin} and $\boldsymbol{\beta}_{nl}$, which enables us to apply function selection on both the linear and the nonlinear part of a spline. With function selection, we refer to being able to shrinking a whole effect such as $\boldsymbol{D}_{nl}\boldsymbol{\beta}_{nl}$ to zero if no real effect is found. This is done by putting a specific prior on $\boldsymbol{\beta}_{nl}$, on which we will elaborate in a following section.

In addition, we can use $\boldsymbol{D}_{lin}$ and $\boldsymbol{D}_{nl}$ to create design matrices for bivariate splines, which consist of four orthogonal components. In that way, we obtain a decomposed version of $f_{uv}(\boldsymbol{X}_u, \boldsymbol{X}_v)$ in (18). Thus, we are able to decompose all effects that are used in the generalized STAR model, such that we can penalize parts of effects that were previously unpenalized.

The construction of these new design matrices $\boldsymbol{D}_{lin}$ and $\boldsymbol{D}_{nl}$ is done separately in the following way. We drop the subscripts in the following sections to improve readability. However, we construct a $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_{j,nl}$ for every $f_j(\boldsymbol{X}_j)$ in (18). Next to that, we also describe how we obtain the design matrix $\boldsymbol{D}_w$ for an indicator variable $\boldsymbol{X}_w$. Before diving into the details of how the design matrices are created, we give a summary of all steps that are taken to obtain the orthogonal design matrices for the SSGAM:

1. Create all design matrices $\boldsymbol{D}_{j,nl}$ for the nonlinear part of $\boldsymbol{B}_j\boldsymbol{\delta}_j$ by making use of the spectral decomposition of $\boldsymbol{B}_j\boldsymbol{\delta}_j$.

2. Create all design matrices $\boldsymbol{D}_{j,lin} = \boldsymbol{X}_j$ for the linear trend of $\boldsymbol{B}_j\boldsymbol{\delta}_j$ and $\boldsymbol{D}_w = \boldsymbol{X}_w$ for every indicator variable.

3. Orthogonalize each $\boldsymbol{D}_{j,nl}$ on $[\boldsymbol{1}_N, \boldsymbol{D}_{j,lin}]$ and each $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_w$ on $[\boldsymbol{1}_N]$.

4. Obtain the four design matrices for each interaction effect between $\boldsymbol{X}_u$ and $\boldsymbol{X}_v$ by multiplying the design matrices $\boldsymbol{D}_{u,nl}$, $\boldsymbol{D}_{u,lin}$,$\boldsymbol{D}_{v,nl}$ and $\boldsymbol{D}_{v,lin}$.

5. Obtain the two design matrices for each interaction effect between $\boldsymbol{X}_u$ and $\boldsymbol{X}_w$ by multiplying the design matrices $\boldsymbol{D}_{u,nl}$, $\boldsymbol{D}_{u,lin}$ and $\boldsymbol{D}_w$.

6. Calculate the reduced rank representation for the design matrices for each interaction by using the singular value decompositions of the design matrices.

7. Orthogonalize the reduced rank representation of the design matrices for the interaction on the design matrices with which they were created.

8. Scale all design matrices such that they are of comparable size.

3.4.2 Design Matrix for the Nonlinear Part of a Spline

For each univariate Bayesian P-spline as in (10) we obtain the design matrix of the nonlinear part of the spline $\boldsymbol{D}_{nl}$ by applying a spectral decomposition on the $\boldsymbol{B\delta}$.

For the improper prior distribution $\boldsymbol{\delta}|\tau^2 \sim N(0, \tau^2\boldsymbol{P}^+)$, the effect $\boldsymbol{B\delta}$ has improper prior distribution $\boldsymbol{B\delta}|\tau^2 \sim N(\boldsymbol{0}, \tau^2\boldsymbol{BP}^+\boldsymbol{B}')$. The spectral decomposition of the $\boldsymbol{BP}^+\boldsymbol{B}'$ matrix of size $N \times N$ can be taken such that $\boldsymbol{BP}^+\boldsymbol{B}' = \boldsymbol{UVU}'$, with orthonormal $\boldsymbol{U}$ and diagonal matrix $\boldsymbol{V}$. Here $\boldsymbol{UVU}'$ is the orthogonal basis representation of the covariance of the improper prior of $\boldsymbol{B\delta}$. When a penalty matrix with difference penalty of order d is used, $\boldsymbol{P}$ has rank $M - d$. This causes all eigenvalues, except the first $M - d$, in $\boldsymbol{V}$ to be equal to zero. In this case we can rewrite

$$\boldsymbol{BP}^+\boldsymbol{B}' = \boldsymbol{UVU}' = [\boldsymbol{U}_+\boldsymbol{U}_0]' \begin{bmatrix} \boldsymbol{V}_+ & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} [\boldsymbol{U}_+\boldsymbol{U}_0],$$

where $\boldsymbol{U}_+$ is the $N \times (M - d)$ matrix of eigenvectors that corresponds with the first $M-d$ eigenvectors with nonzero eigenvalue in $\boldsymbol{V}_+$. The $N \times (M-d)$ design matrix for the penalized part of $\boldsymbol{B\delta}$ is obtained as $\boldsymbol{D}_{nl} = \boldsymbol{U}_+\boldsymbol{V}_+^{1/2}$. This reparametrization uses a different prior $\boldsymbol{\beta}_{nl}|v_{nl}^2 \sim N(\boldsymbol{0}, v_{nl}^2\boldsymbol{I})$, with prior variance v^2 and $(M - d) \times (M - d)$ identity matrix $\boldsymbol{I}$. Just as desired, this $\boldsymbol{D}_{nl}\boldsymbol{\beta}_{nl}$ has a Gaussian distribution that is proportional to the improper prior of $\boldsymbol{B\delta}$ (Rue and Held, 2005, eq. 3.16) but only parametrizes the penalized part of $\boldsymbol{B\delta}$.

As we only need the first $M - d$ eigenvectors and eigenvalues for creating $\boldsymbol{D}_{nl}$, it is impractical and computationally demanding to calculate all N eigenvectors in the $N \times N$ matrix $\boldsymbol{BP}^+\boldsymbol{B}'$. Instead, we only calculate the first $M - d$ eigenvectors by using the fast truncated bidiagonalization algorithm (Baglama and Reichel, 2005).

Next to this, we only use a subset of the $M - d$ eigenvectors to obtain $\boldsymbol{D}_{nl}$ to reduce computational complexity even further. Typically, only the first few columns in $\boldsymbol{U}_+$ represent the majority of variability in $\boldsymbol{B\delta}$. Thus, the first S_{nl} columns from $\boldsymbol{U}_+$ are taken which represent 0.995 of sum of all eigenvalues in $\boldsymbol{V}_+$. In this way, we can reduce the number of columns of $\boldsymbol{D}_{nl}$, while retaining the majority of variance.

Additionally, this allows us to add a large number of knots (and thus increasing $M-d$) without increasing the number of columns in $\boldsymbol{D}_{nl}$. The number of eigenvectors in the subset remains roughly equal past a moderate number of knots Scheipl et al. (2012). We select the number of knots as the number of unique values of $\boldsymbol{X}_j$.

In this application, we are able to calculate the spectral decomposition of the full $\boldsymbol{BP}^+\boldsymbol{B}'$. We should note that the matrix $\boldsymbol{BP}^+\boldsymbol{B}'$ is of size $N \times N$. When

N is large, this matrix might often be too large to fit in memory, such that we cannot compute the eigenvalue decomposition. As a solution to this problem, we can calculate $\boldsymbol{D}_{nl}$ on a random subset of rows in $\boldsymbol{B}$ which just fits in memory. Next, we use interpolation to obtain the design matrix for the other rows. We know the original value of the predictor that corresponds to each row in the design matrix $\boldsymbol{D}_{j,nl}$. For each column in $\boldsymbol{D}_{j,nl}$, we can use spline interpolation to obtain the design matrix for all observations of the predictor. For this, we can use a natural cubic spline, which uses linear extrapolation.

To summarize, we can represent the nonlinear part of $\boldsymbol{B}_j\boldsymbol{\delta}_j$ by using the spectral decomposition of the covariance of a Bayesian P-spline. We represent the effect of the nonlinear part as $\boldsymbol{D}_{j,nl}\boldsymbol{\beta}_{j,pen}$, with $\boldsymbol{\beta}_{j,nl}|v^2_{j,nl} \sim N(\boldsymbol{0}, v^2_{j,nl}\boldsymbol{I})$. In order to complete the decomposition of $\boldsymbol{B}_j\boldsymbol{\delta}_j$, we need to add the unpenalized part of $\boldsymbol{B}_j\boldsymbol{\delta}_j$ explicitly.

3.4.3 Design Matrix for the Unpenalized Part of a Spline

For an univariate spline with second order difference penalty, The unpenalized part of the spline corresponds to an intercept and a linear trend. Thus, we would like to include these two part into $\boldsymbol{D}_{lin}$. However, we will cause identification issues if an intercept is added for every spline in the model. To circumvent this issue, all intercepts are removed and one global intercept term η_0 is added to ensure identifiability. Thus, to model the unpenalized part of a univariate spline $f_j(\boldsymbol{X}_j)$ we only use a linear trend $\boldsymbol{D}_{j,lin}\beta_{j,lin} = \boldsymbol{X}_j\beta_{j,lin}$. For the coefficient of the linear trend $\beta_{j,lin}$, we use the prior $\beta_{j,lin}|v_{j,lin} \sim N(0, v^2_{j,lin})$.

Next to splines of numeric variables, we also add indicator variables to the model. As design matrix $\boldsymbol{D}_w$ for an indicator variable $\boldsymbol{X}_w$, only one component is used. The design matrix $\boldsymbol{D}_w$ simply corresponds to the original predictor $\boldsymbol{X}_w$, which only takes the values 1 and 0. In this way we model the effect as $\boldsymbol{D}_w\beta_w$ with $\beta_w|v_w \sim N(0, v^2_w)$.

3.4.4 Orthogonalizations

In order to improve the separability of the effects modeled with the design matrices, we orthogonalize them on design matrices that have an overlapping column space. Using the general notation $\boldsymbol{D}$ for one of the design matrices, we orthogonalize the design matrices as $(\boldsymbol{I} - \boldsymbol{Z}(\boldsymbol{Z}'\boldsymbol{Z})^{-1}\boldsymbol{Z}')\boldsymbol{D}$. Here, $\boldsymbol{Z}$ corresponds to a matrix that contains the design matrices that have overlapping column space with $\boldsymbol{D}$.

For the two design matrices of a univariate spline $\boldsymbol{D}_{nl}$ and $\boldsymbol{D}_{lin}$, we orthogonalize $\boldsymbol{D}_{nl}$ on $\boldsymbol{Z}_{nl} = [\boldsymbol{1}_N, \boldsymbol{D}_{lin}]$. Next we orthogonalize $\boldsymbol{D}_{lin}$ on $\boldsymbol{Z}_{lin} = [\boldsymbol{1}_N]$. We orthogonalize the design matrix of an indicator variable $\boldsymbol{D}_w$ on $\boldsymbol{D}_w = [\boldsymbol{1}_N]$. In this way, we center all effects around zero and remove the trend from the design matrix of the nonlinear effect. This ensures that all design matrices are orthogonal to the effects that should be estimated with a different component. This gives us the orthogonalized design matrices $\boldsymbol{D}_{j,nl}$, $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_w$. Next, these design matrices are used to obtain the design matrices for interaction effects.

3.4.5 Design Matrices for a Interaction Effects

In the model (18), we also included bivariate effects $f_{uv}(\boldsymbol{X}_u, \boldsymbol{X}_v)$. We can use the obtained components $\boldsymbol{D}_{u,lin}$, $\boldsymbol{D}_{u,nl}$, $\boldsymbol{D}_{v,lin}$ and $\boldsymbol{D}_{v,nl}$ to obtain a decomposed version of a bivariate effect of the predictors. Next to this, we can obtain a decomposed version of a interaction effect between a numerical and indicator variable $f_{uw}(\boldsymbol{X}_u, \boldsymbol{X}_w)$ by

using $\boldsymbol{D}_{u,lin}$,$\boldsymbol{D}_{u,nl}$ and $\boldsymbol{D}_{w}$. As we obtain decomposed versions of the interaction effects, we can also apply function selection to every component of the bivariate spline which was previously unpenalized in the STAR model in (17).

We model a bivariate spline for two numeric variables $\boldsymbol{X}_u$ and $\boldsymbol{X}_v$ with four components which can all be given an interpretation. For the nonlinear components of the bivariate spline we construct the design matrix $\boldsymbol{D}_{uv,nl}$ of size $N \times (S_{u,pen}S_{v,pen})$ as

$$\begin{aligned}\boldsymbol{D}_{uv,nl} = \Big[&\boldsymbol{D}_{u,nl,1} \circ \boldsymbol{D}_{v,nl,1}, \boldsymbol{D}_{u,nl,1} \circ \boldsymbol{D}_{v,nl,2}, \ldots, \\ &\boldsymbol{D}_{u,nl,S_{u,nl}-1} \circ \boldsymbol{D}_{v,nl,S_{v,nl}}, \boldsymbol{D}_{u,nl,S_{u,nl}} \circ \boldsymbol{D}_{v,nl,S_{v,nl}}\Big].\end{aligned} \tag{19}$$

Here, the $\circ$ denotes element-wise multiplication. $\boldsymbol{D}_{u,nl,s}$ denotes the s^{th} column of the $N \times S_{u,nl}$ matrix $\boldsymbol{D}_{u,nl}$. For the linear trend, we use design matrix $\boldsymbol{D}_{uv,lin} = (\boldsymbol{D}_{u,lin} \circ \boldsymbol{D}_{v,lin})$. For the two linear interaction effects, we use design matrices $\boldsymbol{D}_{uv,u} = (\boldsymbol{D}_{u,nl} \circ \boldsymbol{D}_{v,lin}\boldsymbol{1}_u')$ and $\boldsymbol{D}_{uv,v} = (\boldsymbol{D}_{u,lin}\boldsymbol{1}_v' \circ \boldsymbol{D}_{v,nl})$. Just as is the case for univariate splines, no constant effect is added to prevent identification issues.

With these design matrices, we model a bivariate spline with four components as

$$f_{uv}(\boldsymbol{X}_u, \boldsymbol{X}_v) = \boldsymbol{D}_{uv,nl}\boldsymbol{\beta}_{uv,nl} + \boldsymbol{D}_{uv,lin}\beta_{uv,lin} + \boldsymbol{D}_{uv,u}\boldsymbol{\beta}_{uv,u} + \boldsymbol{D}_{uv,v}\boldsymbol{\beta}_{uv,v}. \tag{20}$$

For all coefficients, prior distributions $\boldsymbol{\beta}_{uv,nl}|v^2_{uv,nl} \sim N(\boldsymbol{0}, v^2_{uv,nl}\boldsymbol{I})$, $\boldsymbol{\beta}_{uv,u}|v^2_{uv,u} \sim N(\boldsymbol{0}, v^2_{uv,u}\boldsymbol{I})$, $\boldsymbol{\beta}_{uv,v}|v^2_{uv,v} \sim N(\boldsymbol{0}, v^2_{uv,v}\boldsymbol{I})$ and $\beta_{uv,lin}|v^2_{uv,lin} \sim N(0, v^2_{uv,lin})$ is used, with identity matrices $\boldsymbol{I}$ of appropriate sizes. Note that all four components in (20) have their own variance parameter v^2. This will allow for separate function selection, which is described in a following section.

For an interaction between a numeric variable $\boldsymbol{X}_u$ and an indicator variable $\boldsymbol{X}_w$, only two design matrices are used. We use design matrix $\boldsymbol{D}_{uw,nl} = (\boldsymbol{D}_{u,nl} \circ \boldsymbol{D}_w\boldsymbol{1}_u')$ for the nonlinear interaction and design matrix $\boldsymbol{D}_{uw,lin} = (\boldsymbol{D}_{u,lin} \circ \boldsymbol{D}_w)$ for the linear interaction. The interaction for $f_{uw}(\boldsymbol{X}_u, \boldsymbol{X}_w)$ is modeled as

$$f_{uw}(\boldsymbol{X}_u, \boldsymbol{X}_w) = \boldsymbol{D}_{uw,nl}\boldsymbol{\beta}_{uw,nl} + \boldsymbol{D}_{uw,lin}\beta_{uw,lin}, \tag{21}$$

where we use prior distributions $\boldsymbol{\beta}_{uw,nl}|v^2_{uw,nl} \sim N(\boldsymbol{0}, v^2_{uw,nl}\boldsymbol{I})$, $\beta_{uw,lin}|v^2_{uw,lin} \sim N(\boldsymbol{0}, v^2_{uw,lin})$.

Just like the univariate splines, we project the obtain design matrices on the components to make it easier to separate the effects. For $\boldsymbol{D}_{uv,nl}$ we use

$$\boldsymbol{Z}_{uv,nl} = [\boldsymbol{1}_N, \boldsymbol{D}_{u,nl}, \boldsymbol{D}_{u,lin}, \boldsymbol{D}_{v,nl}, \boldsymbol{D}_{v,lin}].$$

For the linear interaction effects we use $Z_{uv,u} = [\boldsymbol{1}_N, \boldsymbol{D}_{u,nl}, \boldsymbol{D}_{v,lin}]$ for $\boldsymbol{D}_{uv,u}$ and $Z_{uv,v} = [\boldsymbol{1}_N, \boldsymbol{D}_{v,nl}, \boldsymbol{D}_{u,lin}]$ for $\boldsymbol{D}_{uv,v}$. For the linear trend $\boldsymbol{D}_{uv,lin}$ we use $Z_{uv,lin} = \boldsymbol{1}_N$. The interactions between an indicator variable and numeric variable are projected with $Z_{uw,nl} = [\boldsymbol{1}_N, \boldsymbol{D}_{u,nl}, \boldsymbol{D}_w]$ for $\boldsymbol{D}_{uw,nl}$ and $Z_{uw,lin} = [\boldsymbol{1}_N, \boldsymbol{D}_{u,lin}, \boldsymbol{D}_w]$ for $\boldsymbol{D}_{uw,lin}$.

To reduce the number columns of the design matrices for the interactions, we use a reduced rank representation for all of the obtained design matrices for the interactions, which we denote with $\boldsymbol{D}_{int}$. Here, the Singular Value Decomposition (SVD) of each design matrix is taken, such that we obtain $\boldsymbol{D}_{int} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'$. The first S columns from $\boldsymbol{U}$ are taken which represent 0.999 of the sum of all singular values in $\boldsymbol{\Sigma}$. Next we obtain the reduced rank approximation for each design matrix for the interactions as $\boldsymbol{D}_{int} = \boldsymbol{U}_{sel}\boldsymbol{\Sigma}^{1/2}_{sel}$ where $\boldsymbol{U}_{sel}$ denotes the selection of the first S columns in $\boldsymbol{U}$ and $\boldsymbol{\Sigma}_{sel}$ the $(S \times S)$ diagonal matrix of $\boldsymbol{\Sigma}$.

3.4.6 Rescaling the design matrices

Lastly, we scale all design matrices. In this section we denote any of the introduced design matrices as $\boldsymbol{D}_p$. Thus, this can be a design matrix such as $\boldsymbol{D}_{j,nl}$, $\boldsymbol{D}_{uv,u}$ or $\boldsymbol{D}_{uw,lin}$. With $\boldsymbol{\beta}_p$, we denote the corresponding parameter of $\boldsymbol{D}_p$. The function selection that is discussed next, depends on the size of the estimated $\boldsymbol{\beta}_p$ parameters. If the design matrices $\boldsymbol{D}_p$ vary a lot in scale for the different components, the estimated $\boldsymbol{\beta}_p$ are not a good proxy of the size of the effect $\boldsymbol{D}_p\boldsymbol{\beta}_p$.

In order to have a similar norm for each matrix $\boldsymbol{D}_p$, Scheipl (2011) propose to scale each $\boldsymbol{D}_p$ such that they have a Frobenius norm equal to 0.5. This Frobenius norm is defined as $||\boldsymbol{D}_p||_F = \sqrt{trace(\boldsymbol{D}_p'\boldsymbol{D}_p)}$. We carry out the scaling by dividing all elements in each $\boldsymbol{D}_p$ by $2||\boldsymbol{D}_p||_F$. In this way, the size of the $\boldsymbol{\beta}_p$ becomes a better proxy for the size of the effect, on which we would like to base the function selection.

This results in the two design matrices $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_{j,nl}$ to model an univariate spline for $\boldsymbol{X}_j$, design matrix $\boldsymbol{D}_w$ to model the effect of an indicator variable $\boldsymbol{X}_w$, the four design matrices $\boldsymbol{D}_{uv,nl}$, $\boldsymbol{D}_{uv,lin}$, $\boldsymbol{D}_{uv,u}$ and $\boldsymbol{D}_{uv,v}$ to model the interaction of two numeric variables $\boldsymbol{X}_u$ and $\boldsymbol{X}_v$ and two design matrices $\boldsymbol{D}_{uw,lin}$ and $\boldsymbol{D}_{uw,nl}$ to model the interaction between a numeric variable $\boldsymbol{X}_u$ and indicator variable $\boldsymbol{X}_w$. Each matrix is orthogonalized and scaled, such that they model a different effect and their corresponding $\boldsymbol{\beta}_p$ parameter can be using as a proxy of the effect size $\boldsymbol{D}_p\boldsymbol{\beta}_p$.

To give an idea of how the new design matrices relate to the original predictors, Figure 5 shows the columns from the obtained design matrices for the two components of an univariate spline $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_{j,nl}$, for three simulated predictors.

The first row shows the histograms of the three simulated predictors we consider. These simulated predictors closely match the distribution of some of the predictors that are used in the application. The first column shows a normally distributed variable. The second column shows a Box-Cox transformed variable with a minimum value of 0 which was very skewed before transformation. The third column shows a predictor which has some fixed upper and lower bound which are often observed.

The second row shows the columns of the obtained orthogonalized design matrices $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_{j,nl}$ after we apply the described methodology. Here, the red line represents the component $\boldsymbol{D}_{j,lin}$, which is the centered linear trend of the spline. The other lines represent the component $\boldsymbol{D}_{j,nl}$, for which the number columns $S_{j,nl}$ is 9,7 and 7 for the three predictors, respectively. As can be seen, the columns of $\boldsymbol{D}_{j,nl}$ closely resemble polynomials which peak at different quantiles of the data. We also observe that many functions of $\boldsymbol{D}_{j,nl}$ have roughly the same value at values of x with a lot of observed values, such as $x = 5.5$ in the middle panel, or $x = -100$ and $x = 100$ in the right panel. Ultimately, the smooth effect we obtain are a linear combination of these functions.

3.4.7 SSGAM Model Specification

With the newly obtained design matrices , we have separated each original Bayesian P-spline $\boldsymbol{B\delta}$ into multiple, orthogonal, scaled components. We decompose each univariate spline into two components, each indicator effect in one component, each interaction between two numeric variables into four components and each interaction between a numeric variable and a categorical variable into two components. For each of these components, we estimate a parameter. Now, we can rewrite the model from (17) into the specification for the SSGAM as

$$\begin{aligned}\boldsymbol{\eta} = \eta_0 + \sum_j (\boldsymbol{D}_{j,nl}\boldsymbol{\beta}_{j,nl} + \boldsymbol{D}_{j,lin}\boldsymbol{\beta}_{j,lin}) + \sum_w \boldsymbol{D}_w\beta_w + \\ \sum_{uv} (\boldsymbol{D}_{uv,nl}\boldsymbol{\beta}_{uv,nl} + \boldsymbol{D}_{uv,u}\boldsymbol{\beta}_{uv,u} + \boldsymbol{D}_{uv,v}\boldsymbol{\beta}_{uv,v} + \boldsymbol{D}_{uv,lin}\beta_{uv,lin}) + \\ \sum_{uw} (\boldsymbol{D}_{uw,nl}\boldsymbol{\beta}_{uw,nl} + \boldsymbol{D}_{uw,lin}\beta_{uw,lin}) = \eta_0 + \sum_{p=1}^{P} \boldsymbol{D}_p\boldsymbol{\beta}_p, \quad (22)\end{aligned}$$

with global intercept η_0 with an uninformative prior $\eta_0 \sim N(0, 5)$. Here, we denote the $\boldsymbol{\eta}$ as a sum of $\boldsymbol{D}_p\boldsymbol{\beta}_p$, to which we will refer as components. The $\boldsymbol{D}_p$ denotes the $N \times S_p$ design matrix for the p^{th} component. S_p varies over p, as each component has a different number of columns in $\boldsymbol{D}_p$. Here, each $\boldsymbol{D}_p$ can correspond to any of the design matrices such as $\boldsymbol{D}_{j,nl}$, $\boldsymbol{D}_{uv,u}$ or $\boldsymbol{D}_{uw,lin}$, with $\boldsymbol{\beta}_p$ as its corresponding parameter.

For each $\boldsymbol{\beta}_p$, we use prior distribution $\boldsymbol{\beta}_p|v_p^2 \sim N(\mathbf{0}, v_p^2\boldsymbol{I}_p)$, with $\boldsymbol{I}_p$ a $S_p \times S_p$ identity matrix. If we use J_{num} numeric variables and J_{cat} indicator variables and add interaction effects for all pairs of numeric variables and pairs of indicator and numeric predictors, we know that $P = 2J_{num} + J_{cat} + 4\frac{J_{num}(J_{num}-1)}{2} + 2J_{cat}J_{num}$. In order to carry out function selection for each component in the SSGAM, Scheipl et al. (2012) propose a novel multiplicative parameterization for the $\boldsymbol{\beta}_p$, which we discuss in the next section.

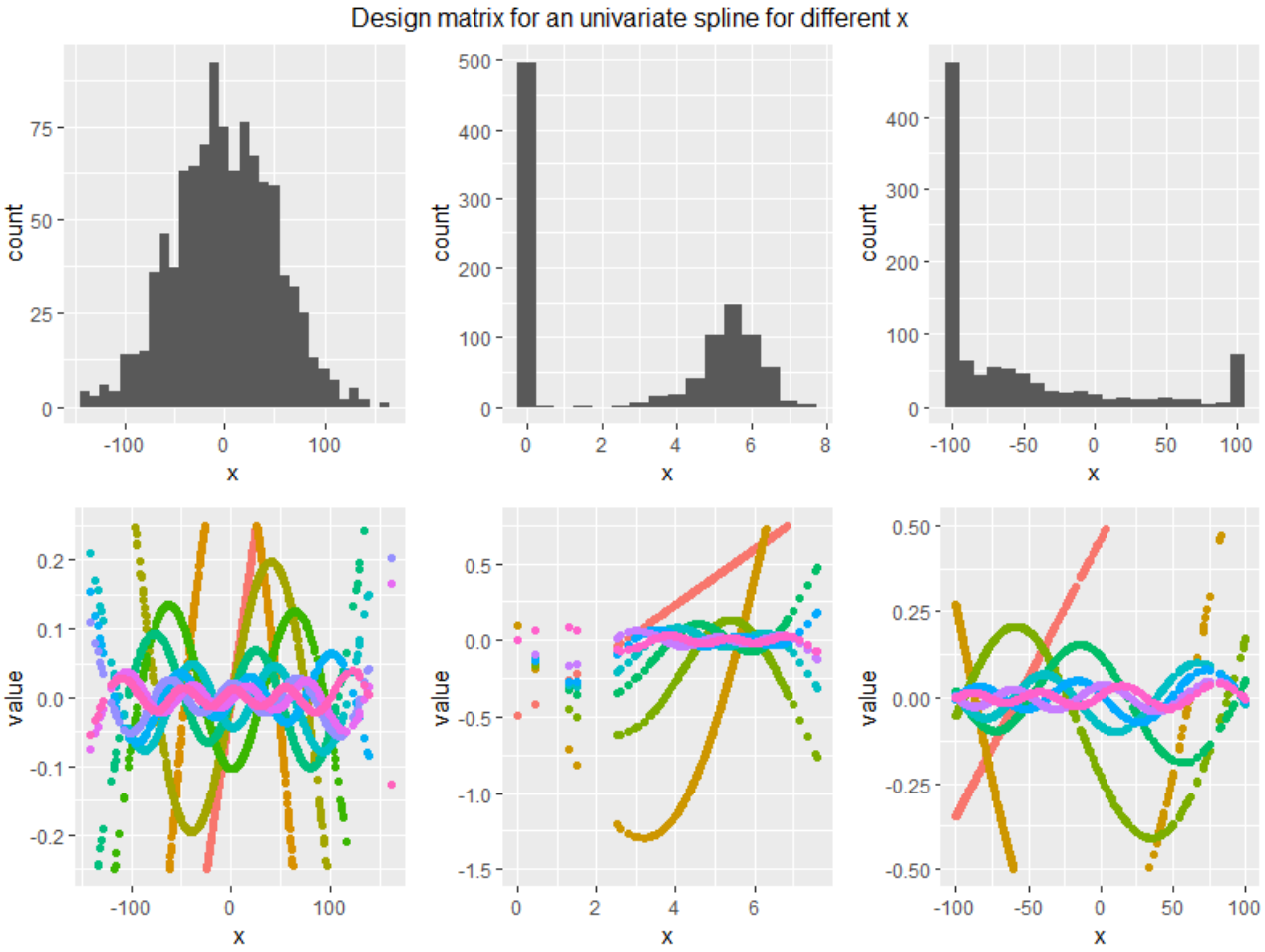

Figure 5: Design matrices $\boldsymbol{D}_{j,lin}$ and $\boldsymbol{D}_{j,nl}$ for three simulated predictors with 1000 observations. The red line in the right panel represent the design matrix of the unpenalized linear trend of the spline $\boldsymbol{D}_{j,lin}$, whereas the other lines represent the design matrix of the penalized nonlinear effect of the spline $\boldsymbol{D}_{j,nl}$.

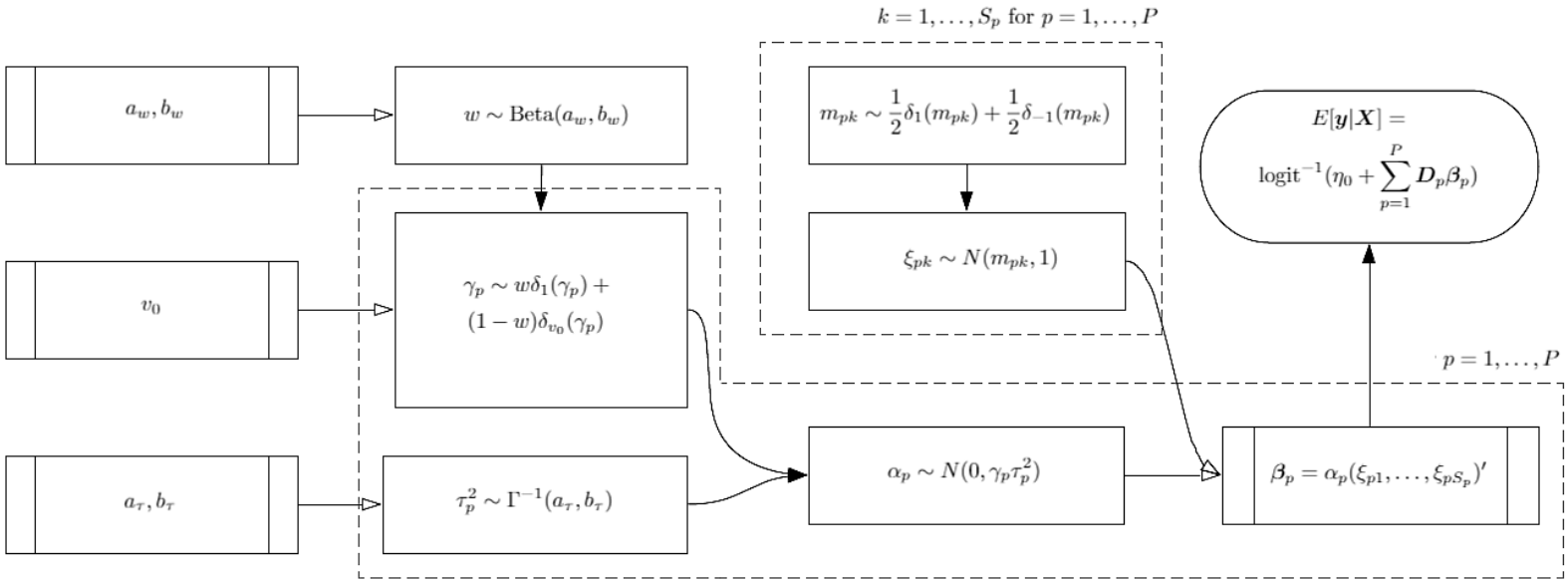

Figure 6: Schematic overview of the peNMIG prior for $\boldsymbol{\beta}_p$. An arrow with an open head corresponds to a determinstc relation, whereas an arrow with a closed head corresponds to a stochastic relation. Dotted blocks denote parameters which are sampled for all P components, or all S_p parameters for a component.

3.4.8 Function Selection

When function selection is applied, we shrink an the entire effect of a component $\boldsymbol{D}_p\boldsymbol{\beta}_p$ to zero, by shrinking all elements in the parameter vector $\boldsymbol{\beta}_p$ to zero. Only if we find that the effect of the component $\boldsymbol{D}_p\boldsymbol{\beta}_p$ is large enough, we allow for nonzero estimation of $\boldsymbol{\beta}_p$.

In order to perform function selection for each $\boldsymbol{D}_p\boldsymbol{\beta}_p$ component, Scheipl et al. (2012) propose the following multiplicative parametrization for $\boldsymbol{\beta}_p$. A graphic representation of how all parameters are related to each other is shown in Figure 6, which might be helpful for keeping an overview of the introduced parameters. In this parametrization, $\boldsymbol{\beta}_p = \alpha_p\boldsymbol{\xi}_p$, where α_p is a scalar parameter and $\boldsymbol{\xi}_p = [\xi_{p1}, \ldots, \xi_{pS_p}]'$ a vector of length S_p. In terms of interpretation, α_p can be interpreted as the global importance parameter for component p, whereas $\boldsymbol{\xi}_p$ corresponds to the direction of the effect for each of the S_p elements in $\boldsymbol{\beta}_p$.

For α_p, the global importance parameter for component p, it is assumed that

$$\alpha_p \sim N(0, v_p^2 = \gamma_p\tau_p^2), \quad \tau_p^2 \sim IG(a_\tau, b_\tau) \quad \text{and} \quad \gamma_p \sim w\delta_1(\gamma_p) + (1-w)\delta_{v_0}(\gamma_p).$$

Here a_τ and b_τ are the selected shape and scale hyperparameters for τ_p^2. $\delta_x(\gamma_p)$ denotes a Dirac delta distribution, which is a proper distribution that only has probability mass at $\gamma_p = x$. Thus, γ_p takes the value 1 with probability w and v_0 with probability $1-w$. The mixture weights w are uniformly distributed on the interval 0 to 1 as $w \sim Beta(a_w, b_w)$ with $a_w = b_w = 1$. However, other a_w and b_w could be chosen if we would have prior knowledge about how many components exert an influence on the response variable.

The variance component of α_p is obtained as $v_p^2 = \gamma_p\tau_p^2$. With the described prior distributions for τ_p^2 and γ_p, the prior distribution for v_p^2 is a bi-modal mixture of inverse gamma distributions. When $\gamma_p = v_0$, the prior variance for the importance parameter α_p equals $v_p^2 = v_0\tau_p^2$. A small v_0 such as $v_0 = 0.005$ is chosen, which results in a small v_p^2. As $\alpha_p|\gamma_p = v_0 \sim N(0, v_0\tau_p^2)$, a prior with a narrow probability mass centered around zero is obtained, also referred to as the spike. When $\gamma_p = 1$, the prior $\alpha_p|\gamma_p = 1 \sim N(0, \tau_p^2)$ is obtained. In this case, a wider prior distribution centered around zero is obtained for α_p, also referred to as the slab.

The use of these distribution is to either shrink the α_p value to zero, or have a prior distribution with probability density for a wide range of values. Figure 7 shows the prior density for $\alpha|\gamma$ for values $a_\tau = 5$, $b_\tau = 25$ and $v_0 = 0.05$ as an illustration.

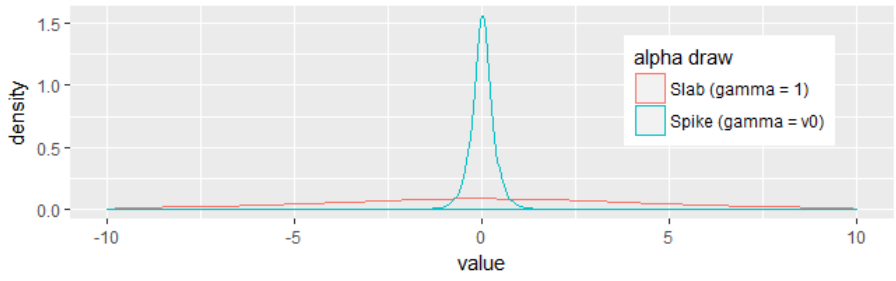

Figure 7: Draws of $\alpha_p|\gamma_p = 1$ and $\alpha_p|\gamma_p = v_0$ from its prior density for $a_\tau = 5$, $b_\tau = 25$ and $v_0 = 0.05$.

The selection of the spike (a draw of $\gamma_p = v_0$) has the function of shrinking the α_p to zero. The slab (a draw of $\gamma_p = 1$) allows for drawing values for α_p which are far from zero. In this case the prior variance equals τ_p^2. Thus, in order to obtain a wide prior distribution for $\alpha_p|\gamma_p = 1$, the τ_p^2 should be sufficiently large. We achieve this by choosing the hyperparameters for τ_p^2 such that there is little probability mass for values of τ_p^2 close to zero. Therefore, we select a larger b_τ than a_τ, such as default selected values $(a_\tau, b_\tau) = (5, 25)$.

Scheipl et al. (2012) show that after integrating out τ_p^2 and γ_p from $\alpha_p \sim N(0, v_p^2 = \tau_p^2\gamma_p)$, the distribution of $\alpha_p|w$ can be denoted as a mixture of two scaled t-distribution

$$\alpha_p|w \sim (1-w)t(df, s_0) + w\, t(df, s_1),$$

where $df = 2a_\tau$, $s_0 = \sqrt{v_0 b_\tau / a_\tau}$ and $s_1 = \sqrt{b_\tau / a_\tau}$. Thus, by using this structure of prior distributions, a spike and slab prior with t-distributions is induced for $\alpha_p|w$.

This combination of used prior distribution for α_p is referred to as a normal-mixture-of-inverse gamma (NMIG) distribution denoted as $\alpha_p \sim \text{NMIG}(\boldsymbol{\theta})$ with the set of chosen hyperparameters $\boldsymbol{\theta} = \{v_0, a_\tau, b_\tau, a_w, b_w\}$.

We obtain the final parameter vector $\boldsymbol{\beta}_p$, by multiplying the global importance parameter α_p with $\boldsymbol{\xi}_p$, which determines the direction for each of the elements in $\boldsymbol{\beta}_p$. Each element in $\boldsymbol{\xi}_p = (\xi_{p1}, \ldots, \xi_{pS_p})'$ has prior distribution

$$\xi_{pk}|m_{pk} \sim N(m_{pk}, 1), \text{ with } m_{pk} \sim \frac{1}{2}\delta_1(m_{pk}) + \frac{1}{2}\delta_{-1}(m_{pk}),$$

independently for all elements $k = 1, \ldots, S_p$ and all components $p = 1, \ldots, P$. As a large amount of the prior mass is close to 1 or -1, this parameter does not heavily influence the scale of importance for parameters in $\boldsymbol{\beta}_p$. Thus, the importance interpretation of α_p for component p is preserved. This specification also yields a marginal prior for $\boldsymbol{\beta}_p$ that is less concentrated on small absolute values compared to the case when using prior $\xi_{pk} \sim N(0, 1)$.

We complete the prior specification for $\boldsymbol{\beta}_p$ by assuming prior independence between α_p and $\boldsymbol{\xi}_p$. This results in the proposed Parameter-Expanded NMIG (peNMIG) prior distribution for $\boldsymbol{\beta}_p$, denoted by $\boldsymbol{\beta}_p \sim \text{peNMIG}(\boldsymbol{\theta})$.

To summarize, we parametrize the importance of each component $\boldsymbol{D}_p\boldsymbol{\beta}_p$ with a scalar parameter α_p. For each α_p, a spike-and-slab prior is used, such that the importance can be shrunken to zero if no real effect is found for component $\boldsymbol{D}_p\boldsymbol{\beta}_p$. Next, we obtain $\boldsymbol{\beta}_p = \alpha_p\boldsymbol{\xi}_p$, where we can interpret $\boldsymbol{\xi}_p$ as the direction of the effect for each of the columns in $\boldsymbol{D}_p$.

Alternatively, we could use $\boldsymbol{\beta}_p \sim N(\boldsymbol{0}, \gamma_p\tau_p^2\boldsymbol{I}_{S_p})$ for each $\boldsymbol{\beta}_p$, which does not introduce these extra variables α_p and $\boldsymbol{\xi}_p$. Scheipl (2011) have shown that this specification causes mixing problems, as it results in very low probabilities to jump from $\gamma = 1$ to $\gamma = v_0$ while sampling. Luckily, the proposed peNMIG prior for $\boldsymbol{\beta}_p$ remedies this problem, such that function selection can be carried out.

Scheipl et al. (2012) studied the hyperparameter sensitivity in a large simulation study and applications and concluded that the prediction accuracy is very robust for the different configurations, whereas variable selection is sensitive to selections of v_0 and (a_τ, b_τ). To investigate this sensitivity, we use four hyperparameter settings $(a_w, b_w) = (1, 1)$, $(a_\tau, b_\tau) \in \{(5, 25), (5, 50)\}$ and $v_0 \in \{0.00025, 0.005\}$.

3.4.9 Sampling Scheme

To obtain posterior draws of $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p)'$ with prior distribution $\boldsymbol{\beta}_p \sim \text{peNMIG}(\boldsymbol{\theta})$ for $p = 1, \ldots, P$, we use a blockwise Metropolis-within-Gibbs sampler. We use the MCMC sampling scheme from Brezger and Lang (2006) implemented in the `spikeSlabGAM` package in `R` (Scheipl, 2011). Algorithm 1 shows a summary of the sampling procedure, which is described in more detail below. We denote design matrices $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_P]$, binary response variable $\boldsymbol{y} = (y_1, \ldots, y_N)$ and parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_P)'$, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_P)'$, $\boldsymbol{\tau}^2 = (\tau_1^2, \ldots, \tau_P^2)'$, $\boldsymbol{\xi} = (\boldsymbol{\xi}_1', \ldots, \boldsymbol{\xi}_P')'$ and $\boldsymbol{m} = (\boldsymbol{m}_1', \ldots, \boldsymbol{m}_P')'$ with $\boldsymbol{m}_p = (m_{p1}, \ldots, m_{pS_p})'$ for $p = 1, \ldots, P$.

Algorithm 1 MCMC sampler for the peNMIG with binary responses

```
Initialize w^(0), τ^2(0), γ^(0) and m^(0) using their prior distributions.
Initialize η_0^(0), β^(0), α^(0) and ξ^(0) by using IWLS from Scheipl (2010).
for iterations m = 1, ..., M do
    for blocks for α, b = 1, ..., B_α do
        Update α_{b_b}^(m) using the MH algorithm with a P-IWLS proposal
    end for
    Update all elements in m^(t) using the full conditional distribution in (23)
    for blocks for ξ, b = 1, ..., B_ξ do
        Update ξ_{b_b}^(m) using the MH algorithm with a P-IWLS proposal
    end for
    Rescale ξ^(m) and α^(m)
    Update η_0^(m) using the MH algorithm with a P-IWLS proposal
    Update all elements in τ^2(m) using the full conditional distribution in (23)
    Update all elements in γ^(m) using the full conditional distribution in (23)
    Update w^(m) using the full conditional distribution in (23)
end for
```

We obtain initial values $w^{(0)}$, $\boldsymbol{\tau}^{2(0)}$, $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{m}^{(0)}$ by taking a draw from their prior distributions. To obtain $\eta_0^{(0)}$, $\boldsymbol{\alpha}^{(0)}$ and $\boldsymbol{\xi}^{(0)}$, starting values for $\boldsymbol{\beta}^{(0)}$ are simulated, which are then used to obtain starting values $\eta_0^{(0)}$, $\boldsymbol{\alpha}^{(0)}$ and $\boldsymbol{\xi}^{(0)}$. We obtain $\boldsymbol{\beta}^{(0)}$ by using Iteratively Weighted Least Squares (IWLS), which uses the QR-decomposition update method proposed in Scheipl (2010) described in Algorithm A.1 in the Appendix. This gives us starting values for all parameters in the model.

Next, we use the full conditional distributions to obtain draws with Gibbs sampling. For parameters w, $\boldsymbol{\tau}^2$, $\boldsymbol{\gamma}$ and $\boldsymbol{m}$, a closed form for the full conditional distribution is known. Specifically,

$$\begin{aligned} w|\cdot &\sim Beta(a_w + \sum_{p=1}^{P} \delta_1(\gamma_p), b_w + \sum_{p=1}^{P} \delta_{v_0}(\gamma_p)), \\ \tau_p^2|\cdot &\sim \Gamma^{-1}(a_\tau + S_p/2, b_\tau + \frac{\boldsymbol{\beta}_p'\boldsymbol{\beta}_p}{2\gamma_p}), \\ \frac{P[\gamma_p = 1|\cdot]}{P[\gamma_p = v_0|\cdot]} &= \sqrt{v_0} \exp\left(\frac{1 - v_0}{2v_0} \frac{\alpha_p^2}{\tau_p^2}\right), \\ P[m_{pk} = 1|\cdot] &= (1 + \exp(-2\xi_{pk}))^{-1}. \end{aligned} \tag{23}$$

The full conditional distributions for η_0, $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ do not have a known closed form expression. To draw samples from the full conditional distributions for η_0, $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$, we use a modification of the Metropolis-Hastings (MH) algorithm proposed by Brezger and Lang (2006, Section 3.1.1, scheme 1) that uses Penalized Iteratively Weighted Least Squares (P-IWLS).

This method updates the parameters by using proposals from a Gaussian approximation around the approximate mode of the full conditional distributions. For more details, see Brezger and Lang (2006, Section 3.1.1, scheme 1).

Scheipl et al. (2012) propose to use a modification of this algorithm to reduce the computational complexity. Instead of using the mode of the approximated full conditional distributions, the mean from the proposal distribution of the previous step is used instead of the posterior mode.

The parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ are not updated in one step, but are updated sequentially in blocks of 5 parameters for $\boldsymbol{\alpha}$ and 15 parameters for $\boldsymbol{\xi}$, conditional on the state of all other parameters. In this way, $\boldsymbol{\alpha}$ is divided in blocks $\boldsymbol{\alpha} = (\boldsymbol{\alpha}'_{b_1}, \ldots, \boldsymbol{\alpha}'_{b_{B_\alpha}})$ with $B_\alpha = \lceil P/5 \rceil$ blocks and $\boldsymbol{\xi}$ is divided in blocks $\boldsymbol{\xi} = (\boldsymbol{\xi}'_{b_1}, \ldots, \boldsymbol{\xi}'_{b_{B_\xi}})$ with $B_\xi = \lceil (\sum_{p=1}^{P} S_p)/15 \rceil$. This causes the Metropolis-Hastings sampler to have higher acceptance probabilities, which makes it more likely to obtain new posterior draws for the parameters. In order to solve identification issues for $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$, we rescale the drawn proposal of the parameters as $\boldsymbol{\xi}_p = \frac{S_p}{\sum_{k=1}^{S_p} |\xi_{pk}|} \boldsymbol{\xi}_p$ and $\alpha_p = \frac{\sum_{k=1}^{S_p} |\xi_{pk}|}{S_p} \alpha_p$ for $p = 1, \ldots, P$. In this way, we obtain the same $\boldsymbol{\beta}_p = \alpha_p \boldsymbol{\xi}_p$. Next to that, the posterior mode that is used in the P-IWLS update is shifted to the new approximated mode. After a large rescaling, the acceptance rate might be low, as the proposal density might no longer be well adapted to the posterior distribution (Scheipl, 2010). This rescaling also makes sure that we maintain the interpretation of α_p as importance parameter for component p and $\boldsymbol{\xi}_p$ as the parameter which determines the directions of the effects for component p.

3.4.10 SSGAM Evaluation Metrics

For making prediction with the SSGAM, we take the posterior mean of the probabilities as $\hat{y}_i = M^{-1} \sum_{m=1}^{M} (1 + e^{-\eta_i^{(m)}})^{-1}$. Here, $\boldsymbol{\eta}^{(m)} = \eta_0^{(m)} + \sum_{p=1}^{P} \boldsymbol{D}_p \boldsymbol{\beta}_p^{(m)}$ where $\eta_0^{(m)}$ and $\boldsymbol{\beta}_p^{(m)}$ denote the m^{th} posterior sample of $\eta_0$and $\boldsymbol{\beta}_p$, respectively.

In order to select which components in an SSGAM exert influence on the response variable, we can use the selection probabilities $\boldsymbol{p} = (p_1, \ldots, p_P)$. We can estimate these probabilities with the draws of $\boldsymbol{\gamma}$ as $\hat{p}_p = M^{-1} \sum_{m=1}^{M} I[\gamma_p^{(m)} = 1]$. However, we calculate the Rao-Blackwellized estimator of p_p as

$$p_p^{(m)} = P[\gamma_p = 1 | \alpha_p^{(m)}, \tau_p^{2(m)}] = 1 - \left(1 + \sqrt{v_0} \exp \left\{ \frac{1 - v_0}{2 v_0} \frac{(\alpha_p^{(m)})^2}{\tau_p^{2(m)}} \right\} \right)^{-1} \tag{24}$$

for $m = 1, \ldots, M$, such that we can estimate the inclusion probability as $\hat{p}_p = M^{-1} \sum_{m=1}^{M} p_p^{(m)}$. This estimator provides a better estimate, as we draw γ_p conditional on α_p and τ_p^2. We can make a better estimate of the inclusion probability by using α_p and τ_p^2 compared to using the discrete draws of γ_p. With these estimated inclusion probabilities $\hat{p}_p$, we can make general decisions on which components to include in a new model. Scheipl (2010) propose to include all components with $\hat{p}_p \geq 0.5$.

This rule is found to result in optimal predictive power by Barbieri and Berger (2004) for variable selection under strong conditions. These conditions are not met in the SSGAM, but simulations show that this decision rule performs well regardless.

In addition to the inclusion probability $\hat{p}_p$, we calculate a proxy for the importance of component p as $\pi_p = \bar{\boldsymbol{\eta}}_p' \bar{\boldsymbol{\eta}}_{-1} / \bar{\boldsymbol{\eta}}_{-1}' \bar{\boldsymbol{\eta}}_{-1}$, where $\bar{\boldsymbol{\eta}}_p$ corresponds to the posterior expectation of the drawn effects $\boldsymbol{\eta}_p = \boldsymbol{D}_p \boldsymbol{\beta}_p$ and $\bar{\boldsymbol{\eta}}_{-1}$ corresponds to the posterior expectation of $\boldsymbol{\eta} - \eta_0 \mathbf{1}_N$. Thus, π_p measures how much the effect of the p^{th} component correlates with the effect of all components, which is a proxy for the component importance (Gu, 1992).

To make predictions for new data, Scheipl (2011) proposes to use interpolation, such that the design matrices for the new data are not required for making predictions. Due to all the steps that are taken in the construction of the design matrices, we do not have a closed form expression to create the design matrices for new data. however, we do have the posterior of effects for each component $\boldsymbol{D}_p \boldsymbol{\beta}_p$ and the corresponding $\boldsymbol{X}$ values for the train set. We obtain the prediction for each effect by interpolating or extrapolating the results from the train set with a spline. For univariate effects, interpolation is done with a natural cubic spline. In this way, we use linear extrapolation for values outside the range of $\boldsymbol{X}$. For interaction effects, bivariate interpolation is done with the `R`-package `akima` (Akima and Gebhardt, 2016).

3.5 Evaluation Criteria

In order to assess whether the MCMC chains have converged, we inspect the traceplots of the draws as an informal preliminary check. In addition, we use more formal convergence diagnostics to test whether the chains have converged.

For the BART and DART models, drawing the samples in one long chain is recommended (Linero, 2018; Chipman et al., 2010). For a single chain of draws, the Geweke diagnostic proposed by Geweke (1991) is a suitable convergence statistic to check for convergence. Therefore, we decide to use this statistic for the DART and BART model. This statistic tests whether the mean of two parts of the posterior draws are not significantly different. A chain of n draws of parameter $\boldsymbol{\theta}$, is divided into two fractions, n_1 and n_2 which contain the first $\lceil n/10 \rceil$ draws $\boldsymbol{\theta}_1$ and last $\lceil n/2 \rceil$ draws $\boldsymbol{\theta}_2$, respectively. Next, the means of these fractions of the sample $\bar{\boldsymbol{\theta}}_1$ and $\bar{\boldsymbol{\theta}}_2$ are calculated. The variance of $\bar{\boldsymbol{\theta}}_1$ is calculated as $\hat{s}_1^2 = \frac{\hat{\omega}_1^2}{n_1}(1 + 2\sum_j^{n_1 - 1}(1 - j/n)\hat{\rho}_{1j})$, where $\hat{\omega}_1^2$ is the estimated variance of $\boldsymbol{\theta}_1$ and $\hat{\rho}_{1j}$ the estimated j^{th} order autocorrelation of $\boldsymbol{\theta}_1$. This variance is also calculated for $\bar{\boldsymbol{\theta}}_2$. The Geweke diagnostic is defined as $\frac{\bar{\boldsymbol{\theta}}_1 - \bar{\boldsymbol{\theta}}_2}{\sqrt{\hat{s}_1^2 + \hat{s}_2^2}}$ which is $N(0, 1)$ distributed for large n. For the SSGAM, we use 3 chains of MCMC draws. In order to test whether these chains have converged, we use the Gelman-Rubin convergence statistic (denoted as $\hat{R}$), proposed by Gelman and Rubin (1992), which is a more suitable pick for multiple chains of draws. This test compares the variance of the drawn parameter values across the chains.

For the SSGAM, we use 3 chains of MCMC draws. In order to test whether these chains have converged, we use the Gelman-Rubin convergence statistic (denoted as $\hat{R}$), proposed by Gelman and Rubin (1992), which is a commonly to test for convergence when multiple chains of draws are used. This test compares the variance of the drawn parameter values across the chains. $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m$ denote the parameter draws from m independent Markov chains, each with n draws. For each chain, the variance $s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{\theta}_{ij} - \bar{\boldsymbol{\theta}}_j)^2$ is estimated, where $\bar{\boldsymbol{\theta}}_j$ denotes the sample mean of $\boldsymbol{\theta}_j$.

Next, the mean within variance $W = m^{-1} \sum_{j=1}^{m} s_j^2$ and variance between means $B = \frac{n}{m-1} \sum_{j=1}^{m} (\bar{\boldsymbol{\theta}}_j - \bar{\bar{\boldsymbol{\theta}}})^2$ are calculated, where $\bar{\bar{\boldsymbol{\theta}}}$ denotes the mean of sample means $\bar{\boldsymbol{\theta}}_j$. The pooled variance of $\boldsymbol{\theta}$ is estimated as $\widehat{Var}(\boldsymbol{\theta}) = (1 - 1/n)W + (1/n)B$ and $\hat{R}$ is obtained as $\hat{R} = \sqrt{\frac{\widehat{Var}(\boldsymbol{\theta})}{W}}$. This diagnostic approaches one as the estimate of the pooled variance gets closer to the mean within variance of the m chains. In practice, one often considers parameters to have converged if they have a statistic value smaller than 1.1.

We measure the performance of the methods with the mean predictive deviance, which is defined as $\bar{D} = -\frac{2}{N} \sum_{i=1}^{N} L(y_i|\eta_i)$, where $L(y_i|\eta_i)$ denotes the log-likelihood contribution of individual i, $L(y_i|\eta_i) = y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$, where $\hat{y}_i$ denotes the predicted probability for individual i. Next to that, the Area Under the Receiver Operating Characteristic curve (AUROC) is used to measure the predictive performance of the model, which can be easily calculated using the `ROCR` package in `R` (Sing et al., 2005). The Receiver Operating Characteristic curve is measures how the true positive rate changes for different false positive rates, as the prediction threshold is varied. Next the AUROC is calculated as the area under this curve, which is a proxy for how well a model is able to discriminate between a true and false observation.

As prediction value for all Bayesian methods, we use the posterior mean of the sampled predictions. Both evaluation criteria are measured in and out of the training sample to see if the estimated model generalizes well.

4 Data Description

We use the SSGAM to model the probability the premiere episode of Designated Survivor, aired on 21 September 2016 on the ABC channel from 10:00 PM to 11:00 PM, is watched by an individual. The Nielsen Television panel consists of 58033 individuals for which TV viewing behavior is measured. As the panel is not a representative subsample of the population in the USA, Nielsen provides individual weights w_i, which corresponds to how many Americans they represent to obtain a representative subsample of the USA in terms of age and gender. In order to obtain a data set on which it is feasible to estimate the SSGAM, we draw a sample of 11606 individuals (20% of the full sample) with the probability $w_i(\sum_{i=1}^{58033} w_i)^{-1}$ to draw individual i. In this way, the drawn subsample is representative of the USA population on age and gender, such that we do not have to adjust the methods for estimation with individual weights.

Following Webster and Wakshlag (1983), we consider live television viewing to be a two-step process. First, an individual decides to watch television or do some other activity. Second, a choice for a specific program is made. In this research, the focus is to model the second step of choosing a program for an individual that already has chosen to view television. Thus, out of the subsample of 11606 individuals, we only select the individuals that have been watching television for at least one minute during the prime time block from 8:00 PM to 11:00 PM on 21 September. In this way, a subsample of $N = 8777$ individuals is obtained.

The binary response variable $\boldsymbol{y}$ is a vector of length N for which an element equals 1 if an individual watched at least one minute of the series, and 0 otherwise. $\sum_{i=1}^{N} y_i = 1437$, such that roughly 16% of the individuals have seen at least one minute of the Designated Survivor premiere. The data set contains several types of variables which describe exposure to advertising, social demographics and TV viewing behavior.

We denote ad exposures by $\boldsymbol{A}$, the $N \times K$ matrix which contains the total exposure to an ad type in minutes. We include a predictor for $K = 9$ ad types. The ad types vary in two ways. First, the ad can be on the same channel as the premiere, or on a different channel. Ads on the channel where Designated Survivor is broadcast (ABC) are denoted as an on-channel ads. Ads which are not broadcast on ABC are split in two groups. Ads that are aired on channels which are owned by the ABC network[3] are denoted as cross-channel ads. The ads that are aired on channels outside of the ABC owned channels are denoted as off-channel ads. For ABC, this distinction is of interest, as ads on their owned channels might be less expensive compared to channels they do not own. Second, ads of 6 different spot lengths are aired, which might vary in effect.

The three ads on ABC in the hour prior to the broadcast of the premiere are removed from the data set. The effects of these spots are hard to distinguish from the tendency to keep looking at the same TV channel without zapping away. Thus, these ads are removed to circumvent the possible issue of finding that these ads are extremely effective in persuading TV viewers to watch the premiere.

For simplicity, we do not take the time between exposures into account. As a predictor, we use the sum of minutes of advertising of a certain ad type since that start of the advertisement campaign one month earlier. Thus, we implicitly assume

[3]At the time of the premiere, ABC had broadcast ads on owned TV channels ABC, Freeform, ESPN, ESPN2, History, A&E network, Lifetime Television and Lifetime Movie.

that the effect of advertising on an individual does not diminish over time. Ideally, we would like to use a retention rate which describes what proportion of exposure to advertising is carried over to the next week, such that the accumulated minutes of ad exposure slowly diminish over the weeks. However, estimating the retention parameter is out of scope for this method, as we cannot add it to the SSGAM model specification with ease.

These variables are highly positively skewed, as the majority of individuals have been exposed for less than two minutes, while a huge variation in minutes is found for the rest of individuals have a large variation in minutes of ads seen. For the SSGAM, Scheipl (2010) recommend to apply a transformation on the predictors to reduce the skewness to prevent possible numerical instability. For this purpose, a Box-Cox transformation is applied.

We describe the individual characteristics with $\boldsymbol{Z}$, the $N \times L$ matrix which contains a combination of 10 numerical variables and categorical variables. For categorical variables, we include $C-1$ dummy variables for a categorical variable with C categories, removing the largest category for identification. After creating dummy variables for the categorical variables, $L = 40$ columns are obtained.

We describe past viewing behavior variables with $\boldsymbol{V}$, the $N \times Q$ matrix which contains $Q = 153$ numerical variables. We describe the past viewing behavior by using the viewing duration from the previous week in hours[4]. For all channels, dayparts and genres that the Nielsen panel uses to categorize TV viewing, we construct a numerical variable that describes the number of hours that a specific kind of TV program is viewed in the previous week. For example, a predictor is created for the number of hours an individual watched FOX, watched during the "early fringe" daypart, or watched the genre "general drama" in the previous week. A genre or channel is removed from the data set if more than 95% of the individuals watched it for zero minutes. As these variables are also highly positively skewed, we apply a Box-Cox transformation as well.

This results in the data set $\boldsymbol{X} = [\boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{V}]$ of size $N \times (K+L+Q)$ (8780×203) on which we perform variable selection. By using DART to model $\boldsymbol{y}$ with $\boldsymbol{X}$, a new data set of selected variables is obtained. The data set $\tilde{\boldsymbol{X}}$ is obtained which can be further split down into $[\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{V}}]$ with sizes $N \times K_{sel}$, $N \times L_{sel}$ and $N \times Q_{sel}$, respectively.

We use the Friedman H-statistic to find which of the variables in $\boldsymbol{W} = [\tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{V}}]$ have an interaction effect with each of the selected advertising variables $\tilde{\boldsymbol{A}}_k$, which denotes the k^{th} column in $\tilde{\boldsymbol{A}}$. For each $\tilde{\boldsymbol{A}}_k$, the three variables with the largest H statistic are selected, which we denote as the $N \times 3$ matrix $\tilde{\boldsymbol{W}}_k$, for $k = 1, \ldots, K_{sel}$. We include interaction effects between $\tilde{\boldsymbol{W}}_{ks}$ and $\tilde{\boldsymbol{A}}_k$, for $s = 1, 2, 3$, where $\tilde{\boldsymbol{W}}_{ks}$ denotes the s^{th} column in $\tilde{\boldsymbol{W}}_k$.

Denoted as a sum of nonlinear functions as in (18), the model corresponds to

$$E[\boldsymbol{y}|\boldsymbol{\eta}] = \left(1 + e^{-\boldsymbol{\eta}}\right)^{-1} \text{ with}$$
$$\boldsymbol{\eta} = \sum_{k=1}^{K_{sel}} f_A(\tilde{\boldsymbol{A}}_k) + \sum_{l=1}^{L_{sel}} f_l(\tilde{\boldsymbol{Z}}_l) + \sum_{q=1}^{Q_{sel}} f_q(\tilde{\boldsymbol{V}}_q) + \sum_{k=1}^{K_{sel}} \sum_{s=1}^{3} f_{ks}(\tilde{\boldsymbol{A}}_k, \tilde{\boldsymbol{W}}_{ks}). \quad (25)$$

[4]The number of hours of watched TV in a week can be large due to multiple televisions that are turned on at the same time. In addition, we are aware that the data from the Nielsen Panel may contain inaccurate observations due to incorrect recording of viewing sessions. Nevertheless, the obtained quantities are the best available proxy for measuring TV viewing behavior.

After creating the design matrices for all components we obtain

$$\boldsymbol{\eta} = \eta_0 + \sum_{p=1}^{P} \boldsymbol{D}_p \boldsymbol{\beta}_p \tag{26}$$

with the constructed design matrices $\boldsymbol{D}_p$ and each $\boldsymbol{\beta}_p \sim \text{peNMIG}(\boldsymbol{\theta})$. We use the described SSGAM methodology to sample the posteriors of the parameters.

For evaluating the out-of-sample performance of the models, a random subsample of 2195 individuals (25%) is selected as test sample. We use the remaining 6582 individuals (75%) to train the models on.

4.1 Models for Comparison

We evaluate both the variable selection and model fit. As both tasks require different models, the evaluation is done separately. For both these modeling tasks, we use a variety of models and hyperparameter settings and compare the results.

Variable selection is done by using both BART proposed by Chipman et al. (2010) and DART proposed by Linero (2018). For BART, the hyperparameters are varied with $T \in \{50, 200\}$ and $\lambda \in \{1, 2\}$. For DART, the hyperparameters are varied with $T \in \{50, 200\}$, $\lambda \in \{1, 2\}$ and $\rho \in \{20, 50, 203\}$. The performance of the variable selection is based on the number of variables that are selected to exert influence on the response variable. For the modeling, we use one of the selected sets of predictors. We base this selection on the number of predictors selected and the average number of splits in a tree.

The probability of watching the premiere of Designated Survivor is modeled with six models. The main model that we described in the previous section uses the set of variables selected by DART, adds interaction between $\tilde{\boldsymbol{A}}_k$ and $\tilde{\boldsymbol{W}}_k$ for $k = 1, \ldots, K_{sel}$ and uses an SSGAM to estimate smooth effects of the predictors. we abbreviate this model as DARTSSGAM. The hyperparameters for this method are varied with $(a_\tau, b_\tau) \in \{(5, 25), (5, 50)\}$ and $v_0 \in \{0.005, 0.00025\}$. In order to compare the performance of this model, we use the following parametric and nonparametric models.

The parametric benchmark is a probit model, estimated with maximum likelihood estimation. With this method, we can visualize the relations between the predictors and response variable, but have to choose a functional form of the model a priori. As the logit link function is nonlinear, interaction effects implicitly exist for the marginal effects of the predictors. We us two different sets of predictors. One of the data sets uses all predictors, whereas the other method makes use of the predictors selected by DART.

The first data set contains all 9 transformed advertising variables $\boldsymbol{A}$ and the square root of the predictors in $\boldsymbol{A}$ to allow for more nonlinearity in advertising effectiveness. Furthermore, all 40 demographic variables $\boldsymbol{Z}$ and a subset of the principle components of $\boldsymbol{V}$ are used as predictors.

As $\boldsymbol{V}$ consist of 153 numeric predictors, we can use Principal Component Analysis (PCA) to reduce the number of variables used in this model, while retaining the majority of variance in the data. In order to determine how many Principal Components (PCs) should be included, we use a permutation test. In this permutation test, we compare the distribution of eigenvalues from the Singular Value Decomposition (SVD) of $\boldsymbol{V}$ with the eigenvalues of those of $\boldsymbol{V}_{perm}$. This matrix $\boldsymbol{V}_{perm}$ is obtained

by randomly permuting the columns in $\boldsymbol{V}$, such that the multivariate relations are removed, but the univariate properties of the columns in $\boldsymbol{V}_{perm}$ remain the same. Next we generate a confidence interval for the eigenvalues of the SVD of $\boldsymbol{V}$ by calculating the distribution of eigenvalues for multiple $\boldsymbol{V}_{perm}$. This will show whether the eigenvalues from the SVD of $\boldsymbol{V}$ are significantly different from those of randomly permuted matrices $\boldsymbol{V}_{perm}$, and thus should be included as predictor. To determine the confidence interval, we use 1000 randomly permuted matrices $\boldsymbol{V}_{perm}$ and the 0.99^{th} percentile of eigenvalues of the permuted matrices is used as critical value.

From the literature, we know that the variable which denotes the number of seconds an individual watched the preceding program is of great importance. For that reason, we decide to add it to the model linearly and not include it in the matrix $\boldsymbol{V}$ that is decomposed into PCs. It would be unfair not to include this variable explicitly, given its found importance in previous research. We abbreviate this probit model as PCAProbit.

The second data set we use for the probit model contains all variables that are found to be important by the DART variable selection, $\tilde{\boldsymbol{X}}$. In addition to these variables, we add the square roots of $\tilde{\boldsymbol{A}}$ to allow for more nonlinearity in advertising effectiveness. We abbreviate this model as DARTProbit.

As nonparametric benchmark model, we use the BART and DART models that were used in the variable selection step. These models also make predictions of the probability of watching the premiere of Designated Survivor. The same hyperparameter configurations are used, such that re-estimating the models is not necessary. Visualization of the effects of the predictors is more difficult, as we would need to use additional methods such as Partial Dependency Functions to gain insight into how the predictors relate to the response variable. BART is expected to perform very well on predictive performance, as Chipman et al. (2010) show that it performs competitively for prediction a binary response variable. The predictive performance of DART is expected to be lower than BART, given that it is designed for variable selection and not necessarily for prediction. Due to its sparsity inducing Dirichlet prior for variable selection probabilities in the trees, this method is expected to select a lower number of variables for its trees compared to BART, which might result in lower predictive performance. We abbreviate this model as BART and DART.

Finally, we use another models that uses the SSGAM methodology. This SSGAM uses the full set of variables $\boldsymbol{X}$ and adds interactions between $\boldsymbol{A}$ and all $\boldsymbol{Z}$. We do not add the interactions between all $\boldsymbol{A}$ and $\boldsymbol{V}$, as this would result in adding more than 1500 interactions, which is too heavily computationally. As the spike-and-slab priors enable regularized estimation, it is theoretically possible to use a large data set. However, after running for 65 hours on a modern laptop (Intel i7-4600U processors with 2.69GHz) without finishing, we aborted the sampling. Thus, using a data set with such a large number of variables really seems too heavy computationally for this research.

With this selection of models and evaluation criteria, we can compare the models in three ways. First, we evaluate the predictive performance. Second, we compare the insights that a model provides in terms of relations between response variable and predictors. Third, we compare the models in terms of practical use, as some models might need a lot of computation time or hyperparameter tuning.

5 Results

This section contains the results from the applied methods. First, we describe the results from the variable selection and interaction detection. Second, we present the output from the SSGAM by showing the relations between predictors and response variable that have been found in the data. Third, we compare the performance of all used models.

5.1 Variable Selection

For all BART and DART models we sample the number of draws that is recommended by Linero (2018). We draw 10000 samples, where the first 5000 are discarded as burn-in period. We use a thinning factor of 5, such that we obtain a posterior sample of 1000 draws. To check whether the chain has converged for hyperparameter setup $T = 200$, $\lambda = 1$ and $\rho = 20$, we compare the output to the output of a DART model with more draws. This DART model to check for convergence uses 35000 draws, where the first 25000 draws are discarded as burn-in. We use a thinning factor of 10, such that we also obtain 1000 posterior draws.

As can be seen in Figure 8 both samplers converged to a similar range of values for the average number of splits in a tree in the DART model with $T = 200$, $\lambda = 1$ and $\rho = 20$. Thus, using 5000 burn-in draws seems adequate in order to reach convergence for the size of the individual trees in the ensemble. The Geweke Diagnostic for the average number of splits is -0.128, which also shows that convergence is likely.

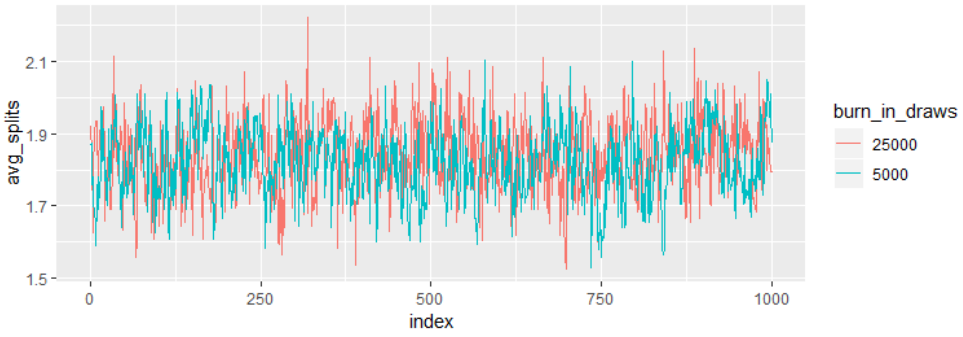

Figure 8: Draws of the average number of splits in a tree for the DART model with $T = 200$, $\lambda = 1$ and $\rho = 20$. A sampler with 5000 burn-in draws and a thinning factor of 5 is compared to the draws of a sampler with 25000 burn-in draws and a thinning factor of 10.

Figure A.1 in the Appendix shows the distribution of Geweke Diagnostics of splitting probabilities, which is not significantly different from zero for 183 out of the 203 variables, using $\alpha = 0.05$ as significance level. Thus, we find multiple splitting probabilities where the posterior mean at the start of the sample is significantly different from the posterior mean at the end of the sample. For the DART model to check for convergence, 178 out of the 203 variables have drawn parameters with a Geweke diagnostics that are not significantly different from 0. Thus, increasing the number of draws in the burn-in sample, or increasing the thinning factor does not cause the

posterior to have a more stable mean. When inspecting the traceplots for variables with a large Geweke diagnostic, we see that the chain can remain stuck in a posterior mode. This can be seen in Figure A.2 in the Appendix, which shows the traceplot for one of the predictors with a Geweke diagnostic of -2.18. Linero (2018) describe that the Markov chain can become stuck in a mode of the posterior when highly correlated predictors are used, which is the case in this data set. They also show that the performance of the variable selection of DART remains good, even when the posterior is multimodal. Thus, we decide to use the drawn posterior samples as they are.

Table 1 shows the performance of the DART and BART models for all considered hyperparameter configurations $T \in \{50, 200\}$, $\lambda \in \{1, 2\}$ for BART, and also $\rho \in \{20, 50, 203\}$ for DART. The other hyperparameters are fixed at the values $\gamma = 0.95$, $a_\alpha = 0.5$, $b_\alpha = 1$ and $k = 2$, recommended by Chipman et al. (2010). A predictor is selected if it is used to split on in at least 50% of the ensembles.

First, we investigate the effect of the number of trees in the ensemble. For BART, increasing T causes more different predictors to be used for splitting. As BART does not assume a sparsity inducing prior for the splitting probabilities, more different variables are used to split on in the ensemble when T is increased. As the BART models with $T = 200$ select the majority of the variables at least once, the use of BART does not seem to be a good pick for the purpose of variable selection when we use this rule for selecting predictors.

For DART, the difference between $T = 50$ and $T = 200$ is less pronounced. In 4 out of 6 cases, fewer variables get selected for a larger number of trees. However, the difference between the number of selected variables is always smaller than four. Thus, when a different number of trees is used, the Dirichlet prior keeps the number of selected predictors relatively stable compared to BART.

Second, we investigate the effect of the hyperparameter λ for the prior on the tree depth. When $\lambda = 1$, the average number of splits in each individual tree is higher compared to the models with $\lambda = 2$. Thus, the trees with $\lambda = 1$ are deeper in general. For BART, slightly more different predictors are selected when the trees are deeper. For DART, the number of variables selected stays the same or decreases in 5 out of 6 cases when we use $\lambda = 1$. Thus, this implies that trees with $\lambda = 1$ add more interactions between the selected predictors, or make multiple splits on the selected predictors in each individual tree. Thus, if interaction effects are large enough, this hyperparameter setting for DART seems more suitable for selecting predictors for which interaction effects are found. We do not have a metric to measure the interaction importance with the splitting probabilities, but this observation shows the potential of using DART for interaction detection.

For DART, we are able to set ρ, which reflects our initial guess of the number of variables that exert influence on the response variable. As expected, the number of selected predictors increases with ρ in 6 out of 8 cases. The largest shift is found when we increase ρ from 50 to 203. However, this only causes 6 additional predictors to be selected in the most extreme case. we conclude that the number of variables selected is only slightly sensitive to ρ, as only a small increase is found for larger ρ.

When comparing the evaluation metrics, it becomes clear that the BART models perform slightly better than the DART models. The BART models especially perform better in-sample, as the worst BART is still better than the best DART. However, when comparing the out-of-sample performance, a slightly lower, but more stable performance is found for DART. In general, the performance of all DART models is very stable across the hyperparameter settings.

As data set for the next step, the subset of the 11 predictor selected by the DART model with $T = 200$, $\lambda = 1$ and $\rho = 20$ is chosen. We choose this model because the goal of this variable selection step is to reduce the size of the data set such that estimation with SSGAM is feasible. As the performance of the DART models is comparable, we prefer picking a model which selects the lowest number of predictors and a higher number of average splits.

The subsets of selected predictors for all DART models are similar. Table A.1 in the Appendix shows how many DART models selected each predictor. The three variables ctrl_duration_leadin, sd_age and vb_channel_nbc are selected in every DART model. Surprisingly, only two of the advertising variables are selected in more than half of the DART models. In the model we selected, the on-channel advertising is included in the subset of predictors.

The DART models that select a larger number of predictors often select predictors that are not selected by other DART models. This puts doubt on how large the influence of those predictors is. The predictors that are included in the DART model we decide to use are often included in many other DART models. Thus, it seems reasonable that this is a subset which contains many influential predictors. We show a description of the selected predictors in Table A.2 in the Appendix. The selection of this subset of predictors results in data set $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{V}}]$ of size $N \times (K_{sel} + L_{sel} + Q_{sel})$ with $K_{sel} = 1$, $L_{sel} = 5$ and $Q_{sel} = 5$.

Table 1: Performance of the DART and BART models for used hyperparameter settings. Best performing BART and DART models in bold.

BART												
T					50	50	200	200				
λ					2	1	2	1				
Predictors selected					25	28	154	160				
Average number of splits					1.18	1.34	1.05	1.13				
AUROC (in)					0.851	0.849	0.858	**0.858**				
AUROC (out)					0.802	0.805	0.809	**0.813**				
$\bar{D}$ (in)					0.654	0.653	0.645	**0.641**				
$\bar{D}$ (out)					0.679	0.675	0.671	**0.666**				
DART												
T	50	50	200	200	50	50	200	200	50	50	200	200
λ	2	1	2	1	2	1	2	1	2	1	2	1
ρ	20	20	20	20	50	50	50	50	203	203	203	203
Predictors selected	13	15	11	11	13	12	13	12	18	18	17	14
Average number of splits	1.33	1.75	1.26	1.82	1.24	1.7	1.26	1.83	1.26	1.6	1.28	1.81
AUROC (in)	0.839	0.840	0.837	0.833	0.839	0.841	0.84	0.839	0.842	**0.845**	0.841	0.842
AUROC (out)	**0.810**	0.808	0.806	0.806	0.806	0.805	0.806	0.806	0.805	0.806	0.806	0.807
$\bar{D}$ (in)	0.662	0.659	0.666	0.670	0.663	0.656	0.664	0.661	0.655	**0.655**	0.662	0.657
$\bar{D}$ (out)	**0.668**	**0.668**	0.671	0.670	0.673	0.672	0.672	0.671	0.671	0.671	0.673	0.672

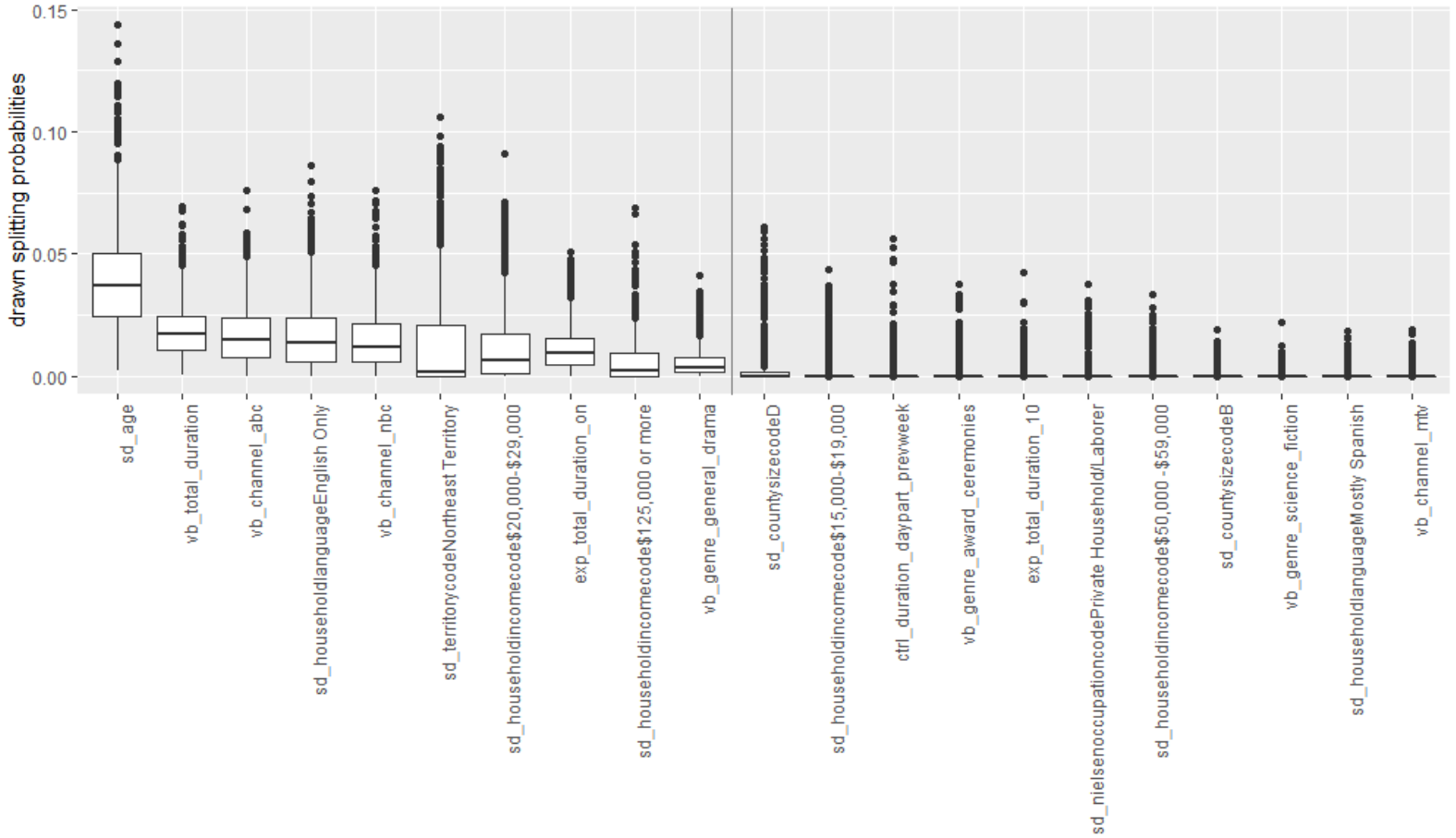

Figure 9: Boxplots of the drawn posterior splitting probabilities for the predictors that have been selected at least once to form a split on in 5% of the ensembles, ordered on the posterior means. posterior splitting probabilities for ctrl_duration_leadin with a posterior mean of 0.82 removed for a clearer overview. All predictors to the left of the grey line are selected, as they are used for splitting in at least 50% of the ensembles. Results shown for DART with $T = 200$, $\rho = 20$ and $\lambda = 1$.

Figure 9 shows the posterior draws of the splitting probabilities for all predictors that are added in at least 5% of the ensembles for the DART with $T = 200$, $\lambda = 1$ and $\rho = 20$. These posteriors give us information on the relative importance of the selected predictors. We removed the posterior of ctrl_duration_leadin from the figure, which had a much higher posterior mean of 0.82 to improve the readability. These high splitting probabilities for ctrl_duration_leadin already show that this variable was the most important by far, as the posterior mean of the splitting probability of the second most important variable sd_age is more than twenty times lower. As expected, the posterior means of the splitting probabilities of the selected variables are higher than those of the variables that are not selected.

For the predictors which are not selected, the splitting probability in the posterior regularly jumps between zero and a nonzero splitting probability. However, the majority of draws of the splitting probability is very close to zero. For the set of selected predictors, nonzero splitting probabilities are sampled more frequently, such that the median is not so close to zero. Here, we can see that the sparsity inducing Dirichlet prior only allows a small number of predictors to have a nonzero splitting probability, while the splitting probability for the majority of the predictors is close to zero.

To find the variables that interact with the selected advertising variable, we use a Random Forest to model $\boldsymbol{y}$ with $\tilde{\boldsymbol{X}}$. Next, partial dependency plots are calculated, such that we can calculate Friedman's H statistic to quantify the importance of the interactions between the minutes of on-channel ads seen $\tilde{\boldsymbol{A}}$ and the selected predictors $[\tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{V}}]$. We obtain the H statistics shown in Figure A.3 in the Appendix. The interactions with the three largest H statistics are added to obtain $\tilde{\boldsymbol{W}}_1$, which corresponds to predictors that interact with on-channel advertising most clearly in the Random Forest. We add the interactions between on-channel advertising and total viewing hours, total viewing hours on the ABC TV channel and the age of the respondent.

Next, we use the set of selected predictors and interactions as input for the SSGAM, which allows us to visualize how the predictors effect the probability of watching the premiere episode of Designated Survivor.

5.2 SSGAM

After creating design matrices for the selected predictors and interactions, the SSGAM uses $P = 30$ components, with $S = \sum_{p=1}^{P} S_p = 182$ parameters for all components combined. This corresponds with parameter vectors $\boldsymbol{\alpha}$,$\boldsymbol{\gamma}$ and $\boldsymbol{\tau}^2$ of length P and vectors $\boldsymbol{\beta}$, $\boldsymbol{\xi}$ and $\boldsymbol{m}$ of length S. First we discuss the convergence diagnostics. Second, we describe the which components are found to be important. Third, we visualize the estimated smooth effects.

5.2.1 Convergence Diagnostics

For the SSGAM, we draw 3 chains of 26000 samples in parallel in roughly 3 hours on a modern laptop (Intel i7-4600U 2.68GHz). The first 1000 draws are discarded as burn-in period and a thinning values of 5 is used, as is recommended by Scheipl et al. (2012). With these settings, we obtain 15000 posterior draws.

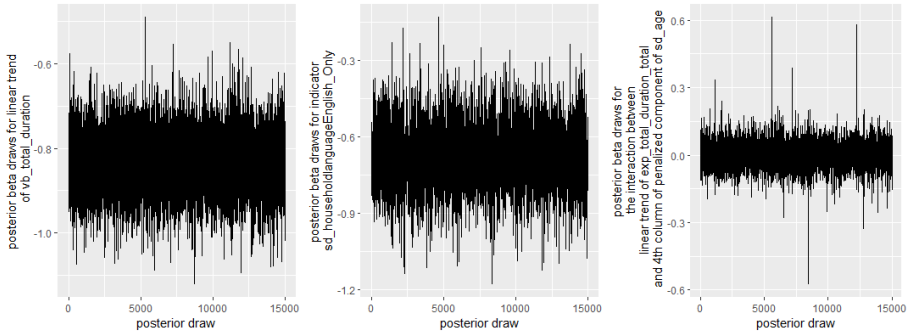

Figure 10: Traceplot of three randomly selected β_{pk} from the posterior sample of $\boldsymbol{\beta}$ from the SSGAM with hyperparameters $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$.

As an informal check, we inspect the traceplots of the elements in the drawn $\boldsymbol{\beta}$. Figure 10 shows this for a random selection of three drawn β_{pk} for hyperparameter setting $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$, which evolve around a single mode. As a more formal check to investigate if all the chains have converged, we use the $\hat{R}$. This statistic compares the posterior draws of $\boldsymbol{\beta}$ from the three MCMC chains. The $\hat{R}$ values are shown in Figure 11, which are very close to 1 for all β_{pk} parameters, which shows that it is likely that the chains converged.

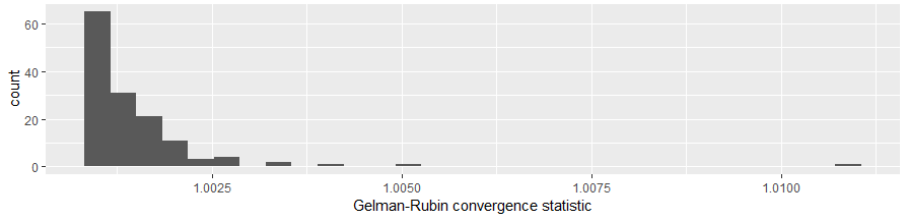

Figure 11: $\hat{R}$ values for β_{pk} for $k = 1, \ldots, S_p$ and $p = 1, \ldots, P$ from the SSGAM model with hyperparameters $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$.

For the hyperparameter settings with $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 25)$, the $\hat{R}$ values of the β_{pk} parameters for the components that model the interaction effects are often higher than 1.1, which indicates that the chain might not have converged. Figure 12 shows the traceplot of posterior draws of one of the β_{kp} with a $\hat{R}$ of 1.4, where the output of the three chains is concatenated. As we can see, the chains switch regularly from drawing $\gamma_p = 1$ to $\gamma_p = v_0$, which cause this chain of draws to be bi-modal. As the $\hat{R}$ diagnostic uses the variance of parameter draws within a chain, it is not a surprise that we conclude that the parameters did not converge with this statistic. This also seems to be the case for $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 50)$. For these two hyperparameter settings, we doubled the number of burn-in draws to see whether this could solve this problem, but it still kept occurring. Thus, when the amount of shrinkage is increased by picking $v_0 = 0.00025$, the sampler seems to have more difficulty with obtaining chains that converge to one posterior mode.

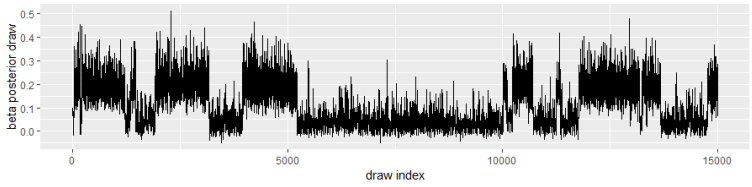

Figure 12: Posterior draws of β_{kp} for the first column of the design matrix that models the interaction between the penalized component hours of general drama seen and linear component of on-channel advertising. Samples taken for the SSGAM model with hyperparameter $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 25)$.

The predictive performance of the SSGAMs for the different hyperparameter configurations is shown in Table 2. We find that almost no difference in predictive performance is found across the hyperparameter settings. Thus, the bimodal posterior distributions that are obtained by picking $v_0 = 0.00025$ do not seem to be a problem in terms of predictive performance. Nevertheless, we choose to display the results of $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$ in all following results, such that we are sure we are looking at posteriors which are more likely to have converged.

Table 2: Performance of SSGAM models for the considered hyperparameter settings

v_0	0.00025	0.005	0.00025	0.005
(a_τ, b_τ)	(5,25)	(5,25)	(5,50)	(5,50)
AUROC (in)	0.826	0.828	0.827	0.828
AUROC (out)	0.805	0.805	0.805	0.805
$\bar{D}$ (in)	0.687	0.686	0.685	0.686
$\bar{D}$ (out)	0.679	0.679	0.679	0.679

5.2.2 Importance of Components

Table 3 shows the combinations of components that often have a high inclusion probability. It shows the 6 most common combinations of components with $P[\gamma_p = 1] \geq 0.5$ with corresponding proportions of draws with this combination. Each row in this table denotes one of the components $\boldsymbol{D}_p\boldsymbol{\beta}_p$ for $p = 1, \ldots, P$. Each component in the table

corresponds to a component in (22). Recall that each univariate spline is decomposed in two components, and each bivariate spline is decomposed in four components. With $\texttt{lin}(\cdot)$, we denote the centered linear trend represented by $\boldsymbol{D}_{j,lin}$. With $\texttt{nonlin}(\cdot)$ we denote the nonlinear component represented by the design matrix $\boldsymbol{D}_{j,nl}$. With $\texttt{ind}(\cdot)$, we denote an indicator variable represented by the design matrix $\boldsymbol{D}_w$. For a bivariate spline, each component is denoted as an interaction between two components with the $\times$ symbol. For example, $\texttt{lin}(\boldsymbol{X}_u)\times\ \texttt{nonlin}(\boldsymbol{X}_v)$ denotes an effect which is estimated with design matrix $\boldsymbol{D}_{uv,v}$ and $\texttt{nonlin}(\boldsymbol{X}_u)\times\ \texttt{nonlin}(\boldsymbol{X}_v)$ denotes an effect which is estimated with design matrix $\boldsymbol{D}_{uv,nl}$ when relating the components to the definitions used in (22).

Table 3: The 6 most common combinations of functions with $P[\gamma_p = 1] \geq 0.5$ and corresponding proportions, mean inclusion probabilities $\hat{p}_p$, influence of a component π_p, and the number of columns in the design matrix S_p for all components in the SSGAM with $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$.

	Combination of components (proportion)								
	1	2	3	4	5	6			
Component	(0.061)	(0.046)	(0.035)	(0.028)	(0.024)	(0.021)	$\hat{p}_p$	π_p	S_p
lin(ctrl_duration_leadin)	x	x	x	x	x	x	1.000	0.307	1
nonlin(ctrl_duration_leadin)	x	x	x	x	x	x	1.000	0.078	7
lin(vb_total_duration_transf)	x	x	x	x	x	x	0.972	0.003	1
nonlin(vb_total_duration_transf)							0.037	0.000	9
lin(vb_genre_general_drama_transf)	x		x	x	x	x	0.646	0.016	1
nonlin(vb_genre_general_drama_transf)							0.058	0.012	8
lin(vb_channel_abc_transf)							0.352	0.085	1
nonlin(vb_channel_abc_transf)			x		x		0.427	0.018	7
lin(vb_channel_nbc_transf)	x	x	x	x	x	x	0.803	0.102	1
nonlin(vb_channel_nbc_transf)							0.041	0.002	8
lin(sd_age)	x	x	x	x	x	x	0.825	0.093	1
nonlin(sd_age)	x	x	x	x	x	x	0.935	0.022	9
ind(sd_householdlanguageEnglish_Only)							0.328	0.040	1
ind(sd_territorycodeNortheast_Territory)							0.088	0.000	1
ind(sd_householdincomecode125000_or_more)							0.158	0.002	1
ind(sd_householdincomecode20000to29000)				x	x		0.421	0.027	1
lin(exp_total_duration_on_transf)	x	x	x	x	x	x	0.932	0.182	1
nonlin(exp_total_duration_on_transf)						x	0.354	0.017	7
lin(sd_age) ×lin(exp_total_duration_on_transf)							0.036	0.000	1
nonlin(sd_age) ×lin(exp_total_duration_on_transf)							0.036	0.000	8
lin(vb_total_duration_transf) ×lin(exp_total_duration_on_transf)							0.036	0.000	1
nonlin(vb_total_duration_transf) ×lin(exp_total_duration_on_transf)							0.037	0.000	8
lin(vb_channel_abc_transf) ×lin(exp_total_duration_on_transf)							0.083	0.005	1
nonlin(vb_channel_abc_transf) ×lin(exp_total_duration_on_transf)							0.038	0.002	6
lin(sd_age) ×nonlin(exp_total_duration_on_transf)							0.047	-0.002	6
nonlin(sd_age) ×nonlin(exp_total_duration_on_transf)							0.036	0.000	26
lin(vb_total_duration_transf) ×nonlin(exp_total_duration_on_transf)							0.037	-0.002	6
nonlin(vb_total_duration_transf) ×nonlin(exp_total_duration_on_transf)							0.036	0.000	27
lin(vb_channel_abc_transf) ×nonlin(exp_total_duration_on_transf)							0.051	-0.009	6
nonlin(vb_channel_abc_transf) ×nonlin(exp_total_duration_on_transf)							0.038	0.000	21

We see that both components for total minutes seen of the previous program (ctrl_duration_lead_in) and age of the respondent (sd_age) have high inclusion probabilities. Especially viewership of the previous program contributes a large part to the predictions given its large π_p values. This large effect confirms the earlier findings of Rust et al. (1992) that it is important to include a predictor on viewer-

ship of the preceding program. The posterior of the splitting probabilities in DART also showed that these two predictors are of large importance. We also find relatively high inclusion probabilities for the two components of exposure on channel ads (exp_total_duration_on_transf). Thus, this type of advertising seems to have an important influence on the viewing probability. However, none of the interaction components with advertising have a high inclusion probability. We conclude that the interaction effects between on-channel advertising and having a certain age or watching a lot of television are small.

Other important components are the linear components of previous week's viewing hours in total (vb_total_duration_transf), viewing hours of the genre general drama (vb_genre_general_drama_transf) and viewing hours of the NBC channel (vb_channel_nbc_transf). To a lesser extent, we find that both components of the viewing hours of the ABC channel from the previous week (vb_channel_abc_transf) have a high inclusion probability. We observe that the method does not necessarily have high inclusion probabilities for both the linear and nonlinear components, but is able to select the one or the other.

Next to age, other demographic variables which have a relatively high inclusion probability is the indicator for a household that only speaks English (sd_householdlanguageEnglish_Only) and the indicator for a household with an income between \$20,000 and \$29,000 (sd_householdincomecode20000to29000).

For hyperparameter setting $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 25)$ the obtained inclusion probabilities are higher compared to the hyperparameter setting $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$. Table A.3 in the Appendix shows the inclusion probabilities for this hyperparameter setting, where the $\hat{p}_p$ values are much closer to 1. we can explain this with the degree of shrinkage that is induced with this hyperparameter setting.

Recall that the importance parameter $\alpha_p \sim N(0, \gamma_p \tau_p^2)$, where the γ_p takes either the value v_0 when a spike draw is made, or 1 when a slab draw is made. If an α_p value of 0.1 needs to be sampled as the importance of a component, the following prior probabilities are obtained in both hyperparameter settings. We assume that $\tau^2 = 6$ is drawn, which is very close to the prior mean of τ^2 with $(a_\tau, b_\tau) = (5, 25)$.

With $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$, the shrinkage is only modest. For a spike draw ($\gamma_p = v_0$) with these hyperparameters, the prior probability for drawing $\alpha_p \geq 0.1$ is roughly 28%. In this case, we are able to obtain a low inclusion probability $\hat{p}_p$, while we sample $\alpha_p = 0.1$.

With $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 25)$, the shrinkage is heavier. In this case, the prior probability for drawing $\alpha_p \geq 0.1$ is roughly 0.5% when a spike draw is made. Thus, we are likely to need a slab draw ($\gamma_p = 1$) in order to draw $\alpha_p = 0.1$, given the low prior probability for this α_p when $\gamma_p = v_0$.

Thus, we obtain a higher inclusion probability in the case of $v_0 = 0.00025$, while the same effect is sampled. As we have seen in our application, we conclude that the inclusion probabilities are sensitive to the chosen hyperparameter settings. With a hyperparameter setting like $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 25)$ that induces heavy shrinkage, any small effect will obtain a high inclusion probability. With a hyperparameter setting like $v_0 = 0.005$ and $(a_\tau, b_\tau) = (5, 25)$ that induces more modest shrinkage, only large effects will obtain a high inclusion probability. Which interpretation is preferred depends on the goal of the researcher. To gain more insight into which components are of largest importance, we visualize the estimated smooth effects.

5.2.3 Visualization of Component Effects

The sampled intercept η_0 has a posterior mean of -2.2. To obtain the complete $\boldsymbol{\eta}$, we add the effect of all components $\sum_{p=1}^{P} \boldsymbol{D}_p\boldsymbol{\beta}_p$. We visualize the functional form of the component effects $\boldsymbol{D}_p\boldsymbol{\beta}_p$ to provide insights into how a predictor influences the response variable. As a Bayesian method is used, we can easily gain knowledge on the uncertainty of sums of $\boldsymbol{D}_p\boldsymbol{\beta}_p$ for different p. To obtain information on how a predictor influences the response variable, we use the sum of posteriors of all component effects $\boldsymbol{D}_p\boldsymbol{\beta}_p$ that use a certain predictor. In this way, we can easily visualize the total effect of a predictor, including its credible interval. In all visualizations, the y-axis shows this total effect of a predictor on an individual, which we denote as η. Just like is done for prediction, we obtain the effects $\boldsymbol{D}_p\boldsymbol{\beta}_p$ by applying spline interpolation for a range (or grid) of predictor values.

Figure 13 shows the effects of the 2 $\boldsymbol{D}_p\boldsymbol{\beta}_p$ components that are used for modeling the effect of the number of seconds the individual watched to the preceding program. Here we can clearly see the difference between the linear component in the left panel and the nonlinear effect in the middle panel. By construction, the nonlinear effect is centered around zero and does not follow a trend, due to the orthogonalization of the design matrix. By summing the posteriors of the two components, we obtain the total effect of watching the preceding program in the right panel, which we name the lead-in effect.

A possible explanation for the non-monotonicity of the lead-in effect can be found when investigating the timing of the ad breaks of the preceding program. As can be seen in Figure 14, many viewers stopped viewing the preceding program at 900, 1200 and 1620 seconds into the time block of the show. It is likely that this drop in the number of viewers occurred due to an ad break. The number of seconds between each ad break corresponds to 900, 300 and 420 seconds. Next to this, individuals that watched between 1400 and 1500 seconds of the program are likely to have been zapping away during commercials as well. These numbers of seconds match closely with the found local minima in the lead-in effect. Additionally, it is likely that an individual that have only seen a few minutes of the preceding program, tuned in early before starting to view the premiere of Designated Survivor. This matches with the first local maximum that is found for the lead-in effect.

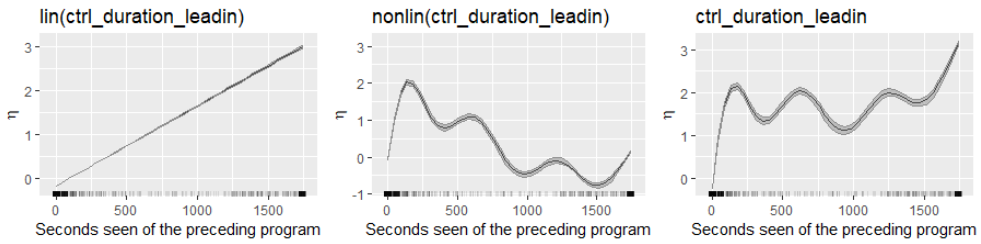

Figure 13: Estimated effects for the number of seconds that have been seen from the program preceding Designated Survivor. Left panel shows the linear component of the spline. Middle component shows the nonlinear component. Right panel shows the sum of the two components. Black line presents the posterior mean and the gray area the 10-90% credible interval of the effect. Ticks at x-axis denote the observed values.

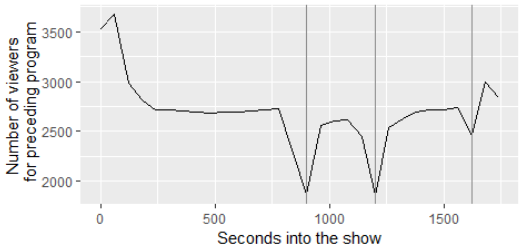

Figure 14: Number of total panel viewers for the preceding program. Vertical lines denote likely moments of ad breaks.

Thus, the number of seconds you have been watching the previous program is a good proxy whether you zap away during ad breaks, or whether you are a viewer who decides to tune in early. Here, we can see how the SSGAM allows us to model the smooth effect of a predictor of a very specific form.

As we have shown for the lead-in effect, each effect can be decomposed into its components $\boldsymbol{D}_p\boldsymbol{\beta}_p$. However, all following visualizations show the total effect of a predictor, which is more interesting for the purpose of interpreting the effect of a predictor.

The effect of transformed on-channel advertising minutes is shown in Figure A.4 in the Appendix. As we prefer to have a quantity on the x-axis which we can interpret more clearly, we transform the x-values back to their original values. After transforming the advertising exposure back to minutes of ads seen, we obtain the effect shown in Figure 15.

The minutes of ads seen on ABC are not randomly distributed across the individuals, as viewers that have seen a lot of ABC are likely to have seen more ads. Thus, a non-random subsample of individuals is exposed to advertising. As we include variables that measure the effect of viewing behavior and demographics, we assume that we appropriately control for having this non-random subsample of individuals with exposure to advertising. If this assumption holds, we can interpret the found effect as the effect of seeing on-channel advertising. However, we should note that more attention should be paid to adding the correct control variables to have more certainty on the estimated effect of advertising. This remains out of scope for this research, but we acknowledge the importance.

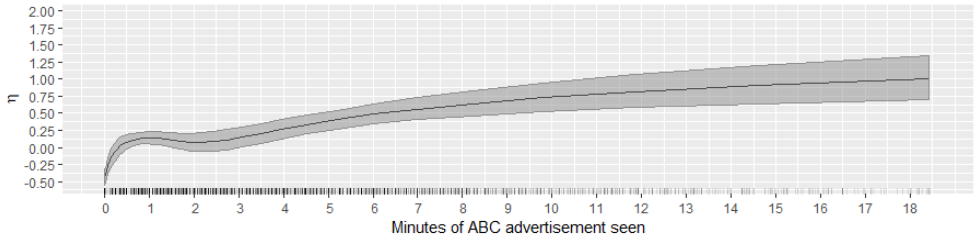

Figure 15: Effect of the on-channel advertising in total minutes of ads seen. Black line presents the posterior mean and the gray area the 10-90% credible interval of the effect.

In general, the advertising effectiveness diminishes after multiple exposures, as is often assumed in the literature. For 1 to 3.5 minutes of advertising seen, the effect remains roughly constant. Reaching new individuals with ads is more effective compared to exposing someone to an ad that has already seen ads previously. A shift from zero to one minute of ads seen increases the η with roughly 0.625 on average, whereas the η is only increased by roughly 0.125 for a shift from five to six minutes. Thus, we find that the additional exposures to advertising keep having a positive effect, but the increase in effectiveness is obtained by exposing new individuals to advertising. Therefore, an advertiser should be most concerned with reaching new individuals with their ads.

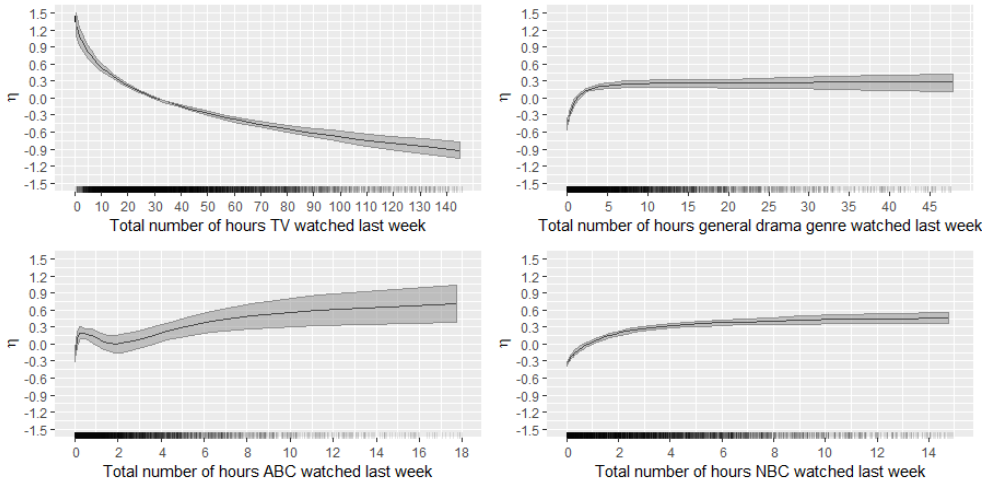

Figure 16: Effects for the viewing behavior variables. Black line presents the posterior mean and the gray area the 10-90% credible interval of the effect.

We show the effects of the variables that describe TV viewing behavior in the previous week in Figure 16, which are also transformed back to hours watched in the previous week. As can be seen in the upper left panel, we find lower η for individuals that watched a lot of TV in the previous week. Especially individuals that spent few hours watching TV in the previous week have an increased η. If an individual watches a lot of general drama or NBC, the η is increased, as can be seen in the right panels. For the genre general drama, a sharply diminishing curve is found. After roughly five hours of general drama, the effect no longer increases. As the genre of Designated Survivor also is general drama, we interpret this effect as genre loyalty, which increases η with a maximum of approximately 0.80 compared to someone who does not watch any general drama. For NBC, we conclude that Designated Survivor naturally appeals to frequent NBC viewers. For ABC in the bottom left panel, the effect is less pronounced. Individuals with roughly 30 minutes of ABC viewing in the previous week have an increased η of 0.45 on average. A reason for this might be that these individuals follow a single TV series on ABC and are likely to start following Designated Survivor as well. After 30 minutes, the effect diminishes until roughly 2.5 hours. These individuals might be viewers of different types of programming on ABC. After 2.5 hours, the effect keeps rising steadily. These individuals are likely to

be loyal to the ABC channel and watch the majority of content.

We show the effects of the demographic variables in Figure 17. Age, language of the household and a the yearly household income have the largest effect. For the age of the respondent, a nonlinear effect is found. On average, younger individuals have a lower η compared to older individuals. For example, the difference in η for an individual of 20 and 70 years old is 0.925 on average. The effects of the indicator variables for living in the Northeast Territory and for earning more than $125,000 per year are a bit lower.

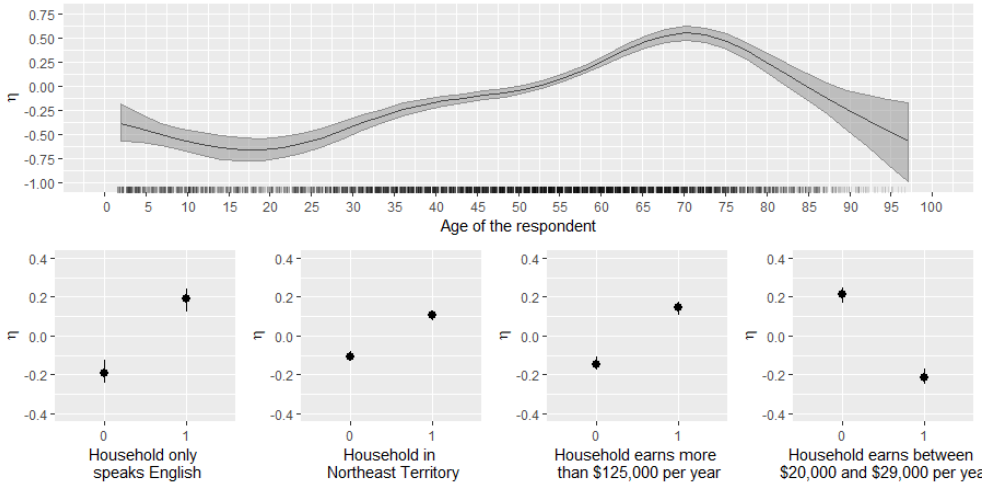

Figure 17: Effects for the social demographics. Black line and points present the posterior mean and the gray area the 10-90% credible interval of the effect.

Just like univariate effects, we can visualize the bivariate effects. The upper panel of Figure 18 shows the sum of all components that describe the total effect of age and exposure to advertising. The figure shows the cumulative effect of all eight components (2 for both univariate splines and 4 for the bivariate spline). In this way we can see how the η differs over different ages and minutes of exposure to on-channel ads.

The bottom panel of Figure 18 shows the interaction effect after removing the effect of the univariate effects of advertising and age. The posterior mean of η is small and zero is included in the 10-90% credible interval for the majority of the observed values. Thus, adding the four bivariate components to model the interaction effect between age and on-channel advertising exposure have little effect.

There are only two combinations of age and advertising exposure where $\eta = 0$ is not contained in the 10-90% credible interval. For individuals aged between 25 and 60 that have seen between 0 and 1 minutes of on-channel ads, the η is roughly 0.10 lower on average. For individuals aged between 80 and 95 that have seen 30 seconds of on-channel advertising, the η is roughly 0.15 higher on average. Apart from these two small effects, there is little posterior support for interaction effects between age and on-channel advertising. Similar conclusions are drawn for interactions with total ABC viewing or total TV viewing. Thus, the interaction effects are small.

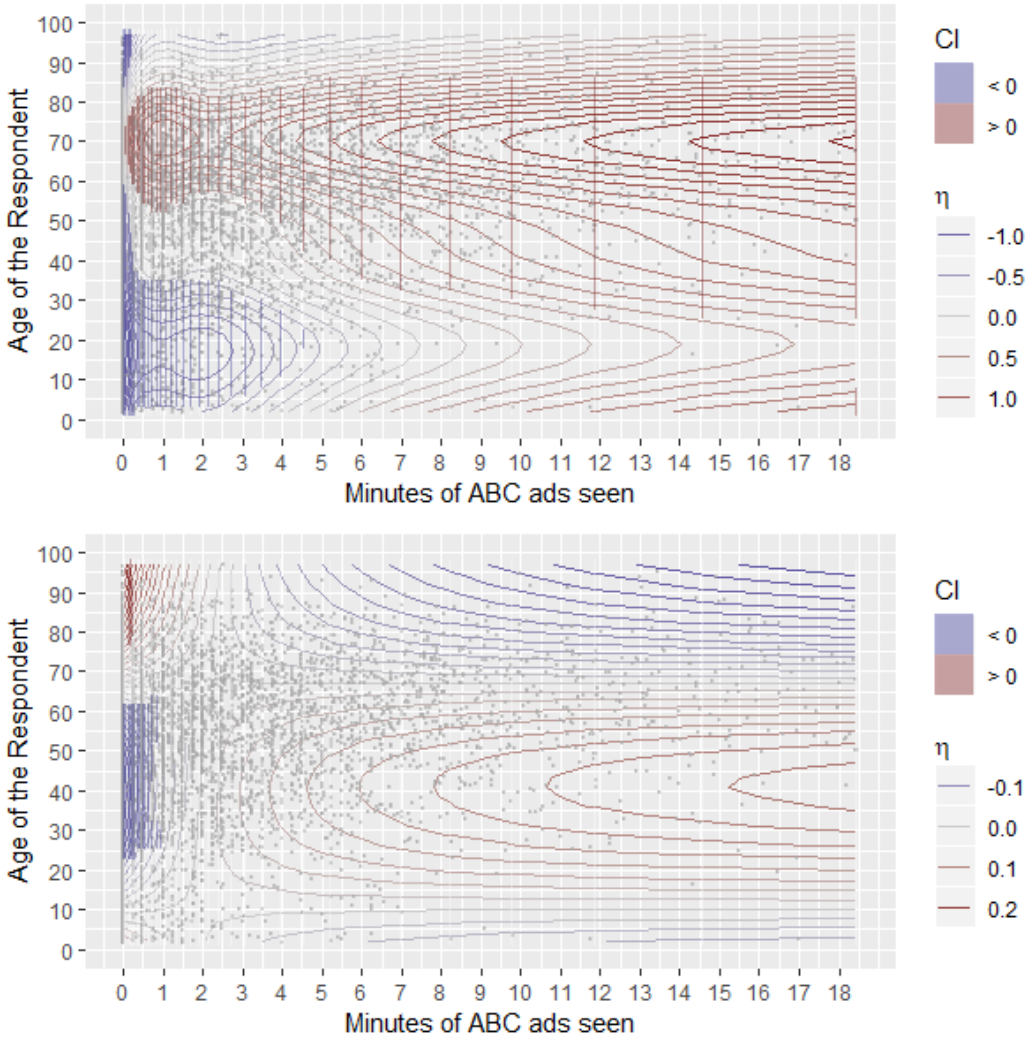

Figure 18: Interaction effect between age and total exposure to advertising. Upper panel shows a sum of the 8 components which are used to model the univariate splines and the bivariate interaction. Lower panel shows the sum of effects for the 4 components used to model the bivariate interaction. Regions with vertical stripes denote the combinations of age and advertising where $\eta = 0$ is not included in the 10-90% posterior region. Points in the plot denote observed values.

We conclude that SSGAM is capable of providing detailed insight into how the predictors relate to the response variable. With the estimated functions, we gained more knowledge on the lead-in effect and found supporting evidence on the diminishing returns of advertising. Next to this, we obtained a clear pictures of how viewing behavior and individual characteristics affect the probability of choosing a certain TV program.

5.3 Performance comparison

Table 4 shows the evaluation criteria for each method with the hyperparameter configuration that performed best in-sample. After performing the permutation test, the PCAProbit uses the first 26 principal components as predictors for the viewing behavior variables. We show the obtained eigenvalues and critical values from the permutation test in Figure A.5 in the Appendix.

Table 4: Predictive performance for each model with the hyperparameter setting that results in the best in-sample performance. Best performing methods in bold.

Method	DARTSSGAM	BART	DART	PCAProbit	DARTProbit
AUROC (in)	0.826	**0.858**	0.845	0.839	0.803
AUROC (out)	0.805	0.813	0.806	**0.815**	0.787
$\bar{D}$ (in)	0.685	**0.641**	0.655	0.671	0.718
$\bar{D}$ (out)	0.679	**0.666**	0.671	0.678	0.702
Computation minutes	182	11	4	3	4

As expected, the BART models performs good for both evaluation metrics. The PCAProbit also performs surprisingly well for both evaluation metrics. For the out-of-sample AUROC, it even performs best. Compared to the other methods, this method predicted lower probabilities for some of the 0 observations, which cause the true positive rate to quickly increase for a very low false positive rate. Thus, for some individuals, this method is very well able to predict that they will not watch Designated Survivor.

For the predictive deviance $\bar{D}$, we obtain the best performance with the BART and DART models. The restriction of only selecting a low number of variables for DART does not cause for much lower model fit. The difference between BART and DART is especially small for the out-of-sample criteria.

As expected, lower predictive performance is obtained when using the DARTSSGAM. However, the difference for the out-of-sample criteria is much smaller. Compared to the DARTProbit model, the DARTSSGAM has a higher predictive performance with the same set of predictors. Thus, the nonlinear effects are able to improve the fit compared to a model where linear relations are assumed a priori.

In this application, the SSGAM combined with variable selection using DART performs slightly worse compared to the strong prediction benchmark using BART. However, the added functionality of being able to visualize the nonlinear effects with SSGAM make it an attractive modeling technique to gain insights into how predictors influence the response variable.

6 Discussion

The SSGAM enables a marketing researcher to uncover the relationships between the response variable and the predictors. In combination with DART variable selection, the SSGAM can be applied on data sets with more predictors. The predictive performance of this method is only slightly lower than BART, a model which is known to have strong predictive performance. Thus, the cost of being able to visualize the nonlinear relations is low in this application.

It would be fruitful to apply the methodology to model other TV premiere launches to investigate if similar effects are found. The same model could either be used for different TV series, or it could be used to jointly estimate the effects for multiple series, where some coefficients are allowed to vary across TV series.

Next to that, we can apply the SSGAM methodology on different marketing mix modeling problems. For example, a smooth effect can be used to model price elasticity. With this flexible method, no assumptions have to be made on the effects between price and increase in sales, such that effects such as psychological price barriers are taken care of, without having to define them a priori. As is found in this study, the smooth effects estimated with the SSGAM methodology could uncover new insights into how marketing variables and social demographics influence consumer behavior.

Additionally, we can analyze more factors that influence the effectiveness of advertising. One variable that was found to be important in previous research is the time between seeing ads. One approach would be to estimate how quickly the effect of advertising diminishes over time by estimating what proportion of the advertising effect is retained after each week. In this research, we have seen that almost all advertising types were not selected by the DART variable selection. However, it is also possible to overrule this variable selection procedure and manually add predictors in the model. In this way, we can estimate smooth effects for all advertising types of interest.

In future work, more attention could be paid to measuring the causal effect of advertising. Leamer (1983) warns that inaccurate treatment effects could be obtained when a large set of control variables are used. However, cherry picking a subset of predictor is not the solution to this issue. When we use shrinkage to estimate the effect of control variables, Hahn et al. (2018) show that a bias is induced on the treatment effect in a linear regression context. By using recent methods such as Bayesian Causal Forests proposed by Hahn et al. (2017), we can correct for the bias of the effect of a binary treatment variable. As the SSGAM uses smooth effects to measure the effect of advertising, social demographics and viewing behavior, we cannot directly apply the proposed methodology to correct for the bias of advertising effectiveness that is likely to be present. Thus, how we can estimate the causal smooth effect of a predictor with the SSGAM is an interesting direction of future work.

A limitation of the current study is that it is difficult to evaluate the difference in performance if no variable selection is applied. Adding all predictors to the SSGAM simply was too demanding computationally for this study. This makes it difficult to conclude whether all important effects are picked up by the DART variable selection. However, the predictive performance of BART on the full set of predictors and the SSGAM on the subset of predictors is similar. Thus, it is unlikely the SSGAM could have performed much better with a larger set of predictors.

Next to that, the evaluation metrics for the predictive performance would ideally be based on cross validation. This would stabilize the performance criteria, as it

would be possible that a method might just have a good performance on the one split considered in this study. As the models are fitted for multiple hyperparameter settings, calculating the evaluation criteria for multiple splits was too demanding computationally for this research.

Another improvement could be made on the interaction detection. Currently, the approach with the Friedman H-statistic and a Random Forest is a simple solution to make a small selection of possible interactions. More attention could be paid to select interactions for all variables and detecting higher order interactions. Next to that, we can compare the performance of different methods to detect interactions. A possible approach would be to investigate how the posterior samples from DART can be used to detect interactions. This extension would make DART an even more versatile method for selecting predictors from a large set.

Our proposed combination of using DART variable selection with the SSGAM methodology allows us to estimate smooth effects between the predictors and the response variable for a large data set. This makes this combination of methods an excellent candidate for marketing research applications with many predictors. With the SSGAM, we do not have to make assumptions on the functional form of the effect of a marketing variables, and we shrink small effects to zero to prevent overfitting. With the obtained smooth effects, we can test hypotheses on how predictors relate to the response variable. Next to this, it gives evidence of how a predictor should be added to a model to accurately describe its effect. By using Bayesian inference, the uncertainty of the estimated effects can be analyzed with ease. In our application, this advance in interpretation only comes at the cost of slightly lower predictive performance compared to a strong benchmark. Thus, this combination of methods alleviate the burden of making restrictive assumptions for a researcher, by letting the data do the heavy lifting.

A Appendix

A.1 Starting values for SSGAM sampler

In order to make notation easier, we denote $\boldsymbol{y} = (y_1, \ldots, y_N)$ and $N \times S_D$ matrix $\boldsymbol{D} = [\mathbf{1}, \boldsymbol{D}_1, \ldots, \boldsymbol{D}_P]$ with $S_D = 1 + \sum_{p=1}^{P} S_p$.

Algorithm A.1 Generation of $\boldsymbol{\beta}^{(0)}$,$\boldsymbol{\alpha}^{(0)}$ and $\boldsymbol{\xi}^{(0)}$ for SSGAM with binary responses

1. Calculate initial $\mu_i = (y_i + 0.5)/2$ for $i = 1, \ldots, N$.
2. Calculate initial (or updated) $\sigma_i^2 = \mu_i(1 - \mu_i)$ and $\eta_i = \log(\mu_i/(1 - \mu_i))$ for $i = 1, \ldots, N$.
3. Calculate initial (or updated) $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_N, \mathbf{0}'_P]'$ of length $N + S_D$, with $\tilde{y}_i = (\sigma_i(\eta_i + (y_i - \mu_i)/\sigma_i^2))$ and matrix $\tilde{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{D} \circ \boldsymbol{\sigma} \mathbf{1}'_{S_D} \\ 2^{-1/2} \boldsymbol{I}_{S_D} \end{bmatrix}$ of size $(N + S_D) \times S_D$ with $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)'$.
4. The vector with regression coefficients for $\hat{\boldsymbol{\delta}}$ is initialized (or updated) by fitting $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{D}}\hat{\boldsymbol{\delta}}$, where $\hat{\boldsymbol{\delta}}$ is obtained by using the QR decomposition of $\tilde{\boldsymbol{D}}$.
5. For all $i = 1, \ldots, N$, update $\mu_i = (\tilde{\boldsymbol{D}}_i\hat{\boldsymbol{\delta}})^{-1}$, where $\tilde{\boldsymbol{D}}_i$ corresponds to the i^{th} row of $\tilde{\boldsymbol{D}}$.
6. Repeat steps 2 – 5 for five times, such that $\hat{\boldsymbol{\delta}}$ is updated at every iteration. The final $\hat{\boldsymbol{\delta}} = (\eta_0^{(0)}, \boldsymbol{\beta}^{(0)'})'$.
7. For each chain that is run in parallel, add $N(0, 1)$ noise to $\eta_0^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ to obtain different starting values for each chain.
8. All $\boldsymbol{\beta}_p^{(0)}$ are rescaled by the drawn $\gamma_p^{(0)}\tau_p^{2(0)}$ from the priors for $p = 1, \ldots, P$. This causes the $\boldsymbol{\beta}_p^{(0)}$ with $\gamma_p^{(0)} = v_0$ to be close to zero as starting value.
9. Finally $\alpha_p^{(0)} = S_p^{-1} \sum_{k=1}^{S_p} |\beta_{pk}^{(0)}|$ for $p = 1, \ldots, P$, where $\beta_{pk}^{(0)}$ is the k^{th} element of $\boldsymbol{\beta}_p^{(0)}$. Lastly, the elements in $\boldsymbol{\xi}^{(0)}$ are calculated as $\xi_{pk}^{(0)} = \beta_{pk}^{(0)}/\alpha_p^{(0)}$ for $k = 1, \ldots, S_p$ and $p = 1, \ldots, P$.

A.2 Additional Results for DART Variable Selection

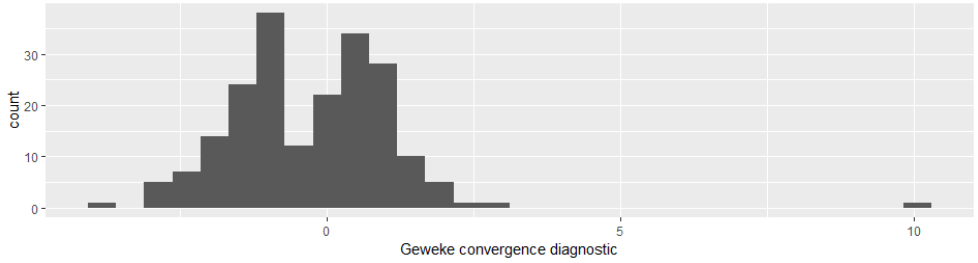

Figure A.1: Geweke convergence diagnostic of the drawn variable inclusion probabilities for DART with $T = 200$, $\lambda = 1$ and $\rho = 20$.

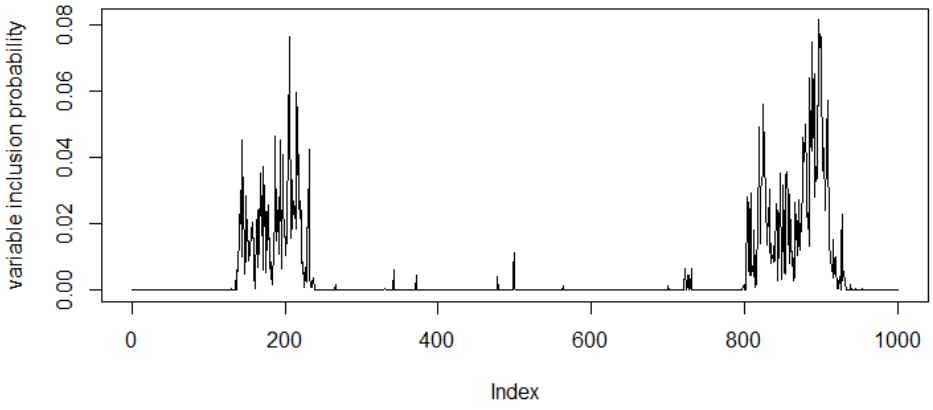

Figure A.2: Traceplot of the splitting probability for the amount of hours an individual has seen programs with the genre award ceremony.

Table A.1: Number of DART models that selected a predictor. Predictors in the subset of the selected DART model with $T = 200$, $\lambda = 1$ and $\rho = 20$ in bold.

Variable name	**selected by number of DARTs**	**Variable name**	**Selected by number of DARTs**
ctrl_duration_leadin	12	exp_total_duration_30	5
sd_age	12	**vb_total_duration**	3
vb_channel_nbc	12	sd_householdincomecode$15,000-$19,000	3
vb_genre_general_drama	10	sd_nsimarketrankranges50-99	2
vb_channel_other	9	sd_nsimarketrankranges100+	2
vb_genre_popular_music	8	vb_genre_participation_variety	2
sd_householdincomecode$125,000 or more	8	vb_genre_science_fiction	2
exp_total_duration_10	8	vb_channel_uni	2
vb_channel_abc	7	**sd_territorycodeNortheast Territory**	2
exp_total_duration_on	7	sd_territorycodePacific Territory	2
vb_genre_award_ceremonies	6	vb_genre_news	1
vb_channel_2_t	6	exp_total_duration_15	1
sd_householdincomecode$20,000-$29,000	6	vb_genre_concert_music	1
sd_householdlanguageEnglish Only	5	vb_channel_top15	1
vb_channel_tel	5	vb_channel_nfln	1

Table A.2: Selected predictors which have been selected at least once to form a split on in 50% of the ensembles for a DART with $T = 200$, $\rho = 20$ and $\lambda = 1$.

Variable name	**Proportion of ensembles which split on this variable**	**Description**
ctrl_duration_leadin	1.000	Number of seconds the individual watched to the preceding TV program
sd_age	1.000	Age of the panelist
sd_householdlanguageEnglish_Only	1.000	Indicator for a household that only speaks English
sd_territorycodeNortheast_Territory	0.537	Indicator for a household that is situated in the North East states in the USA
sd_householdincomecode_$125000_or_more	0.685	Indicator for a household that has a yearly income of more than $125,000
sd_householdincomecode_$20,000-$29,000	0.810	Indicator for a household that has a yearly income between $20,000 and $29,000
vb_total_duration	1.000	Number of seconds the individual watched to the TV the previous week
vb_genre_general_drama	1.000	Number of seconds the individual watched to TV programs with the genre general drama the previous week
vb_channel_abc	1.000	Number of seconds the individual watched to the TV channel ABC the previous week
vb_channel_nbc	1.000	Number of seconds the individual watched to the TV channel NBC the previous week
exp_total_duration_on	1.000	Number of seconds the individual has been exposed to ads for Designated Survivor on ABC

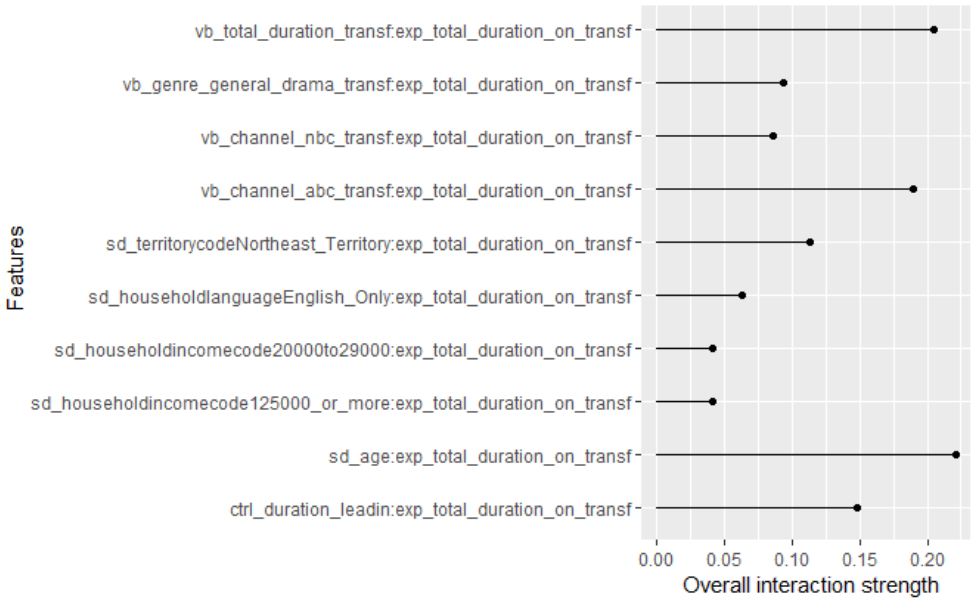

Figure A.3: Friedman H statistic of the interaction between exp_total_duration_on and all other selected predictors for a Random Forest with 250 trees to model $\boldsymbol{y}$ on the set of selected predictors $\tilde{\boldsymbol{X}}$.

A.3 Additional Results for SSGAM

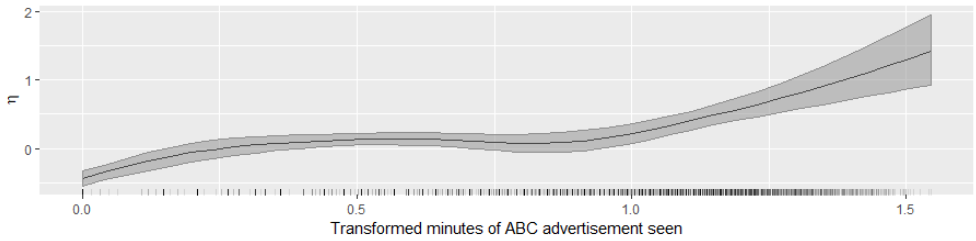

Figure A.4: Total effect of transformed minutes of ABC advertising seen. Black line corresponds to posterior mean, shaded area to the 10-90% credible interval. Ticks at the x-axis denote observed values.

Table A.3: The 6 most common combinations of components with $P[\gamma_p = 1] \geq 0.5$ and corresponding proportions, general inclusion probabilities $\hat{p}_p$, influence of a component π_p, and number of columns in the design matrix S_p for all components in the SSGAM with $v_0 = 0.00025$ and $(a_\tau, b_\tau) = (5, 25)$.

	Combination of components (proportions)								
	1	2	3	4	5	6			
Component	(0.66)	(0.14)	(0.09)	(0.05)	(0.01)	(0.01)	$\hat{p}_p$	π_p	S_p
lin(ctrl_duration_leadin)	x	x	x	x	x	x	1.000	0.301	1
nonlin(ctrl_duration_leadin)	x	x	x	x	x	x	1.000	0.076	7
lin(vb_total_duration_transf)	x	x	x	x	x	x	1.000	0.005	1
nonlin(vb_total_duration_transf)							0.017	0.000	9
lin(vb_genre_general_drama_transf)	x	x	x	x	x	x	1.000	0.017	1
nonlin(vb_genre_general_drama_transf)		x			x		0.221	0.006	8
lin(vb_channel_abc_transf)	x	x	x	x	x	x	1.000	0.113	1
nonlin(vb_channel_abc_transf)	x	x	x	x	x	x	0.996	0.021	7
lin(vb_channel_nbc_transf)	x	x	x	x	x	x	1.000	0.100	1
nonlin(vb_channel_nbc_transf)							0.019	0.000	8
lin(sd_age)	x	x	x	x	x	x	1.000	0.096	1
nonlin(sd_age)	x	x	x	x	x	x	1.000	0.021	9
fct(sd_householdlanguageEnglish_Only)	x	x	x	x	x	x	1.000	0.049	1
fct(sd_territorycodeNortheast_Territory)	x	x		x	x		0.889	0.000	1
fct(sd_householdincomecode125000_or_more)	x	x	x	x	x	x	0.998	0.001	1
fct(sd_householdincomecode20000to29000)	x	x	x	x	x	x	0.998	0.032	1
lin(exp_total_duration_on_transf)	x	x	x	x	x	x	1.000	0.145	1
nonlin(exp_total_duration_on_transf)	x	x	x	x	x	x	0.998	0.021	7
lin(sd_age) ×lin(exp_total_duration_on_transf)							0.017	0.000	1
nonlin(sd_age) ×lin(exp_total_duration_on_transf)							0.017	0.000	8
lin(vb_total_duration_transf) ×lin(exp_total_duration_on_transf)							0.018	0.000	1
nonlin(vb_total_duration_transf) ×lin(exp_total_duration_on_transf)							0.017	0.000	8
lin(vb_channel_abc_transf) ×lin(exp_total_duration_on_transf)				x		x	0.094	0.001	1
nonlin(vb_channel_abc_transf) ×lin(exp_total_duration_on_transf)							0.017	0.000	6
lin(sd_age) ×nonlin(exp_total_duration_on_transf)							0.027	-0.001	6
nonlin(sd_age) ×nonlin(exp_total_duration_on_transf)							0.035	0.000	26
lin(vb_total_duration_transf) ×nonlin(exp_total_duration_on_transf)							0.019	-0.001	6
nonlin(vb_total_duration_transf) ×nonlin(exp_total_duration_on_transf)							0.017	0.000	27
lin(vb_channel_abc_transf) ×nonlin(exp_total_duration_on_transf)							0.022	-0.002	6
nonlin(vb_channel_abc_transf) ×nonlin(exp_total_duration_on_transf)							0.019	-0.001	21

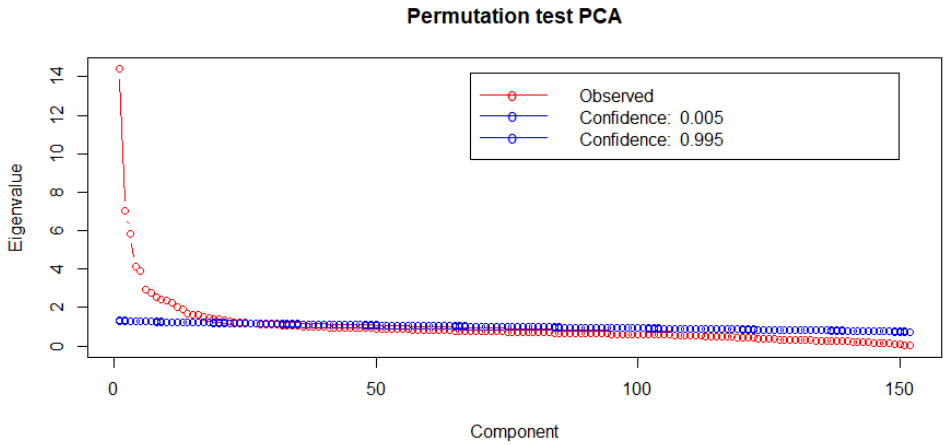

Figure A.5: Obtained eigenvalues for the Principal components of the viewing behavior predictors. blue circles two sided critical values for $\alpha = 0.01$ obtained by performing a permutation test. The first 26 Principal Components are used for modeling.

References

Akima, H. and Gebhardt, A. (2016). *akima: Interpolation of Irregularly and Regularly Spaced Data*. R package version 0.6-2.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

Baglama, J. and Reichel, L. (2005). Augmented implicitly restarted lanczos bidiagonalization methods. *SIAM Journal on Scientific Computing*, 27(1):19–42.

Barbieri, M. M. and Berger, J. O. (2004). Optimal predictive model selection. *Ann. Statist.*, 32(3):870–897.

Ben-Israel, A. and Greville, T. N. E. (2005). *Generalized Inverses*. Springer-Verlag, New York.

Bleich, J., Kapelner, A., George, E. I., and Jensen, S. T. (2014). Variable selection for bart: An application to gene regulation. *Ann. Appl. Stat.*, 8(3):1750–1781.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brezger, A. and Lang, S. (2006). Generalized structured additive regression based on bayesian p-splines. *Computational Statistics & Data Analysis*, 50(4):967 – 991.

Campbell, M. C. and Keller, K. L. (2003). Brand familiarity and advertising repetition effects. *Journal of Consumer Research*, 30(2):292–304.

Chammartin, F., Scholte, R. G., Malone, J. B., Bavia, M. E., Nieto, P., Utzinger, J., and Vounatsou, P. (2013). Modelling the geographical distribution of soil-transmitted helminth infections in bolivia. *Parasites & Vectors*, 6(1):152.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298.

Danaher, P. and Dagger, T. (2012). Using a nested logit model to forecast television ratings. *International Journal of Forecasting*, 28(3):607 – 622.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with *b*-splines and penalties. *Statistical Science*, 11(2):89–102.

Eilers, P. H. C. and Marx, B. D. (2010). Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(6):637–653.

Eilers, P. H. C., Marx, B. D., and Durbán, M. (2015). Twenty years of p-splines. *Statistics and Operations Research Transactions*, 39(2):149–186.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232.

Friedman, J. H. and Popescu, B. E. (2008). Predictive learning via rule ensembles. *Ann. Appl. Stat.*, 2(3):916–954.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Geweke, J. F. (1991). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Staff Report 148, Federal Reserve Bank of Minneapolis.

Gu, C. (1992). Diagnostics for nonparametric regression models with additive terms. *Journal of the American Statistical Association*, 87(420):1051–1058.

Guadagni, P. M. and Little, J. D. C. (1983). A logit model of brand choice calibrated on scanner data. *Marketing Science*, 27(1):29–48.

Hahn, P. R., Carvalho, C. M., He, J., and Puelz, D. (2018). Regularization and confounding in linear regression for treatment effect estimation. *Bayesian Analysis*, 13(1):163–182.

Hahn, P. R., Murray, J. S., and Carvalho, C. M. (2017). Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects. Available on arXiv: https://arxiv.org/pdf/1706.09523.pdf.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1(3):297–310.

Hastie, T. and Tibshirani, R. (2000). Bayesian backfitting. *Statist. Sci.*, 15(3):196–223. With comments and a rejoinder by the authors.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning*. New York: Springer.

Hobert, J. P. and Casella, G. (1996). The effect of improper priors on gibbs sampling in hierarchical linear mixed models. *Journal of the American Statistical Association*, 91(436):1461–1473.

Iyer, G., Soberman, D., and Villas-Boas, J. M. (2005). The targeting of advertising. *Marketing Science*, 24(3):461–476.

Kapelner, A. and Bleich, J. (2016). bartmachine: Machine learning with bayesian additive regression trees. *Journal of Statistical Software, Articles*, 70(4):1–40.

Kneib, T., Konrath, S., and Fahrmeir, L. (2011). High dimensional structured additive regression models: Bayesian regularization, smoothing and predictive performance. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 60(1):51–70.

Krugman, H. E. (1972). Why three exposures may be enough. *Journal of Advertising Research*, 12(6):11–14.

Lai, Y.-S., Biedermann, P., Ekpo, U. F., Garba, A., Mathieu, E., Midzi, N., Mwinzi, P., N'Goran, E. K., Raso, G., Assaré, R. K., Sacko, M., Schur, N., Talla, I., Tchuenté, L.-A. T., Touré, S., Winkler, M. S., Utzinger, J., and Vounatsou, P. (2015). Spatial distribution of schistosomiasis and treatment needs in sub-saharan africa: a systematic review and geostatistical analysis. *The Lancet Infectious Diseases*, 15(8):927 – 940.

Lang, S. and Brezger, A. (2004). Bayesian p-splines. *Journal of Computational and Graphical Statistics*, 13(1):183–212.

Leamer, E. E. (1983). Let's take the con out of econometrics. *The American Economic Review*, 73(1):31–43.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

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

Lovett, M. J. and Staelin, R. (2016). The role of paid, earned, and owned media in building entertainment brands: Reminding, informing, and enhancing enjoyment. *Marketing Science*, 35(1):142–157.

McCulloch, R., Sparapani, R., Gramacy, R., Spanbauer, C., and Pratola, M. (2018). *BART: Bayesian Additive Regression Trees*. R package version 1.8.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *J. Amer. Statist. Assoc.*, 83(404):1023–1036. With comments by James Berger and C. L. Mallows and with a reply by the authors.

Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

Pechmann, C. and Stewart, D. W. (1988). Advertising repetition: A critical review of wearin and wearout. *Current Issues and Research in Advertising*, 11(1-2):285–329.

Rossi, P. E. and Allenby, G. M. (2000). Statistics and marketing. *Journal of the American Statistical Association*, 95(450):635–638.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields*. Chapman & Hall/CRC, London.

Rust, R. T. and Alpert, M. I. (1984). An audience flow model of television viewing choice. *Marketing Science*, 3(2):113–124.

Rust, R. T., Kamakura, W. A., and Alpert, M. I. (1992). Viewer preference segmentation and viewing choice models for network television. *Journal of Advertising*, 21(1):1–18.

Scheipl, F. (2010). Normal-mixture-of-inverse-gamma priors for bayesian regularization and model selection in structured additive regression models. Technical report, Ludwig Maximilians Universität München, Department of Statistics.

Scheipl, F. (2011). spikeSlabGAM: Bayesian variable selection, model choice and regularization for generalized additive mixed models in R. *Journal of Statistical Software*, 43(14):1–24.

Scheipl, F., Fahrmeir, L., and Kneib, T. (2012). Spike-and-slab priors for function selection in structured additive regression models. *Journal of the American Statistical Association*, 107(500):1518–1532.

Schmidt, S. and Eisend, M. (2015). Advertising repetition: A meta-analysis on effective frequency in advertising. *Journal of Advertising*, 44(4):415–428.

Schweidel, D. A. and Moe, W. W. (2016). Binge watching and advertising. *Journal of Marketing*, 80(5):1–19.

Shachar, R. and Emerson, J. W. (2000). Cast demographics, unobserved segments, and heterogeneous switching costs in a television viewing choice model. *Journal of Marketing Research*, 37(2):173–186.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). Rocr: visualizing classifier performance in r. *Bioinformatics*, 21(20):7881.

Tillmanns, S., Ter Hofstede, F., Krafft, M., and Goetz, O. (2017). How to separate the wheat from the chaff: Improved variable selection for new customer acquisition. *Journal of Marketing*, 81(2):99–113.

Vakratsas, D. and Ambler, T. (1999). How advertising works: What do we really know? *Journal of Marketing*, 63(1):26–43.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2):3–28.

Webster, J. and Wakshlag, J. J. (1983). A theory of television program choice. *Communication Research*, 10(4):430–446.

Wedel, M. and Kannan, P. (2016). Marketing analytics for data-rich environments. *Journal of Marketing*, 80(6):97–121.