Annals of Actuarial Science (2024), 1–20
doi:10.1017/S1748499524000241

ORIGINAL RESEARCH PAPER

Distill knowledge of additive tree models into generalized linear models: a new learning approach for non-smooth generalized additive models

Arthur Maillart[1,2] and Christian Robert[2,3]

[1]Detralytics, Saint-Josse-ten-Noode, Belgium; [2]Université Lyon 1, Institut de Science Financière et d'Assurances, Lyon, France; and [3]Laboratory in Finance and Insurance - LFA CREST - Center for Research in Economics and Statistics, Paris, France
Corresponding author: Christian Robert; Email: christian.robert@univ-lyon1.fr

(Received 16 May 2024; revised 15 September 2024; accepted 17 September 2024)

Abstract

Generalized additive models (GAMs) are a leading model class for interpretable machine learning. GAMs were originally defined with smooth shape functions of the predictor variables and trained using smoothing splines. Recently, tree-based GAMs where shape functions are gradient-boosted ensembles of bagged trees were proposed, leaving the door open for the estimation of a broader class of shape functions (e.g. Explainable Boosting Machine (EBM)). In this paper, we introduce a competing three-step GAM learning approach where we combine (i) the knowledge of the way to split the covariates space brought by an additive tree model (ATM), (ii) an ensemble of predictive linear scores derived from generalized linear models (GLMs) using a binning strategy based on the ATM, and (iii) a final GLM to have a prediction model that ensures auto-calibration. Numerical experiments illustrate the competitive performances of our approach on several datasets compared to GAM with splines, EBM, or GLM with binarsity penalization. A case study in trade credit insurance is also provided.

Keywords: Additive tree ensembles; auto-calibration; generalized additive models; generalized linear models; partitioning methods; XAI

1. Introduction

Insurance companies need explainable pricing and reserving models because the decisions they make based on these models can have significant financial and legal implications, but also because they are crucial for building trust with all stakeholders and regulators. Explainability in the context of this paper will refer to the ability to explain or to present in understandable terms to a human how the models make their decisions or predictions. Generalized additive models (GAMs) with smooth functions of the predictor variables, originally developed by Trevor Hastie and Robert Tibshirani (Hastie & Tibshirani, 1986), have emerged as a spearhead of the actuaries' toolbox (see e.g. Wood, 2017 for an introduction with R). The combination of additive nature, smooth functions, interpretability, and transparent variable selection in GAMs makes them highly explainable models that are suitable for a wide range of applications where model interpretability is important:

- GAMs are an extension of generalized linear models (GLMs) that allow for nonlinear relationships between predictors and the response variable. But they retain their additive structure,

meaning that they model the relationship between the predictors and the response variable as a sum of individual functions of the predictors. This additive nature allows for clear separation of the effects of each predictor, making it easy to explain the contribution of each predictor to the overall model prediction. Each function can be visualized and interpreted separately, providing insights into how each predictor affects the response variable.

- GAMs use smooth functions, such as spline functions or other smooth basis functions, to model the relationship between the predictors and the response variable. These smooth functions are typically visually interpretable and can be plotted to understand the shape of the relationship. This makes it easier to explain the model to non-technical stakeholders by visualizing and describing the smooth functions in simple terms.
- GAMs typically use techniques such as cross-validation or information criteria to automatically select the most important predictors to include in the model. This makes the variable selection process transparent and allows for easy explanation of which predictors are included and why, adding to the model's interpretability.

Although GAMs developed by Hastie and Tibshirani are flexible and powerful statistical modeling technique, there are some well-known limits to their use that should be considered when applying this modeling approach. GAMs use smooth, continuous functions to model the relationship between each feature and the response variable. If the true relationship is highly nonlinear or involves abrupt changes, the smoothing process might oversimplify these nuances, leading to underfitting. Moreover, the smoothing methods in GAMs can be sensitive to outliers, and the smooth functions used in GAMs might struggle to fit the data accurately, potentially smoothing over important signals or failing to ignore outliers appropriately. By design, GAMs model the effects of features additively and usually do not automatically include interaction terms between features (though interactions can be manually added). In cases where the relationship between features and the response is significantly influenced by interactions between variables, GAMs might not capture these complexities adequately. Finally, GAMs can be computationally expensive to fit, particularly when a large number of predictors are included in the model. This can make the model difficult to use in practice or in real-time applications.

In the meantime, bagging and boosting techniques as well as neural networks have appeared as effective machine learning methods and have given actuaries great hope for improving their models. But their opacity and the difficulties in understanding and interpreting their results have not led them to replace GAMs or GLMs. An alternative path was therefore to use these modern machine learning methods to improve the estimation of nonlinear relationships between predictors and the response variable.

Explainable Boosting Machine (EBM), developed by Nori et al. (2019), is the prominent example of GAMs with non-smooth functions of the predictor variables that uses a boosting algorithm to make the model's accuracy comparable to state-of-the-art machine learning methods like random forest and boosted trees. An EBM is a tree-based GAM where shape functions are gradient-boosted ensembles of bagged trees. Each tree operates on a single variable and is grown by repeatedly cycling through features forcing the model to sequentially consider each feature as an explanation of the current residual. The EBM-BF (EBM-BestFirst) is a sparse version of EBM that put most weight on a few very important features, and little or no weight on features whose signal could be learned by other stronger, correlated features.

GAMboostLSS, developed by Hofner et al. (2014), is an R package for fitting GAMs for location, shape, and scale (GAMLSS) to potentially high-dimensional data using boosting techniques. GAMLSS extends traditional linear regression models for the mean by allowing to model different parameters (e.g. variance, skewness, and kurtosis). This makes it versatile for handling data with various distributional shapes and complexities.

EBM or GAMboostLSS have quickly gained popularity. But they may have some limitations that should be taken into account when considering their use in a particular machine learning

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

task. For example, they can be less robust to missing data than other machine learning models as they require imputation or removal of missing data before they can be trained. But this limitation is also shared by many other algorithms.

An alternative approach to GAMs using boosting algorithms is to consider GLMs with innovative regularization technique. Binarsity (Alaya et al., 2019) is a new type of regularization or penalization technique specifically designed to handle high-dimensional and sparse one-hot encoded features in linear supervised learning. One-hot encoding can lead to high-dimensional binary feature vectors, especially when dealing with categorical variables with many categories. These high-dimensional feature vectors can pose challenges in linear models. Binarsity encourages group sparsity among the binary features, making the model more interpretable and reducing the risk of overfitting. The strength of the binarsity penalty is controlled by a hyperparameter that needs, however, to be tuned with care for optimal model performance.

The concept of employing a GLM with categorized features to approximate a GAM has also been discussed in Henckaerts et al. (2022). This paper introduces the Model-Agnostic Interpretable Data-driven suRRogate (available as the R package "maidrr"). The proposed method extracts insights from a complex model through partial dependence effects, facilitating intelligent feature engineering. This process involves grouping values of variables, effectively segmenting the feature space and automating the selection of significant variables. Subsequently, a transparent GLM is applied to these binned features, incorporating their critical interactions.

In this paper, we propose a new approach for estimating a GAM with non-smooth feature functions. This approach is at least as competitive as the most recent approaches. It is based on three steps and on an additive tree model (ATM) as an initial machine learning model with high learning capacity. An ATM uses an ensemble of decision trees to make predictions. It is also known as a gradient boosting machine with trees as base learners, or simply a gradient tree boosting model. Our aim is to use the knowledge of the way to split the covariates space brought by the ATM for binning the covariates. In the first step, we fit the ATM. In the second step, for each decision tree of the ensemble, we fit a GLM with the binned covariates, collect the underlying stepwise functions of the GLM predictive scores, and then aggregate them. An additional interest in our approach is that the second step may be slightly modified to derive a surrogate model (i.e. an explanatory model) of the ATM. In the third step, we fit a final GLM to have an *auto-calibrated* prediction model that corrects for possible systematic cross-financing between different price cohorts within the insurance portfolio. The results we obtain on synthetic data show the competitive performance of our approach compared to other methods for estimating GAMs.

The main contribution of this paper is to show how to produce a GAM with non-smooth feature functions using the knowledge acquired by an ATM and by distilling it appropriately in GLMs while performing a series of binning, aggregation, and calibration steps. Unlike recent competing approaches, we moreover provide an auto-calibrated GAM model. Finally, our approach results in an interpretable model based on the additive structure of the GAM, which enables an "all other things being equal" analysis that is much appreciated by actuaries. The rest of this paper is structured as follows. Section 2 introduces the additive structure of the GAM and describes precisely the different steps of our algorithm. Section 3 validates our approach on synthetic data with known ground-truth feature shapes and compares it with the historical GAM with splines, EBM, and GLM with binarsity penalization. A case study in trade credit insurance is provided in Section 4.

2. Methodology

Given a vector of covariates $\mathbf{X} = (X_1, \ldots, X_p)$, a univariate response variable Y, a link function g, and shape smooth functions $f_j, j = 1, \ldots, p$, a GAM can be written as:

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

$$g\big(\mathbb{E}[Y|\mathbf{X}=\boldsymbol{x}]\big)=\beta_0+\sum_{j=1}^{p}f_j\left(x_j\right),\quad \boldsymbol{x}=(x_1,\ldots,x_p).$$

An exponential family distribution is specified for Y (e.g. Gaussian, Binomial, or Poisson distributions). The additive structure allows for clear separation of the effects of each covariate, making it easier to draw insights of the contribution of each covariate to the overall model prediction, while using the functions f_j (usually splines or other basis functions) leads to capture the underlying nonlinearities in the data. Identifiability constraints are in general applied, for example, $\mathbb{E}[f_j\left(X_j\right)]=0$ for $j=1,\ldots,p$, to make the model identifiable. The GAM with pairwise interactions (GA^2M) includes pairwise shape functions:

$$g\big(\mathbb{E}[Y|\mathbf{X}=\boldsymbol{x}]\big)=\beta_0+\sum_{j=1}^{p}f_j\left(x_j\right)+\sum_{(i,j)\in\mathcal{S}_2}f_{i,j}\left(x_i,x_j\right),\quad \boldsymbol{x}=(x_1,\ldots,x_p),$$

where $\mathcal{S}_2$ is a set of non-empty subsets of $\{1,\ldots,p\}$ with cardinality 2, and $f_{i,j}$ are shape smooth bivariate functions. GAMs with or without pairwise interactions are highly interpretable because the impact of each shape functions f_j or $f_{i,j}$ on the prediction can be visualized as a graph, and it is easily understood how a GAM works by reading the different features from the graphs and adding them together.

In GAM with splines approach, basis functions for the splines are first chosen (e.g. the family of cubic splines are piecewise-defined cubic polynomials), then splines require the specification of knot locations (knots are points along the predictor variable where the smoothness of the curve may change), and finally the GAM model is fitted to the data using techniques like least squares estimation or maximum likelihood estimation. It is possible to select smoothing parameters to control the degree of smoothness of each shape functions and to determine how much the smooth functions can deviate from linearity. Techniques like cross-validation are often used to choose these smoothing parameters.

In EBM, the shape functions are fitted through a process that involves creating a set of additive functions while boosting them to improve predictive accuracy. The process starts by initializing the EBM model and setting the number of boosting iterations (the number of weak models to combine). In each boosting iteration and for each covariate, the weak model (a simple decision tree) is trained to approximate the negative gradient of the loss function with respect to the current model's predictions. This weak model aims to correct the errors made by the previous ensemble of additive functions. The additive functions are then modified to incorporate the predictions from the new weak model. To prevent overfitting, a shrinkage or learning rate parameter is applied to the predictions of the weak model before adding them to the additive functions.

For the GLM with binarsity penalization, the idea is to one-hot encode continuous features and to encourage block sparsity in the GLM's coefficients with an appropriate penalization. The model defines one group of binary features for each raw continuous feature (these groups are naturally ordered). The binarsity penalization then combines a group total-variation penalization, with an extra linear constraint in each group to avoid collinearity. This penalization forces the weights of the model to be as constant as possible within a group by selecting a minimal number of relevant cut points.

The Model-Agnostic Interpretable Data-driven suRRogate of Henckaerts et al. (2022) begins with a black-box model that is transformed into an easier-to-understand surrogate. Knowledge is derived from the original model using partial dependence effects, which show how each feature influences the target. These effects guide the binning of values within each feature using dynamic programming. The binning method varies by feature type: for continuous or ordinal features, only neighboring values are grouped together, while for nominal features, any levels can be clustered. This method ensures consistent grouping, fully segmenting the feature space.

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Subsequently, a GLM is applied to this organized data, considering all features in a categorical format along with their important interactions.

We propose a new methodology for estimating the shape functions f_j of a GAM by approximating them as averages of piecewise functions. Let us assume for a moment that the shape functions f_j of the GAM may be written as follows:

$$f_j\left(x_j\right) = \sum_{k=1}^{n_j} \beta_{k,j} \mathbb{I}_{\left\{x_j \in (s_{k,j}, s_{k+1,j}]\right\}}$$

where $n_j \in \mathbb{N}_*$, $\beta_{k,j} \in \mathbb{R}$, and the support of X_j is included in $\bigcup_{k=1}^{n_j} (s_{k,j}, s_{k+1,j}]$. The pairwise interactions $f_{i,j}$ of GA^{2}M may also be written as stepwise functions on $\mathbb{R}^2$ based on Cartesian products of intervals. Then the GAM could be estimated as a GLM (assuming that the number of intervals n_j is not too large). However, there are two issues in using such an approximation. First, the splitting in intervals is not given a priori and should be made according to the shape of the shape functions f_j which are unknown. Second, these functions, although not necessarily continuous, are not assumed to be piecewise constant functions. Our methodology consists in distilling knowledge of an ATM for binning the covariates and in estimating an ensemble of piecewise functions whose aggregation will provide a smoother estimate with less variance of g ($\mathbb{E}[Y|\mathbf{X} = x]$).

One-hot encoding features presents a few challenges and limitations. Specifically, it fragments the features, which disrupts the internal cohesion of each feature, ultimately reducing sometime interpretability. Converting a continuous feature into multiple dummy categorical features may also lead to a loss of information. However bagging and boosting may be used to limit these effects. The core technique in bagging involves averaging the predictions of multiple models, each trained on different subsets of the data. This is particularly effective when the models are non-smooth estimators, as their individual inconsistencies and abrupt changes tend to neutralize each other through averaging. This process effectively smooths out the final estimator and minimizes the variance part of the prediction error. Gradient boosting starts with a base estimator and iteratively improves upon it by adding new models that correct the previous errors (residuals). Each new model (often a decision tree) is focused on improving the areas where the previous models performed poorly. This stepwise refinement can smooth out the abrupt changes and discontinuities typical of non-smooth models because each subsequent model is trained to correct the exaggerated predictions or errors of the prior models. As more models are added to the ensemble, their individual predictions are combined, usually through a weighted average where each model contributes a small incremental improvement. This averaging process tends to smooth out noise and fluctuations in the predictions. In the context of non-smooth base estimators, the averaging of multiple incremental improvements over the residuals can result in a final prediction that is smoother than any individual model's predictions.

2.1 Step 1: knowledge distillation

Knowledge distillation is a technique used to transfer the knowledge learned by a complex, high-performing machine learning model, known as the teacher model, to one or several simpler, smaller models, known as the student models.

In our fitting procedure, an ATM plays the role of the teacher. An ATM is an ensemble model of decision trees such as random forests (Breiman, 2001) and boosted trees (Friedman, 2001). Because of their high prediction performance, ATMs are one of the must-try methods when dealing with real problems. By combining decision trees, ATMs can then capture nonlinear relationships between the input features and the target variable.

Let $\mathcal{D} = \{(x_i, y_i),\ i = 1, \ldots, n\}$ be an observed i.i.d sample. The first step of our approach consists in fitting an ATM on $\mathcal{D}$ using the log-likelihood loss function associated with Y and the link function g of the GAM. The output space of the ATM is a collection of predictions from

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

individual trees, combined according to a chosen ensemble method (random forests or boosted trees). For classification tasks, the output space consists of class probabilities or class labels, while for regression tasks, it contains continuous prediction values. In a random forest, predictions from individual trees are combined by taking a majority vote for classification or an average for regression. In boosting, the predictions of each tree are weighted based on their accuracy and used to update the model in the next iteration. The final prediction is the sum of the predictions from all trees in the ensemble. The input covariate space is therefore splitted by the ATM into regions (rectangles) with an assigned prediction value to each region. We denote by $s_{k,j}^{(l)}$ the ordered k-th split for the j-th variable and the l-th tree (derived from the leaf nodes of the tree). We denote by $n_j^{(l)}$ the number of splits for the j-th variable and the l-th tree.

2.2 Step 2: building ensemble of GLMs

For each tree l, we fit the following GLM to data:

$$g\big(\mathbb{E}[Y|\mathbf{X}=\boldsymbol{x}]\big) = \beta_0^{(l)} + \sum_{j=1}^{p} \sum_{k=1}^{n_j^{(l)}-1} \beta_{k,j}^{(l)} \mathbb{I}_{\left\{x_j \in (s_{k,j}^{(l)}, s_{k+1,j}^{(l)}]\right\}}$$

and consider it as a student model for the knowledge distillation. Pairwise interactions may be added. Let us denote by $\hat{\beta}_{k,j}^{(l)}$ the estimates of the coefficients $\beta_{k,j}^{(l)}$. Each GLM is only a rough approximation of the GAM. By combining the GLM linear scores, we get a more accurate estimate of f_j given by:

$$\hat{f}_j^{(0)}(x_j) = \sum_{l=1}^{L} \omega_l \sum_{k=1}^{n_j^{(l)}} \hat{\beta}_{k,j}^{(l)} \mathbb{I}_{\left\{x_j \in (s_{k,j}^{(l)}, s_{k+1,j}^{(l)}]\right\}}$$

where L is the number of tree partitions in the ensemble of the ATM and ω_l is the weight given to the l-th tree. Weights can be uniform in the case of the bagging approach or can be derived from the weights of the incremental models in the case of the boosting approach. This ensemble approach improves model generalization and helps reduce overfitting of the final GLM.

Remark 2.1. *In machine learning, a surrogate model is a simplified model that approximates the behavior of a more complex and computationally expensive model. Surrogate models are used when the original model is too complex to analyze directly or when it requires significant computational resources to run, making it impractical for iterative tasks like optimization, sensitivity analysis, or real-time prediction. To build a surrogate model of the ATM, instead of fitting the GLM to data, we advice to fit it to ATM predictions. We have noted that this alternative approach also provides an extremely powerful predictive GAM model.*

The link function g is not in general a linear function, and the proposed linear aggregation method may induce biased predictions for the target Y (after taking into account the transformation of the average score by g^{-1}). The third step of our methodology aims at debiasing these predictions.

2.3 Step 3: auto-calibration

Auto-calibration refers to the process of calibrating the outputs of a machine learning model to better match the true probabilities of the output classes in case of classifications or to have a better balance between sums of prediction and sums of observations in case of regressions. Many machine learning models, such as random forests, gradient boosting, and neural networks,

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

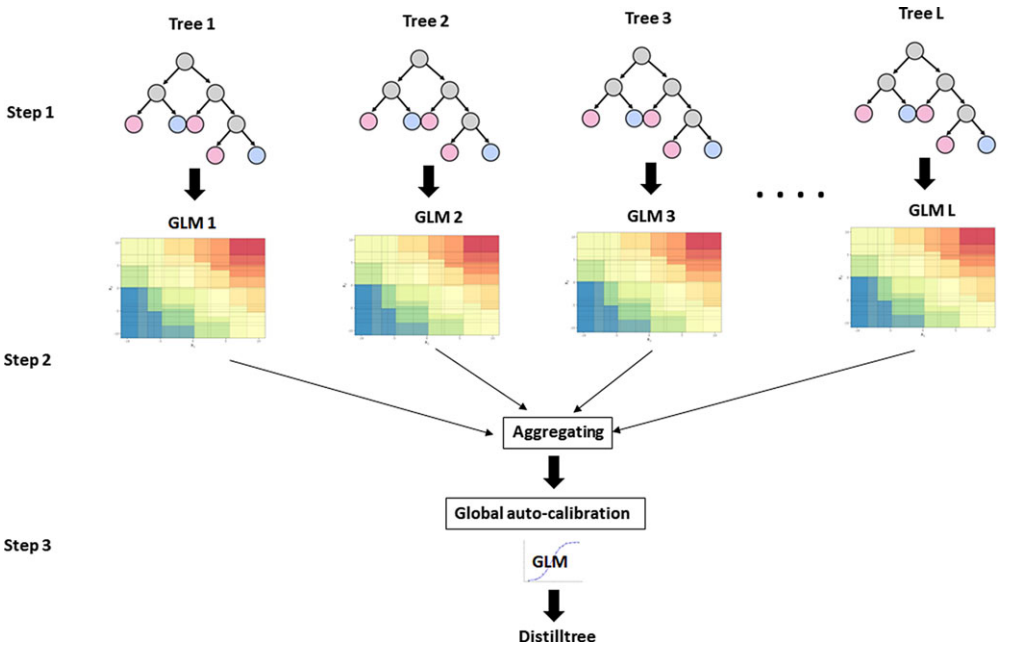

Figure 1. Schematic overview of the estimation procedure.

output scores or probabilities that are not calibrated. Auto-calibration is important in insurance pricing where candidate premiums have to reveal the risk at individual policy level but also enable the global price level to reproduce the experience within the portfolio. In Denuit et al. (2021), the authors propose to correct for bias by adding an extra local GLM step to the analysis with the output of the first step estimate. In Wüthrich & Ziegel (2023), an isotonic recalibration is applied to a given regression model to ensure auto-calibration. In Lindholm et al. (2023), the covariate space is first partitionned using two different approaches: (i) duration-weighted equal-probability binning, and (ii) binning by duration-weighted regression trees, and then a local bias adjustment is implemented.

However, the previous procedures would lead to the destruction of the additive structure of the model derived from Step 2. We therefore favor a global auto-calibration by adding an extra global GLM step with the following model:

$$g\big(\mathbb{E}[Y|\mathbf{X}=\boldsymbol{x}]\big) = \beta_0 + \sum_{j=1}^{p} \alpha_j \hat{f}_j^{(0)}\left(x_j\right).$$

Our final estimates of the shape functions f_j are then given by:

$$\hat{f}_j\left(x_j\right) = \hat{\alpha}_j \hat{f}_j^{(0)}\left(x_j\right), \quad j = 1, \ldots, p.$$

This third step is not mandatory in our GAM learning approach. It may lead sometimes to a deterioration in the performance of the results if the model obtained in Step 2 significantly transforms the estimated probabilities or conditional expectations compared with the empirical ones. The user can decide to keep the results obtained at the end of Step 2 or to perform Step 3 to obtain results closer to the experience of his/her insurance portfolio. In the simulated examples and the case study considered below, we have not observed any significant drop in performance.

Fig. 1 schematizes the several steps of our competing GAM approach.

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Table 1. Shape functions f_j

$f_1(x_1) = 3x_1$	$f_2(x_2) = x_2^3$	$f_3(x_3) = \pi^{x_3}$
$f_4(x_4) = \exp(-2x_4^2)$	$f_5(x_5) = (1 + \lvert x_5 \rvert)^{-1}$	$f_6(x_6) = x_6 \log(\lvert x_6 \rvert)$
$f_7(x_7) = \sqrt{2\lvert x_7 \rvert} + \max(0, x_7)$	$f_8(x_8) = x_8^4 + 2\cos(\pi x_8)$	

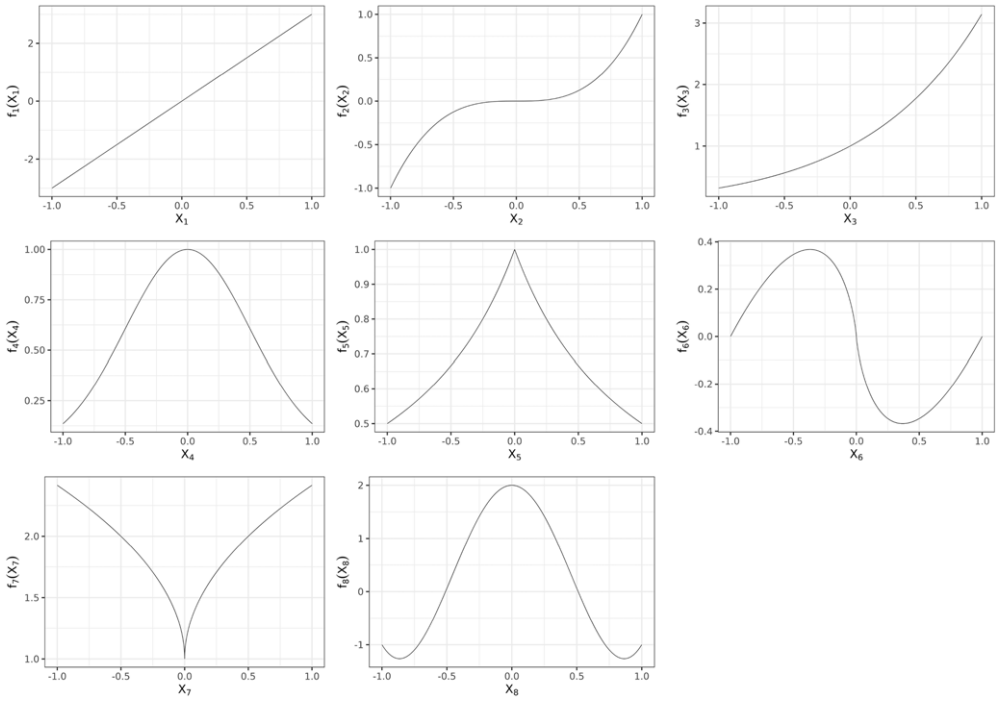

Figure 2. Plots of the shape functions f_j.

3. Validation using synthetic data with known ground truth

We simulate data from both regression and classification models with known ground-truth feature shapes to see if our new methodology can recover these feature shapes. The functions f_j considered are given in Table 1 and are represented graphically in Fig. 2. These linear and highly nonlinear functions have been proposed in Hooker (2004) and also used in Friedman & Popescu (2008), Tsang et al. (2017), and Tan et al. (2018).

We now compare our approach with the GAM with splines, EBM, and GLM with binarsity penalization, for two types of prediction tasks: regression and classification.

3.1 Regression task

We first consider the regression task for which $g(y) = y, y \in \mathbb{R}$, and Y (given $\mathbf{X} = x$) has a Gaussian distribution with mean equal to $\sum_{j=1}^{8} f_j(x_j)$ and standard deviation equal to 0.5. As in Friedman & Popescu (2008), we assume that $\mathbf{X}$ is a random vector whose components are independent and distributed according to the uniform distribution on $(-1, 1)$. As in Tsang et al. (2017), we add to the list of covariates two noise covariates that have no effect on $\mathbb{E}[Y|\mathbf{X} = x]$, X_9 and X_{10}, which have been assumed to be independent on $\mathbf{X}$ and to have uniform distribution on $(-1, 1)$. We simulate samples of size 50,000.

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Table 2. Choice of the hyperparameters for the regression task (based on a 5-fold cross-validation)

Model	Package	Model parameters not set to their default values
GBM	H2O R	`ntrees` = 200, `max_depth` = 9, `learning_rate` = 0.1, `sample_rate` = 1, `col_sample_rate` = 1
GAM	mgcv R	`basis function` = cubic regression splines
EBM	InterpretML	`learning_rate` = 0.01, `interactions` = 0, `validation_size` = 0.15
Binarsity	github *binarsity*	`ncuts` = 30, `C` = 1e4

Table 3. Comparison results based on R^2

	Train R^2	Test R^2
GBM	95.70%	95.56%
GAM	95.60%	95.65%
EBM	95.68%	95.62%
Binarsity	95.45%	95.46%
Distilltrees	95.61%	95.67%

For the ATM, we used GBM of the H2O R package. For the GAM, we used gam of the mgcv R package. For the EBM, we use ebm from the python package InterpretML. For the GLM with binarsity penalization, we used the github repository */SimonBussy/binarsity*. For each model, the choice of the hyperparameters are given in Table 2.

The performances of the various models measured by the R^2 metric are shown in Table 3. They are very close to each other. It is not surprising that GAM performs as well as the other competitors, because it was used to generate the data.

The estimated feature shapes are plotted in Fig. 3. The four learning approaches provide estimated functions that are very close to each other. The functions given by EBM, however, tend to be less smooth than the others. For f_3, f_4, f_5, and f_7, there are level discrepancies between the true functions and the estimated functions. This is a consequence of the identifiability constraints imposed on the various estimation methods, for which the integrals of the estimated shape functions must be equal to 0. The free-signal covariates X_9 and X_{10} are each estimated close to 0 for the four learning approaches.

Table 4 presents the mean absolute deviations for each function being estimated and for each learning method used. Generally, GAM with splines outperforms others as it inherently provides default smooth functions for estimating such functions, giving it a natural advantage. The results of other methods are fairly similar, though EBM performs marginally better than GLM with Binaristy. Notably, Distilltrees is better in detecting the absence of variables better than its competitors (X_9 and X_{10}).

All four learning approaches provide auto-calibrated predictions (or nearly so), see Fig. 4. To obtain this figure, the dataset is sorted based on the values of the predictions of $\mathbb{E}[Y|\mathbf{X} = x]$. The data are then bucketed into 50 equally populated classes based on quantiles. Within each bucket, the average of the predictions is calculated as well as the average of the observations Y. Both averages are then graphed for each class. For all four learning approaches, the points are almost identical and aligned on the $y = x$ line. Distilltrees provides slightly larger deviations from this line than its competitors for the most extreme quantile classes, without these deviations being really significant.

In the Appendix, we have proposed an additional study where the covariates X_1 and X_2, X_3 and X_4 respectively, are almost collinear. This study makes it possible to identify the algorithms

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

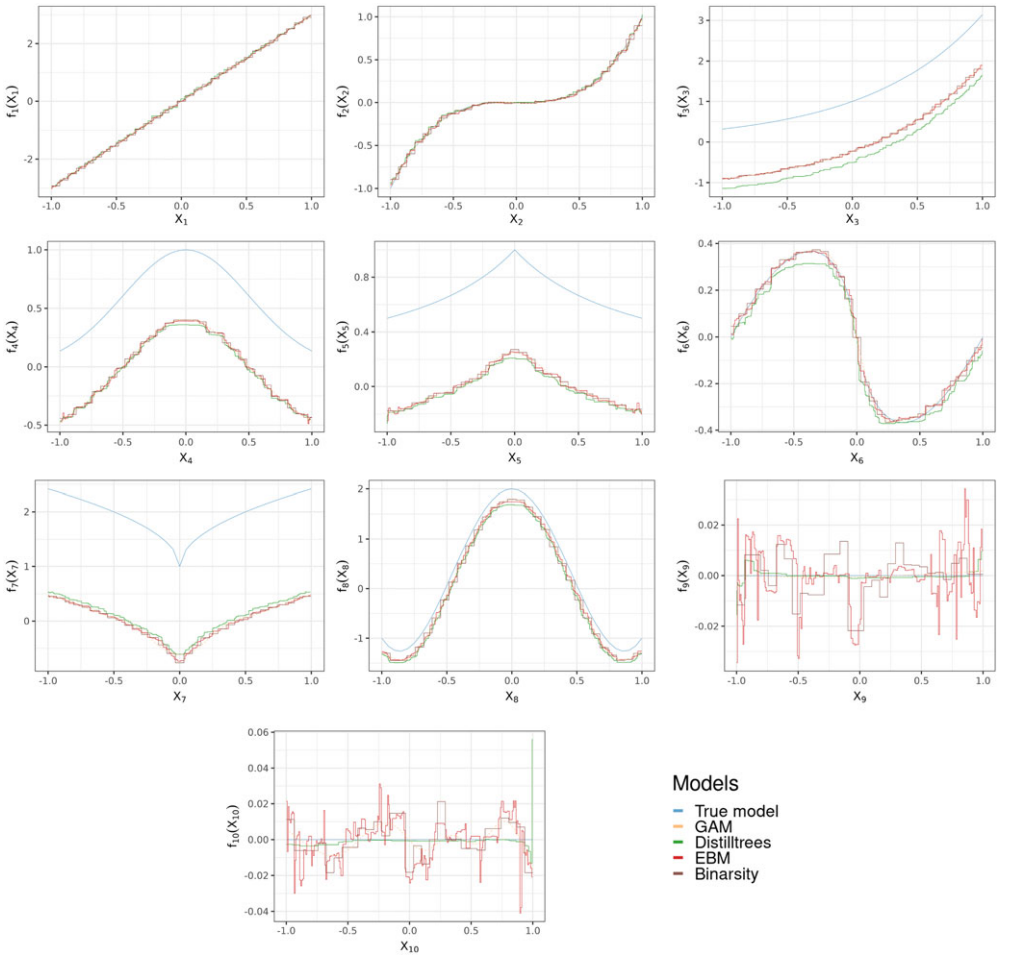

Figure 3. Feature shapes learned using GAM with splines, EBM, Distilltrees, and GLM with binarsity penalization.

that are most robust to the collinearity problems encountered in practice. It is a study that considers one of the worst cases in terms of dependence between covariates. Weaker dependence between covariates produces results that are better in terms of estimating shape functions. We have also studied the sensitivity of the results to sample size by considering a sample 50 times smaller.

3.2 Classification task

We now consider the classification task where $g^{-1}\left(p\right) = \log\left(p/(1-p)\right)$, $p \in (0,1)$, and Y (given $\mathbf{X} = x$) has a Binomial distribution with parameter $g(\sum_{j=1}^{8} f_j\left(x_j\right))$. As for the regression task, we assume that $\mathbf{X}$ is a random vector whose components are independent and distributed according to the uniform distribution on $(-1,1)$. We also add to the list of covariates two noise covariates that have no effect on $\mathbb{E}[Y|\mathbf{X} = x]$, X_9 and X_{10}, which have been assumed to be independent of $\mathbf{X}$ and to have uniform distribution on $(-1,1)$. We simulate samples of size 50,000.

For each model, we used the same packages as those presented in Section 3.1. The choice of the hyperparameters are given in Table 5.

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Table 4. Mean absolute deviation per variable and learning model

Variable	Binarsity	Distilltrees	EBM	GAM
X_1	0.0491	0.0348	0.0224	0.0029
X_2	0.0191	0.0133	0.0131	0.0037
X_3	1.2325	1.4794	1.2320	1.2321
X_4	0.5984	0.6132	0.5987	0.5988
X_5	0.6932	0.7143	0.6939	0.6939
X_6	0.0167	0.0324	0.0113	0.0093
X_7	1.1923	1.1163	1.1926	1.1931
X_8	0.2007	0.2575	0.1972	0.1975
X_9	0.0073	0.0010	0.0075	0.0006
X_{10}	0.0082	0.0014	0.0091	0.0049

Table 5. Choice of the hyperparameters for the classification task (based on a 5-fold cross-validation)

Model	Package	Model parameters not set to their default value
GBM	H2O R	`ntrees` = 200, `max_depth` = 9, `learning_rate` = 0.1, `sample_rate` = 1, `col_sample_rate` = 1
GAM	mgcv R	`basis function` = cubic regression splines
EBM	InterpretML	`learning_rate` = 0.01, `interactions` = 0, `validation_size` = 0.15
Binarsity	github *binarsity*	`ncuts` = 30, `C` = 1e4

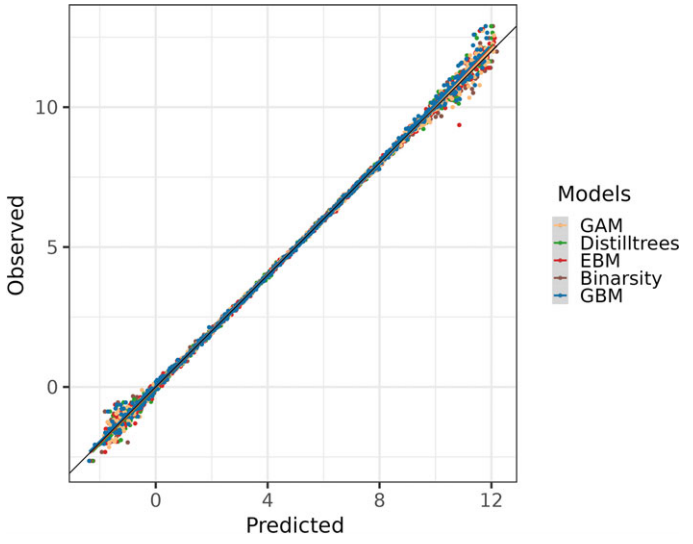

Figure 4. Predicted probabilities vs observed probabilities for the different learning models.

The performances of the various models measured by the *AUC* metric are shown in Table 6. They are still very close to each other as for the regression task.

The estimated feature shapes are plotted in Fig. 5. Compared with the regression task, the functions are a little bit less well estimated. GAM fares best, while providing inherently smooth

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Table 6. Comparison results based on *AUC*

	Train *AUC*	Test *AUC*
GBM	89.47%	89.28%
GAM	89.31%	89.41%
EBM	89.56%	89.29%
Binarsity	89.38%	89.25%
Distilltrees	89.50%	89.23%

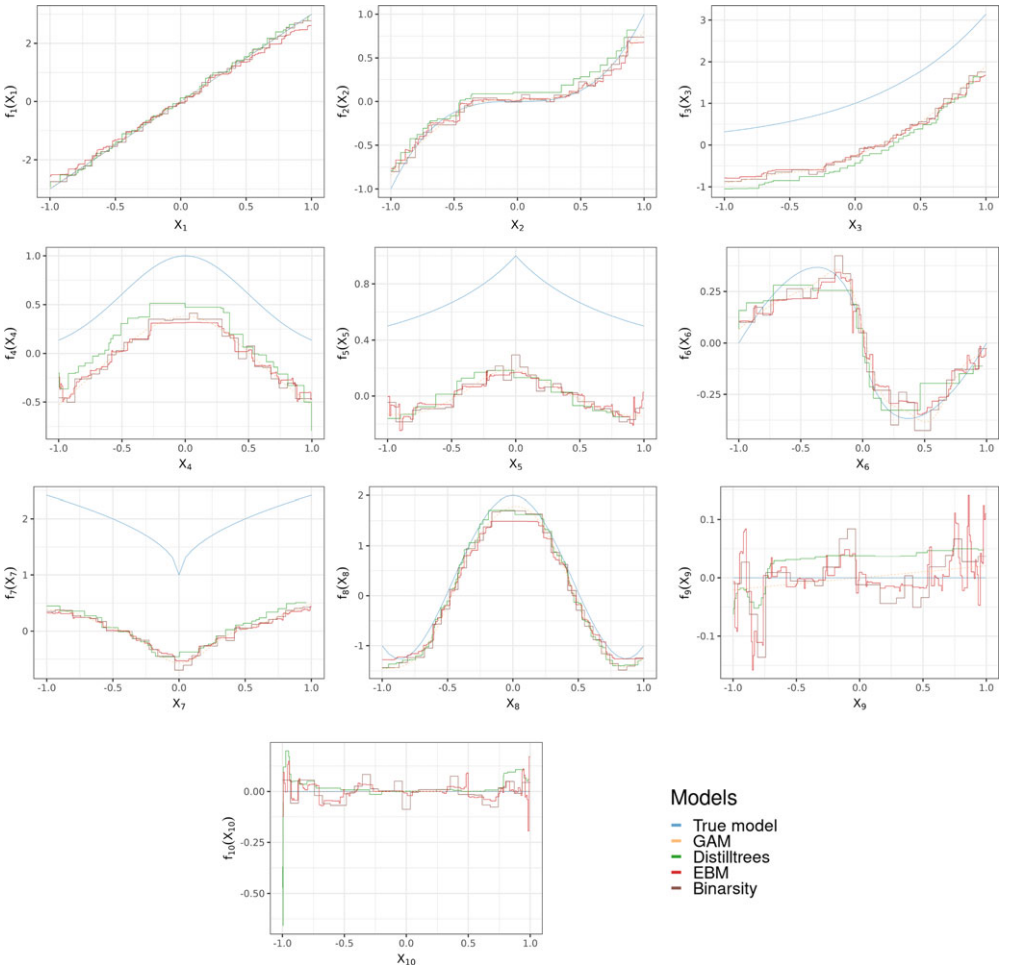

Figure 5. Feature shapes learned using GAM with splines, EBM, Distilltrees, and GLM with binarsity penalization.

estimates. Distilltrees tends to provide smoother estimates than these two other competitors. As for the regression task, for f_3, f_4, f_5, and f_7, there are level discrepancies between the true functions and the estimated functions beacause of the identifiability constraints.

All four learning approaches provide auto-calibrated predictions (see Fig. 6).

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Table 7. Choice of the hyperparameters for the credit insurance case study (based on a 5-fold cross-validation)

Model	Package	Model parameters not set to their default value
XGB	H2O R	`ntrees` = 4000, `max_depth` = 5, `learning_rate` = 0.05, `min_rows` = 10
GAM	mgcv R	`basis function` = cubic regression splines
EBM	InterpretML	`learning_rate` = 0.01, `interactions` = 0, `validation_size` = 0.15
Binarsity	github *binarsity*	`ncuts` = 30, C = 1e5).

Table 8. Variable importance of the XGB used for the ATM

Name of the Covariate	% of scaled Importance of the XGB	Meaning of the covariate
V_2	23.29	Automatic acceptance processing system variable
V_{12}	20.08	Risk exposure variable
V_{17}	13.90	Risk assessment variable
V_{15}	10.96	Ratio V_3/V_2
V_3	9.68	Variable relating to the amount of the request

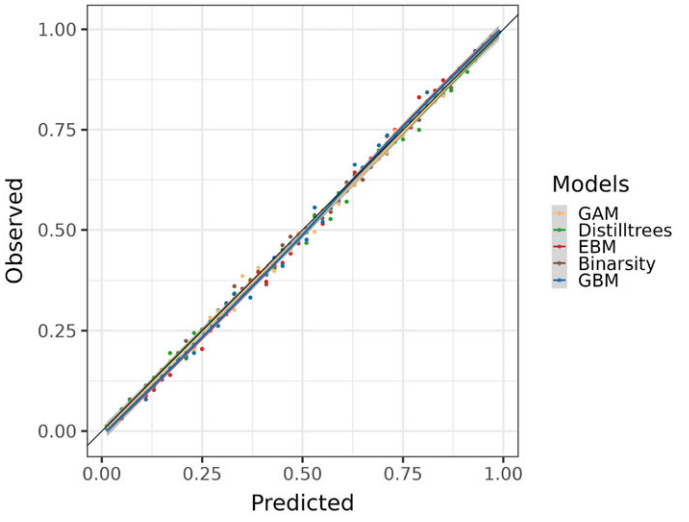

Figure 6. Predicted probabilities vs observed probabilities for the different learning models.

4. A case study

In this section, we provide a case study in trade credit insurance where we compare our new learning approach with GAM with splines, EBM, and GLM with binarsity penalization.

Allianz Trade is an international insurance company specialized in trade credit insurance. Credit insurers provide a range of financial services to businesses to help protect them against the risk of non-payment by their customers. They have to assess the creditworthiness of businesses' customers or clients. Based on their own risk assessment, credit insurers provide businesses with recommendations regarding the credit limits they should extend to their customers.

Allianz Trade provided us with a database of businesses' customers default events. This database also contains information on the financial stability, payment history, and credit ratings

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Table 9. Comparison results based on *AUC*

	Train *AUC*	Test *AUC*
XGB	90.28%	89.38%
GAM	81.67%	81.37%
EBM	88.15%	87.73%
Binarsity	87.68%	86.41%
Distilltrees without auto-calibration	87.86%	87.00%
Distilltrees	87.78%	86.91%

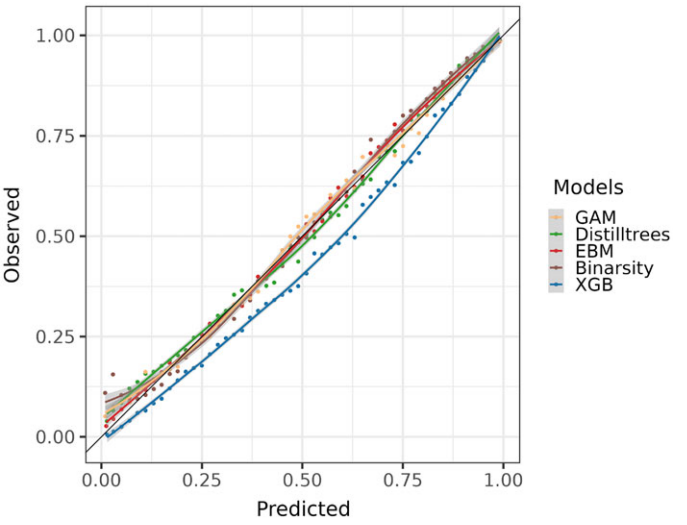

Figure 7. Predicted probabilities vs oberved probabilities for the different learning models.

of its customers to determine the level of risk associated with each businesses' customer. For confidentiality reasons, we cannot give the exact definition of each variable. But we explain the meaning of the most important variables for the models.

For each model (except the ATM), we used the same packages as those presented in Section 3.1. For the ATM, we used XGB of the H2O R package. The choice of the hyperparameters are given in Table 7.

The five most important variables based on the feature importance of the XGB are given in Table 8.

The performances of the various models measured by the *AUC* metric are shown in Table 9. XGB naturally performs best, since it takes into account interactions between covariates and is not constrained by the linear structure of the additive model. However, the performances of EBM, GLM with binarsity, and Distilltrees are not far apart (the difference in performance between the auto-calibrated version of Distilltrees and its non-auto-calibrated version (i.e. stopped at Step 2) is very small and not significant). The performance of GAM is significantly lower than that of its competitors for this case study. XGB performs better, but it is poorly calibrated, unlike the other four learning models (see Fig. 7).

In Fig. 8, we observe that the estimated most important functions are close for the four learning additive models for the variables V_2, V_{12}, V_{17}, and V_3. But for V_{15}, GAM's estimation is quite different from the other methods. GAM proposes higher values for the support of the variable and

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

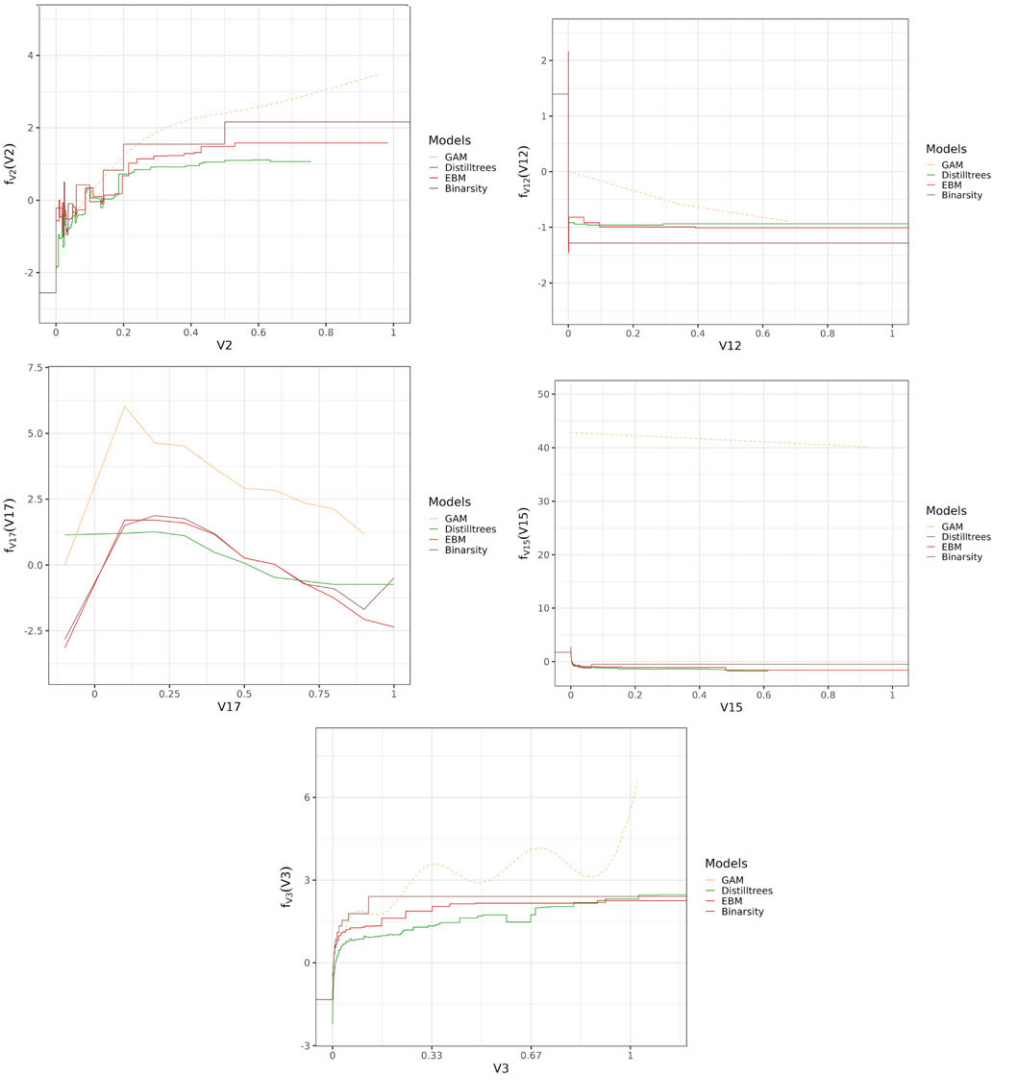

Figure 8. Feature shapes learned using GAM with splines, EBM, Distilltrees, and GLM with binarsity penalization.

much lower values near the end of the support. We also note that Distilltrees provides smoother trajectories than EBM and the GLM with binarsity.

These functions were passed on to Allianz Trade's experts, who were able to compare their business experience with the shapes of the estimated functions. They were fairly convinced of the form provided by Distilltrees for these most important variables. Interpretable credit insurance models are essential for risk management experts because they enhance transparency, facilitate understanding, enable model validation, support regulatory compliance, guide risk mitigation strategies, and improve overall risk management practices.

Distilltrees provides a competitive alternative to EBM and GLM with binarsity. The response functions obtained by the three approaches are not necessarily exactly the same. The intervals where the differences between Distilltrees and EBM or GLM with binarsity are most noticeable need to be studied in more detail by the user and crossed with other variables to

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

understand why the ATM used in Step 1 has led to these differences. It is a possibility of our approach to potentially identify interaction effects that might be overlooked with a simple additive form.

5. Conclusion

In this paper, we propose a new learning model for (not necessarily smooth) shape functions of a GAM model, named Distilltrees. It is based on the idea that it is possible to exploit the knowledge provided by an ATM to tailor the covariates of GLMs. By then using aggregation and an autocalibration procedure, we obtain a learning model as efficient as an EBM or as a regression model with a binarsity penalty. The advantage of our Distilltrees approach is that it does not require any additional hyperparameters, since its predictions are entirely deduced from the results of the ATM model. The purpose of Distilltrees is to capture some of the knowledge from an ATM model and represent it using additive, piecewise constant functions. While there is a clear relationship between the performance of the ATM and the model generated by Distilltrees, the objective is not to refine both simultaneously but rather to leverage the already optimized ATM model. In this context, the ATM serves as a starting point, and Distilltrees builds upon it to create a new model without needing further adjustments or fine-tuning of hyperparameters. This method meets a specific interest from Allianz Trade, the credit insurance company that supplied the data, which had an existing XGB model and was looking for a model that was almost as good but much easier to interpret.

Distilltrees is an interpretable model with excellent performance for tasks such as regression and classification. Users can choose their model for the ATM (random forest, gradient boosting, and variants), or even combine several models with stacking strategies and then use the Distilltrees procedure to obtain shape functions that they can then interpret with their own experience.

Acknowledgment. The authors thank Fabien Vinas, Global Head of Data Analytics & AI at Allianz Trade, for providing the database and for the insightful discussions on the use case. The authors also acknowledge two referees, an Associate Editor, and the Editor-in-Chief for their remarks and comments which greatly enhanced the relevance of this paper.

Data availability statement. The data and code that support the findings of Section 3 are available from Arthur Maillart (a.maillart@detralytics.eu) upon reasonable request.

Funding statement. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing interests. The authors declare none.

References

Alaya, **M. Z.**, **Bussy**, **S.**, **Gaïffas**, **S.**, & **Guilloux**, **A.** (2019). Binarsity: A penalization for one-hot encoded features in linear supervised learning. *Journal of Machine Learning Research*, **20**(118), 1–34. http://jmlr.org/papers/v20/17-170.html

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

Denuit, **M.**, **Charpentier**, **A.**, & **Trufin**, **J.** (2021). Autocalibration and tweedie-dominance for insurance pricing with machine learning. *Insurance: Mathematics and Economics*, **101**, 485–497.

Friedman, **J. H.** (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, **29**(5), 1189–1232.

Friedman, **J. H.**, & **Popescu**, **B. E.** (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, **2**(3).

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

Hastie, **T.**, & **Tibshirani**, **R.** (1986). Generalized additive models. *Statistical Science*, 297–310.

Henckaerts, **R.**, **Antonio**, **K.**, & **Côté**, **M.-P.** (2022). When stakes are high: Balancing accuracy and transparency with model-agnostic interpretable data-driven surrogates. *Expert Systems with Applications*, **202**, 117230. doi: 10.1016/j.eswa.2022.117230 .

Hofner, **B.**, **Mayr**, **A.**, & **Schmid**, **M.** (2014). gamboostlss: An r package for model building and variable selection in the gamlss framework. arXiv preprint arXiv:1407.1774.

Hooker, **G.** (2004). Discovering additive structure in black box functions. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 575–580).

Lindholm, **M.**, **Lindskog**, **F.**, & **Palmquist**, **J.** (2023). Local bias adjustment, duration-weighted probabilities, and automatic construction of tariff cells. *Scandinavian Actuarial Journal*, **2023**(10), 1–28.

Nori, **H.**, **Jenkins**, **S.**, **Koch**, **P.**, & **Caruana**, **R.** (2019). Interpretml: A unified framework for machine learning interpretability. arXiv preprint arXiv:1909.09223.

Tan, **S.**, **Caruana**, **R.**, **Hooker**, **G.**, **Koch**, **P.**, & **Gordo**, **A.** (2018). Learning global additive explanations for neural nets using model distillation.

Tsang, **M.**, **Cheng**, **D.**, & **Liu**, **Y.** (2017). Detecting statistical interactions from neural network weights. arXiv preprint arXiv:1705.04977.

Wood, **S. N.** (2017). *Generalized additive models: An introduction with R*. CRC Press.

Wüthrich, **M. V.**, & **Ziegel**, **J.** (2023). Isotonic recalibration under a low signal-to-noise ratio. arXiv preprint arXiv:2301.02692.

A. Appendix

In this section, we propose to continue the study begun in Section 3.1 using synthetic data to understand whether the assumption of independence between covariates is important for estimating shape functions, and more particularly when some covariates may be almost collinear. In Section 3.1, we assumed that $\mathbf{X}$ is a random vector whose components are independent and distributed according to the uniform distribution on $(-1, 1)$. Now we assume that $\mathbf{X} = (X_1, \ldots, X_8)$ with $X_i = 2\Phi(Y_i) - 1$, $i = 1, \ldots, 8$, where Φ is the cumulative distribution function of the standard Gaussian distribution and $\mathbf{Y} = (Y_1, \ldots, Y_8)$ is a random centered Gaussian vector with covariance matrix given by:

Table A.1 Comparison results based on R^2. Sample size : 50, 000

	Train R^2	Test R^2
GBM	96.33%	96.19%
GAM	96.55%	96.33%
EBM	96.21%	96.13%
Binarsity	96.12%	96.16%
Distilltrees	96.28%	96.28%

Table A.2 Comparison results based on R^2. Sample size : 1, 000

	Train R^2	Test R^2
GBM	97.54%	95.62%
GAM	96.55%	95.62%
EBM	97.44%	95.40%
Binarsity	96.83%	94.46%
Distilltrees	95.61%	95.67%

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

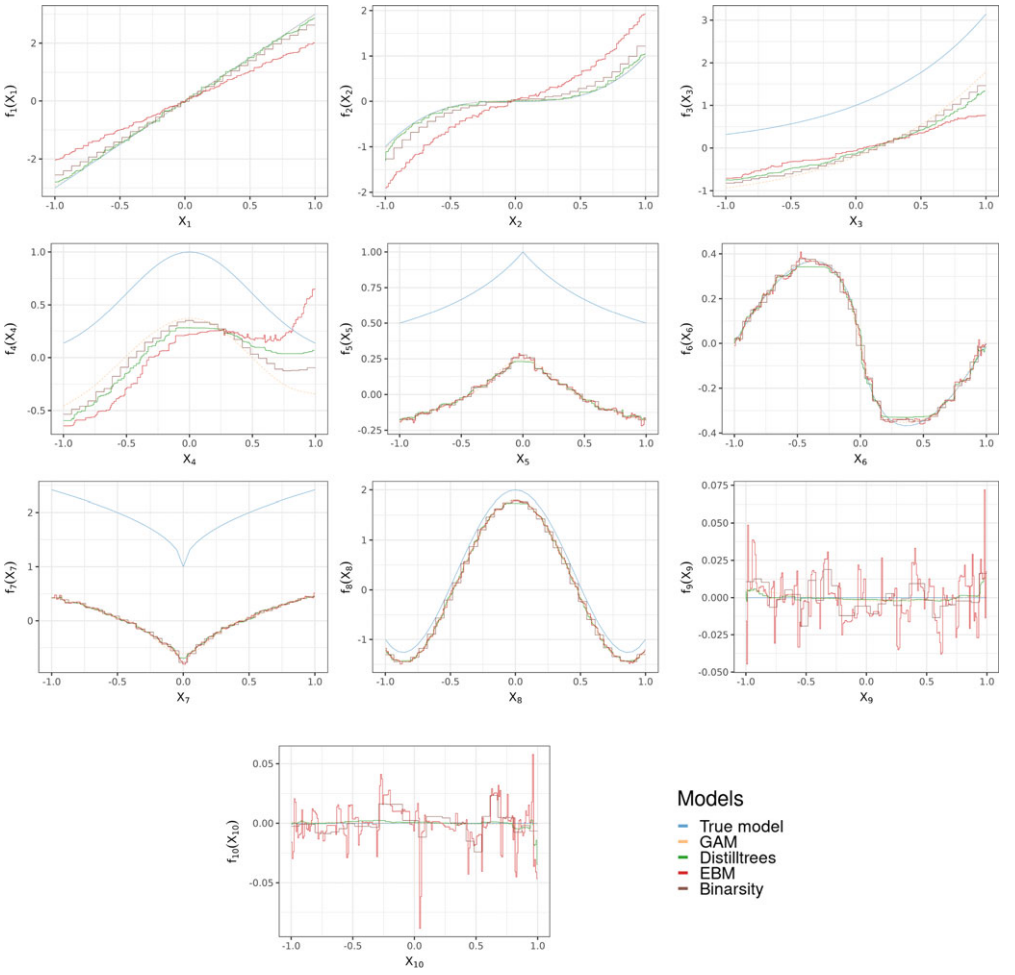

Figure A.1. Feature shapes learned using GAM with splines, EBM, Distilltrees, and GLM with binarsity penalization, with the presence of multicollinearity. Sample size : 50, 000.

$$\begin{pmatrix} 1 & 0.99 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.99 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.99 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.99 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The covariates X_1 and X_2 on the one hand and X_3 and X_4 on the other hand are therefore highly correlated and almost collinear. All the components of $\mathbf{X}$ have still uniform distributions on $(-1, 1)$.

We consider two sample sizes: first, $50,000$ as in Section 3.1 and second $1,000$ to study the robustness of the proposed algorithms for small sample sizes. For each model, we use the same

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

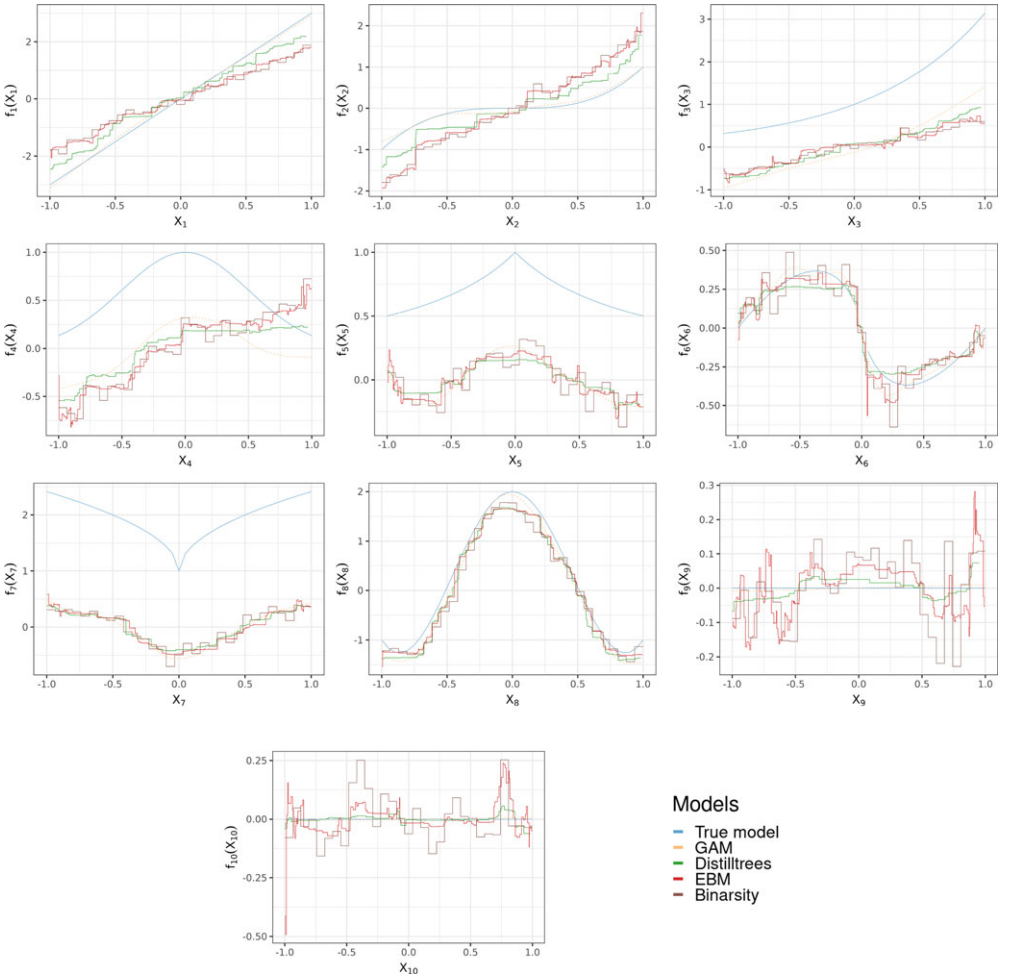

Figure A.2. Feature shapes learned using GAM with splines, EBM, Distilltrees, and GLM with binarsity penalization, with the presence of multicollinearity. Sample size : 1, 000.

packages as those presented in Section 3.1 and keep the same choices for the values of the hyper-parameters (see Table 2). The performances of the various models measured by the R^2 metric are given, respectively, in Table A.1 for the first sample size and in Table A.2 for the second sample size, while the estimated feature shapes are plotted, respectively, in Fig. A.1 for the first sample size and in Fig. A.2 for the second sample size.

For the sample size 50, 000, the performances are close to each other as in the independent case. However, for the sample size 1, 000, performances may differ depending on whether the train set or the test set is considered. We can see that all the algorithms except Distilltrees tend to overfit. The best performances on the test set are obtained for GAM with splines, GBM, and Distilltrees.

Unlike the independent case, the estimated shape functions can now include biases for the covariates X_1, X_2, X_3, and X_4 except for GAM with splines. For the sample size 50, 000, the largest biases are observed for EBM, while GLM with binarsity and Distilltrees have relatively small biases. For the sample size 1, 000, the shape functions are of course less well estimated and the biases are greater than for the sample size 50, 000. We can observe that Distilltrees has significantly smaller biases than EBM and GLM with binarsity.

https://doi.org/10.1017/S1748499524000241 Published online by Cambridge University Press

The presence of multicollinearity is a well-known issue for regression models. The most robust approach for limiting potential bias in additive models seems to be GAM with splines in the case where the response functions are assumed to be smooth. However, the most recent approaches based on machine learning algorithms can lead to significant biases. In our method, multicollinearity seems to be more effectively managed for small sample sizes because in the initial step the ATM constructs partitions in the space of explanatory variables that are more robust against multicollinearity.

Cite this article: Maillart A and Robert C (2024). Distill knowledge of additive tree models into generalized linear models: a new learning approach for non-smooth generalized additive models, *Annals of Actuarial Science*, 1–20. https://doi.org/10.1017/S1748499524000241