BLACK-BOX GUIDED GLM BUILDING WITH NON-LIFE PRICING APPLICATIONS

MATHIAS LINDHOLM* AND JOHAN PALMQUIST†, ‡

Abstract. The paper introduces a method for creating a categorical generalised linear model (GLM) based on information extracted from a given black-box predictor. The procedure for creating the guided GLM is as follows: For each covariate, including interactions, a covariate partition is created using partial dependence (PD) functions calculated based on the given black-box predictor. In order to enhance the predictive performance, an auto-calibration step is used to determine which parts of each covariate partition that should be kept, and which parts that should be merged. Given the covariate and interaction partitions, a standard categorical GLM is fitted using a lasso penalty.

The performance of the proposed method is illustrated using a number of real insurance data sets where gradient boosting machine (GBM) models are used as black-box reference models. From these examples it is seen that the predictive performance of the guided GLMs is very close to that of the corresponding reference GBMs. Further, in the examples, the guided GLMs have few parameters, making the resulting models easy to interpret. It is also seen that the guided GLMs often tend to be close to the corresponding GBMs in terms of fidelity, but there are examples where these differences are non-negligible.

Keywords: Feature extraction, Black-box models, Surrogate models, Regularisation, Auto-calibration

1. Introduction

Generalised linear models (GLMs) or general additive models (GAMs) are the standard benchmark models used in most non-life insurance pricing, see e.g. Ohlsson & Johansson (2010), Wüthrich & Merz (2023). These types of models are well-studied, transparent and, hence, easy to interpret, which is part of their popularity and widespread use in the decision making process. If one instead considers machine learning (ML) methods such as gradient boosting machines (GBMs) and neural networks (NNs), see e.g. Hastie et al. (2009) for a general introduction, and e.g. Denuit et al. (2020) and Wüthrich & Merz (2023), which also discusses actuarial applications, these type of methods tend to outperform GLMs and GAMs in terms of predictive accuracy. A potential problem, however, is that the predictors obtained when using ML-methods tend to be hard to interpret. In this short note we introduce a method for guided construction of a categorical GLM based on a given black-box predictor $\widehat{\mu}(x)$. From a practitioner perspective this is a very tractable approach, since categorical GLMs are well understood and are widely used for non-life insurance pricing, see e.g. Ohlsson & Johansson (2010). This approach is similar to the one introduced in Henckaerts et al. (2022), but our focus is not on maintaining fidelity w.r.t. the original predictor $\widehat{\mu}(x)$, but rather to find an as good categorical GLM as possible. For more on surrogate modelling, see e.g. Hinton et al. (2015), Henckaerts et al. (2022) and the references therein.

The general setup is that we observe (Z, X, W) data, where Z is the response, e.g. number of claims or claim cost, X is a d-dimensional covariate vector, and W is an exposure measure, e.g. policy duration. It will be assumed that Z, given X and W belongs to an exponential dispersion family (EDF), see e.g. Jørgensen & Paes De Souza (1994), Ohlsson & Johansson (2010), Wüthrich

* Department of Mathematics, Stockholm University; lindholm@math.su.se

† Länsförsäkringar Alliance, Stockholm, Sweden

‡ Department of Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden; palmquis@kth.se

& Merz (2023), which includes e.g. the Tweedie distribution. Further, it will be assumed that

$$\mathbb{E}[Z \mid X, W] = W\mu(X) \text{ and } \operatorname{Var}(Z \mid X, W) = W\sigma^2(X), \tag{1}$$

which is common to use in insurance pricing, see e.g. Ohlsson & Johansson (2010), Wüthrich & Merz (2023). Hence, if we let $Y := Z/W$, based on (1) it follows that

$$\mathbb{E}[Y \mid X, W] = \mu(X) = \mathbb{E}[Y \mid X] \text{ and } \operatorname{Var}(Y \mid X, W) = \frac{1}{W}\sigma^2(X). \tag{2}$$

When it comes to building a guided GLM based on an exogenous black-box predictor $\widehat{\mu}(x)$, the exposition will focus on at most two-way interactions, but the generalisation to higher order interactions is straight forward. Further, focus will be on log-linear models, but the assumption of using a log-link function can also be relaxed, and the procedure using other link-functions is analogous to the one described below. The suggested procedure can be summarised as follows: In a first step, start from a general d-dimensional covariate vector $x' := (x_1, \ldots, x_d)' \in \mathbb{X}$, $\mathbb{X} := \mathbb{X}_1 \times \cdots \times \mathbb{X}_d$, where $x_j \in \mathbb{X}_j, j = 1, \ldots, d$, and use a given mean predictor $\widehat{\mu}(x)$ to define categorical versions of the original covariates, x_j, and two-way interactions. This step uses partial dependence (PD) functions, see e.g. Friedman & Popescu (2008), to construct categories, or, equivalently, a partition of $\mathbb{X}_j$. This is the same idea used in Henckaerts et al. (2022), but instead of aiming for fidelity w.r.t. the original PD-function, the number of categories, or the size of the partition, is adjusted using an auto-calibration step, see e.g. Krüger & Ziegel (2021), Denuit et al. (2021). In this way focus is shifted from fidelity w.r.t. the initial predictor to accuracy of the new predictor, since the auto-calibration step will remove categories that do not contribute to the final predictor's predictive performance. In a second step, once the categorical covariates have been constructed, fit a standard categorical GLM with a mean function from (1) of the form

$$\mu(x;\beta) := \exp\left\{\beta_0 + \sum_{j=1}^{d}\sum_{k=1}^{\kappa}\beta_j^{(k)} 1_{\{x_j \in \mathbb{B}_j^{(k)}\}} + \sum_{i=1}^{d}\sum_{i<j}\sum_{k=1}^{\kappa}\beta_{i,j}^{(k)} 1_{\{(x_i,x_j)\in \mathbb{B}_{i,j}^{(k)}\}}\right\}, \tag{3}$$

where $\cup_{k=1}^{\kappa}\mathbb{B}_\bullet^{(k)} =: \mathbb{X}_\bullet$, and where the $\beta_\bullet^{(k)}$s are regression coefficients. Further, EDFs can be parametrised such that $\sigma^2(X) = \phi V(\mu(X))$, where ϕ is the so-called disperision parameter, and V is a variance function. Using this parametrisation together with the moment assumptions (1), gives us that the β-coefficients from (3) can be estimated using the deviance loss function

$$D(y;\beta,\lambda) := \sum_{i=1}^{n} w_i d(y_i, \mu(x_i;\beta)), \tag{4}$$

where $d(y,\mu)$ is the unit deviance function of an EDF, see e.g. Ohlsson & Johansson (2010), Wüthrich & Merz (2023), $\mu(x_i,\beta)$ is from (3).

The remainder of this short note is structured as follows: In Section 2 basic results on PD-functions are provided. Section 2.1 discusses implications and interpretations of using PD-functions, followed by Section 2.2, which describes how PD-functions can be used to partition the covariate space, both marginally and w.r.t. interaction effects, in this way creating categorical covariates. This section also describes how a marginal auto-calibration procedure can be used to remove possibly redundant categories. Section 3 discusses various implementational considerations and describes a full estimation procedure, which is summarised in Algorithm 1. The paper ends with numerical illustrations based on Poisson models applied to real insurance data, see Section 4, followed by concluding remarks in Section 5.

2. Partial dependence functions

The partial dependence function w.r.t. a, potentially exogenously given, (mean) function $\mu(x)$, $x' = (x_1, \ldots, x_d)' \in \mathbb{X}$, and the covariates $x_\mathcal{A}$, $\mathcal{A} \subset \{1, \ldots, d\}$, is given by

$$\operatorname{PD}(x_\mathcal{A}) := \int \mu(x_\mathcal{A}, x_{\mathcal{A}^\complement})\mathrm{d}\mathbb{P}(x_{\mathcal{A}^\complement}), \tag{5}$$

where $\mathcal{A}^{\mathrm{C}} = \{1, \ldots, d\} \setminus \mathcal{A}$, see e.g. Friedman & Popescu (2008). Note that (5) can be rephrased according to

$$\mathrm{PD}(x_{\mathcal{A}}) = \mathbb{E}[\mu(x_{\mathcal{A}}, X_{\mathcal{A}^{\mathrm{C}}})], \tag{6}$$

which illustrates that $\mathrm{PD}(x_{\mathcal{A}})$ quantifies the expected effect of $X_{\mathcal{A}} = x_{\mathcal{A}}$, when breaking all potential dependence between $X_{\mathcal{A}}$ and $X_{\mathcal{A}^{\mathrm{C}}}$, see Friedman & Popescu (2008). In particular, note that if $\mu(x) := \mathbb{E}[Y \mid X = x]$, the PD-function w.r.t. $\mathcal{A}$ is related to the expected effect of $\mathcal{A}$ on Y, when adjusting for potential association between $X_{\mathcal{A}}$ and $X_{\mathcal{A}^{\mathrm{C}}}$, see Zhao & Hastie (2021). Henceforth, all references to μ will, unless stated explicitly, treat μ as a conditional expected value of Y.

Remark 1.

(a) The PD-function (6) *w.r.t. a potentially exogenously given μ is expressed in terms of an unconditional expectation w.r.t. $X_{\mathcal{A}^{\mathrm{C}}}$. This is qualitatively different to*

$$\mu(x_{\mathcal{A}}) := \mathbb{E}[\mu(X_{\mathcal{A}}, X_{\mathcal{A}^{\mathrm{C}}}) \mid X_{\mathcal{A}} = x_{\mathcal{A}}], \tag{7}$$

which relies on the distribution of $X_{\mathcal{A}^{\mathrm{C}}} \mid X_{\mathcal{A}}$.

Further, note that the PD-function aims at isolating the effect of $X_{\mathcal{A}}$, when adjusting for potential association with the remaining covariates. This is not the case for (7)*, where effects in $x_{\mathcal{A}}$ could be an artefact of a strong association with (a subset of the covariates in) $X_{\mathcal{A}^{\mathrm{C}}}$.*

Another related alternative is to use accumulated local effects (ALEs), see Apley & Zhu (2020), which is closely connected to (7)*, but making use of a local approximation, and, hence suffers from similar problems as* (7)*. See also the discussion about PDs and ALEs in Henckaerts et al. (2022).*

(b) If the ambition is to construct a black-box guided (categorical) GLM model, it could be an alternative to apply the black-box model directly to subsets of covariates, i.e.

$$\mu(x_{\mathcal{A}}) := \mathbb{E}[Y \mid X_{\mathcal{A}} = x_{\mathcal{A}}],$$

but recall Remark 1(a), and see Remark 2 below. Also note that this will likely become computationally intensive, and the sub-models based on $\mu(x_{\mathcal{A}})$ are models that would not have been used in practice, and the models are not necessarily consistent with the original full model $\mu(x)$.

(c) In practice, when using PD-functions a potentially exogenous predictor μ can be evaluated without having access to the conditional distribution of $X_{\mathcal{A}^{\mathrm{C}}} \mid X_{\mathcal{A}}$, as opposed to (7)*.*

2.1. **Implications of partial dependence functions.** Consider the following log-linear additive model:

$$\mu(x) := \exp\left\{\beta_0 + \sum_{k=1}^{d} f_k(x_k) + \sum_{k=1}^{d}\sum_{j<k} f_{j,k}(x_j, x_k)\right\}, \tag{8}$$

where the fs are, e.g., basis functions. Hence, if we let $\mathcal{A} = \{j\}$, and introduce $x_{\setminus j} := x_{\mathcal{A}^{\mathrm{C}}}$, it follows that the PD based on (8) w.r.t. x_j reduces to

$$\begin{aligned}\mathrm{PD}(x_j) &= \exp\{f_j(x_j)\}\exp\{\beta_0\}\int \exp\left\{\sum_{i=1}^{d}\sum_{k\neq i} f_{i,k}(x_i, x_k)\right\} \mathrm{d}\mathbb{P}(x_{\setminus j}) \\ &= \exp\{f_j(x_j)\}\nu_{\setminus j}(x_j). \end{aligned} \tag{9}$$

Thus, the PD-function provides a marginalised effect of x_j, but it is not the same as $\exp\{f_j(x_j)\}$. Still, the changes in the PD-function w.r.t. x_j are related to changes in the jth dimension of $\mu(x)$, when adjusting for possible dependence between X_j and $X_{\setminus\{j\}}$, see Remark 1(a). This is also in line with the critique against using PD-functions, in favour of using ALEs, in Apley & Zhu (2020). Note, however, as discussed in the introduction, for our purposes the PD-function is only used to obtain covariate partitions, so that whether the absolute *level* of a marginal effect is correct or not, is of considerably less importance. We will come back to this discussion when describing how to construct covariate partitions in Section 2.2, see also Remark 2(a) below.

Further, note that if $f_{j,k}(\cdot) = 0$ for all j, k it follows that

$$\mathrm{PD}(x_j) \propto \exp\{f_j(x_j)\}. \tag{10}$$

That is, when there are no interaction effects, the PD w.r.t. x_j will, under model specification (8), retrieve the true direct effect of x_j up to scaling.

Analogously, if we instead consider bivariate PD-functions and consider $\mathcal{A} = \{j, k\}$, with $x_{\setminus\{j,k\}} := x_{\mathcal{A}^\subset}$, it follows that

$$\mathrm{PD}(x_j, x_k) = \exp\{f_j(x_j) + f_k(x_k) + f_{j,k}(x_j, x_k)\}\nu_{\setminus\{j,k\}}(x_j, x_k), \tag{11}$$

which neither retrieves the correct bivariate interaction (up to scaling), unless there are no direct effects w.r.t. x_j and x_k, and there are no other interaction effects including either of x_j and x_k, but again, recall Remark 1(a).

Similar relations hold for other link-functions than the log-link, but in this short note focus will be on the log-link function.

2.2. **Covariate engineering, PD-functions, and marginal auto-calibration.** As discussed when introducing the expectation representation of the PD-function in (6), see also Remark 1(a), the PD-function of $X_{\mathcal{A}}$ aims at isolating the expected effect of $X_{\mathcal{A}}$, when adjusting for potential influence from $X_{\mathcal{A}^\subset}$. This suggests to use PD-functions for covariate engineering w.r.t. individual covariates, which allows us to partition the covariate space and, ultimately, construct a data driven categorical GLM. That is, if $\mathrm{PD}(x_j) \in B$ we can construct the corresponding covariate set on the original covariate scale according to

$$x_j \in \mathbb{B} := \{x_j^* \in \mathbb{X}_j : \mathrm{PD}(x_j^*) \in B\}.$$

This allows us to use the PD-function to partition $\mathbb{X}_j$, based on where X_j is similar in terms of PD-function values, which can be generalised to tuples of covariates.

In order to construct a partition based on PD-functions, consider a sequence of $b_j^{(k)}$s such that

$$-\infty =: b_j^{(0)} < b_j^{(1)} < \ldots < b_j^{(\kappa-1)} < b_j^{(\kappa)} := +\infty, \tag{12}$$

and set $B_j^{(k)} := (b_{j,k-1}, b_{j,k}]$, i.e. $\cup_{k=1}^{\kappa} B_j^{(k)} = \mathbb{R}$. The corresponding partition of $\mathbb{X}_j$, denoted $\Pi_j := (\mathbb{B}_j^{(k)})_{k=1}^{\kappa}$, is defined in terms of the parts

$$\mathbb{B}_j^{(k)} := \{x_j^* \in \mathbb{X}_j : \mathrm{PD}(x_j^*) \in B_k\},\ k = 1, \ldots, \kappa. \tag{13}$$

That is, $\cup_{k=1}^{\kappa} \mathbb{B}_j^{(k)} = \mathbb{X}_j$. Thus, without having specified how to obtain a partition of the real line according to (12), including both the size of the partition and the location of split points, it is clear that given such a partition the procedure outlined above can be used to construct a categorical GLM in agreement with (3). Moreover, note that (13) allows us to introduce an auxiliary categorical covariate $\overline{X}_j$, which is a categorical version of X_j:

$$\overline{X}_j := \overline{X}_j(X_j) := \sum_{k=1}^{\kappa} k 1_{\{X_j \in \mathbb{B}_j^{(k)}\}}. \tag{14}$$

The use of categorical covariates $\overline{X}_j$ will simplify the exposition in Setion 3.

Further, if the ambition is to construct a categorical GLM with good predictive accuracy it is reasonable to only keep the parts in the partition Π_j that actually impacts the response. One way to achieve this is to use a marginal auto-calibration step: Define

$$\overline{\mu}_j^{(k)} := \mathbb{E}[Y \mid X_j \in \mathbb{B}_j^{(k)}],\ k = 1, \ldots, \kappa, \tag{15}$$

and introduce the following piece-wise constant mean predictor

$$\overline{\mu}_j(X_j) := \sum_{k=1}^{\kappa} \overline{\mu}_j^{(k)} 1_{\{X_j \in \mathbb{B}_j^{(k)}\}},\ \overline{\mu}_j^{(k)} \in \mathbb{R}. \tag{16}$$

By using the $\overline{\mu}_j^{(k)}$s from (15) it is possible to compare how the mean predictions change as a function of the parts in the partition defined by the $\mathbb{B}_j^{(k)}$s. Thus, the $\overline{\mu}_j$s from (15) can be used to

remove, or merge, $\mathbb{B}_j^{(k)}$s in the partition that lacks an isolated impact on the response. The output of such a procedure is again a partition.

Further, note that

$$\overline{\mu}_j(X_j) = \mathbb{E}[Y \mid \overline{\mu}_j(X_j)], \tag{17}$$

which follows directly from the Tower property, where (17) precisely corresponds to that $\overline{\mu}_j(X_j)$ is auto-calibrated, see Krüger & Ziegel (2021), Denuit et al. (2021). A consequence of this is that, given the information contained in $\overline{\mu}_j(X_j)$, the predictor can not be improved upon.

This procedure is analogously defined for tuples of covariates, and a precise implementation is described in Section 3.

Remark 2.

(a) If we consider a numerical covariate, the idea of using a PD-function to construct a covariate partition is only relevant when the PD-function is not monotone, since otherwise we could just as well partition the covariate directly based on, e.g., quantile values. Note that this comment, of course, if we would change from using PD-functions to using, e.g., ALEs or some other covariate effect measure.

Further, from the above construction it is clear that the PD-function is only used to construct covariate partitions. That is, the actual impact on the response, here measured in terms of PD-functions values, is of lesser importance, as long as the PD-function changes when the covariate values change. Consequently, it is the sensitivity of the measure being used, here PD-functions, that matters, not the level, where the latter is the primary critique for using PD-functions instead of, e.g., ALEs, see Apley & Zhu (2020) and the discussion in Henckaerts et al. (2022). Also recall Remark 1(a) above.

(b) Note that the output of the auto-calibration step (17) *is not a new PD-function, but a conditional expected value. Still, the partitioning will be based on similarity in terms of PD-function values, but those parts in the partition that do not effect the response will be removed. This is believed to be beneficial, since the ambition is to construct a guided categorical GLM with good predictive performance. If one instead favour models with as high fidelity w.r.t. the original black-box predictor, i.e. a so-called surrogate model, see e.g. Henckaerts et al. (2022), the auto-calibration step is problematic for, e.g., ordered categories, since the merging of categories does not respect ordering. The corresponding step in the algorithm of Henckaerts et al. (2022), see their Algorithm 1, merge categories only based on fidelity to the original PD-function, see their equation (2). Also note that for numerical and ordinal covariates the procedure in Henckaerts et al. (2022) only merge PD-function values that have adjacent covariate values.*

(c) Recall Remark 1(a) and note that if we would replace the $\mathrm{PD}(x_j)$ *with* $\mu(x_j)$*, it by construction holds that*

$$\mu(X_j) = \mathbb{E}[Y \mid \mu(X_j)],$$

if $\mu(x_j)$ *is the true mean function. This choice, however, is likely computationally demanding and not practically feasible based on a finite sample, see Lindholm et al. (2023).*

Further, also recall from Remark 1(a), by using $\mu(x_j)$ *we are no longer targeting the isolated effect of* x_j*, due to the possible dependence with the remaining covariates.*

3. Constructing a guided categorical GLM

The first step in creating a guided GLM is to calculate the PD-function values from the external black-box predictor $\widehat{\mu}(x)$. For each covariate dimension j this is done based on either κ equidistributed quantile values if the jth covariate is continuous, or at all categorical levels if the jth covariate dimension is categorical. This defines a partition $\Pi_j = (\mathbb{B}_j^{(k)})$ of $\mathbb{X}_j$.

Given a candidate partition Π_j, evaluate if some parts of the partition should be merged using marginal auto-calibration as discussed in Section 2.2. A simple way to do this is to use regression trees: A regression tree with κ terminal nodes (or leafs), denoted $T(x)$, is defined according to

$$T(x) := \sum_{k=1}^{\kappa} \delta_k 1_{\{x \in \mathbb{G}_k\}}, \ \delta_k \in \mathbb{R}, \tag{18}$$

where $\cup_{k=1}^{\kappa} \mathbb{G}_k =: \mathbb{X}$, see e.g. Hastie et al. (2009), which is of the same form as $\overline{\mu}_j$ from (16). In particular, recall that $\overline{x}_j$ from (14) is a compact way to encode the PD-function partitions, which allows (16) to be re-written according to

$$\overline{\mu}_j(\overline{x}_j) := \sum_{k=1}^{\kappa} \overline{\mu}_j^{(k)} 1_{\{\overline{x}_j = k\}}.$$

Further, in this short note we will use L^2-regression trees estimated using square loss in a greedy manner, using cross-validation (CV), see e.g. Hastie et al. (2015). That is, if we use the categorical representation of x_j, i.e. $\overline{x}_j$ from (14), the empirical loss that will be (greedely) minimised is given by

$$\widehat{\overline{\mu}}_j(\overline{x}_j) := \operatorname*{arg\,min}_{T \in \mathcal{T}_\kappa} \sum_{i=1}^{n} w_i (y_i - T((\overline{x}_j)_i))^2, \tag{19}$$

where $(\overline{x}_j)_i$ denotes the ith observation of the $\overline{x}_j$th categorical covariate, where the w_i weights have been added in order to agree with the GLM assumptions from (1), and where $\mathcal{T}_\kappa$ corresponds to the set of binary regression trees with at most κ terminal nodes. Thus, by minimising (19) redundant levels in the categorical covariate $\overline{x}_j$ will be merged. Recall that this is equivalent to merging redundant $\mathbb{B}_j^{(k)}$s, and this produces an updated, possibly reduced, partition. Further, the motivation of using L^2-trees instead of, e.g., a Tweedie loss is because all Tweedie losses that are special cases of the Bregman deviance losses, see Denuit et al. (2021), result in the same mean predictor for a given partition $\mathbb{B}_k$, see e.g. Lindholm & Nazar (2023). In particular, note that the resulting $\widehat{\overline{\mu}}_j$s correspond to empirical means, regardless of the Tweedie loss functions used, hence making this step model-free. For alternatives to using L^2-regression trees to achieve auto-calibration, see e.g. Denuit et al. (2021), Wüthrich & Ziegel (2023).

If the procedure from Section 2.2 is applied to all covariates and interactions, the resulting number of categorical levels, and, hence, β coefficients to be estimated in (3) can become very large. This suggests that regularisation techniques should be used when fitting the final categorical GLM. One way of achieving this is to use L^1-regularisation, or so-called lasso-regularisation, see e.g. Hastie et al. (2015). If we consider EDF models, this means that we, given the $\mathbb{B}_\bullet^{(k)}$s, use the following penalised deviance loss function

$$D(y; \beta, \lambda) := \sum_{i=1}^{n} w_i d(y_i, \mu(x_i; \beta)) + \lambda |\beta|, \tag{20}$$

which is the loss from (4), but where the L^1-penalty term $\lambda|\beta|$ has been added, where λ is the penalty parameter. The λ-parameter is chosen using k-fold CV.

Moreover, if the covariate vector x is high-dimensional it can be demanding already to evaluate all two-way interactions fully. An alternative is here to consider only those two-way interactions that are believed to have an impact on the final model. This can be achieved by using Friedman's H-statistic, see Friedman & Popescu (2008):

$$H_{j,k} = \frac{\widehat{\mathbb{E}}[(\mathrm{PD}(X_j, X_k) - \mathrm{PD}(X_j) - \mathrm{PD}(X_k))^2]}{\widehat{\mathbb{E}}[\mathrm{PD}(X_j, X_k)^2]}, \tag{21}$$

where $\widehat{\mathbb{E}}[\cdot]$ refers to the empirical expectation. That is, (21) provides an estimate of the amount of excess variation in $\mathrm{PD}(X_j, X_k)$ compared with $\mathrm{PD}(X_j) + \mathrm{PD}(X_k)$.

By combining all of the above, focusing on a categorical GLM with at most two-way interactions, we arrive at Algorithm 1. Of course, if two-way interactions turn out to be insufficient, the procedure can be extended analogously to consider higher order interactions as well.

Remark 3.

(a) Note that there is a qualitative difference between using L^2-trees, or other deviance based binary trees, and using L^1-penalisation: Trees merge categories (parts in a partition), whereas using an L^1-penalty will remove categories, or, equivalently, merge removed categories with a global intercept.

(b) The L^1-penalty from (20) has a single λ applied to all β-coefficients. An alternative is to use a grouped penalty, see e.g. Hastie et al. (2015). That is, one could, e.g., use one λ-penalty for individual covariates and one λ for interaction terms, see e.g. Henckaerts et al. (2022).

Algorithm 1 – Guided GLM

Input.
- Black-box mean function $\widehat{\mu}$
- Observed i.i.d. training data $(y_i, x_i, w_i)_{i=1}^n$
- κ denote the maximum grid range for PD-function
- γ denote the number of interaction terms
- θ_{tree} denote hyperparameters for regression trees

A. Marginal effects

For each dimension j of x

Initial marginal effect: Compute $\text{PD}(x_j)$ based on $\widehat{\mu}(x)$ at each categorical level or at κ^* equally sized quantiles of x_j and construct the candidate categorical version of x_j, $\overline{x}_j$, from (14). Let $\kappa_j^* := |\mathbb{X}_j|$ if $\mathbb{X}_j$ is categorical and finite, otherwise $\kappa_j^* = \kappa$

Auto-calibration: Merge possibly redundant categories in $\overline{x}_j$ by fitting an L^2-regression tree, $\widehat{\overline{\mu}}_j(\overline{x}_j)$, with at most $\max(\kappa, \kappa_j^*)$ terminal nodes according to (19) and hyperparameters θ_{tree}

Output marginal partition: Extract covariate partition $\Pi_j := (\mathbb{B}_j^{(k)})_{k=1}^{\kappa}$ from $\widehat{\overline{\mu}}_j(\overline{x}_j)$

B. Interaction effects

Calculate the Friedman H-statistic for all factor combinations according to (21)

For the factor combinations (x_j, x_l) with the γ highest scores

Initial interaction effect: Compute $\text{PD}(x_j, x_l)$ based on $\widehat{\mu}(x)$ at each integer κ_j^* / κ_l^* / κ level combination of (x_j, x_l) and construct the candidate categorical version of (x_j, x_l), $\overline{x}_{j,l} := \overline{x}_{j,l}(x_j, x_l)$, in analogy with (14)

Auto-calibration: Merge possibly redundant categories in $\overline{x}_{j,l}$ by fitting an L^2-regression tree, $\widehat{\overline{\mu}}_{j,l}(\overline{x}_{j,l})$, with at most $\max(\kappa\kappa_j^*, \kappa\kappa_l^*, \kappa_j^*\kappa_l^*, \kappa^2)$ terminal nodes in analogy with (19) and hyperparameters θ_{tree}.

Output interaction partition: Extract interaction partition $\Pi_{j,l} := (\mathbb{B}_{j,l}^{(k)})_{k=1}^{\kappa}$ from $\widehat{\overline{\mu}}_{j,l}(\overline{x}_{j,l})$

C. Final model

Use the marginal partitions Π_j, from **A.**, and the $\Pi_{j,l}$ interaction partitions, from **B.** to define the structure of the categorical GLM given by (3). Estimate the β-coefficients from (3) using the L^1 penalised deviance from (20). The value of λ is obtained using k-fold CV.

4. Numerical illustrations

In the current section we will construct guided categorical GLMs based on reference models that are GBMs, following the procedure described in Algorithm 1, using the `freMTPL`, `beMTPL`, `auspriv`, and `norauto` data sets available in the `R`-package `CASdataset`, see Dutang & Charpentier (2020). Only Poisson claim count models will be considered, i.e. the Poisson deviance

$$D_{\text{Pois}}(y;\mu) := \sum_{i=1}^{n} w_i \left(y_i \log(y_i) - y_i \log(\mu_i) - y_i - \mu_i\right), \tag{22}$$

will be used for model estimation and prediction evaluation. Concerning data, for all data sets analysed 2/3 of the data have been used for in-sample training, and 1/3 for out-of-sample (hold out) evaluation.

Further, all GBM models use a tree depth of two, 0.01 learning rate and a bag fraction of 0.75 corresponding to the fraction of training data used for each tree iteration. The maximum number of trees is set to 4 000 with the optimal number chosen via 5-fold cross validation and the remaining hyperparameters are the default levels in the `R`-package `GBM`. Hence, hyperparameters for the GBM modelling are the same as those used in Henckaerts et al. (2022), as described in their section 3.2.1.

When implementing Algorithm 1 the number of interaction terms is set to 5 (γ) and the maximum grid size for the PD-functions is set to 30 (κ). Concerning the hyperparameters for the L^2-trees (θ_{tree}), the minimum bucket size is set to 10 and the cost penalty parameter is set to 0.00001 in order to allow for very deep un-pruned trees, after which the optimal tree size, including pruning, is determined using cross validation as implemented according to the `rpart`-package in `R`.

From Algorithm 1 it is clear that there is no ambition to replicate the PD-functions of the initial model, which here is a GBM. An example of PD-functions for the different models for the `freMTPL`-data is given in Figure 1. From Figure 1 it is also seen that the GBM's PD-functions are monotone for the covariates "Vehicle age" and "Bonus Malus", which, hence could have been adjusted directly using an L^2 tree, see Remark 2(a). Moreover, from Figure 1 it is also seen that the number of categories in the guided categorical GLM is reduced by using a final lasso (L^1) step in Algorithm 1. Further, the number of active parameters in the final guided categorical GLM are summarised in Table 1, and it can be noted that the number of parameters tends to be very low.

Furthermore, Table 1 shows the fidelity of the guided categorical GLM w.r.t. the original GBM model, where fidelity is defined as the correlation between the initial GBM mean predictor and the corresponding guided categorical GLM predictor. From, this it is seen that fidelity tends to be rather high for the data sets being analysed, with no fidelity less than 88%. These numbers, however, tend to deviate considerably for `freMTPL` and `beMTPL` compared to the surrogate model of Henckaerts et al. (2022), see their Table 5. This could, at least, partly be caused by the use of different seeds, or other model differences. A more detailed comparison of the differences of the two predictors is seen in the scatter plots provided in Figure 2, which agree with the fidelity calculations.

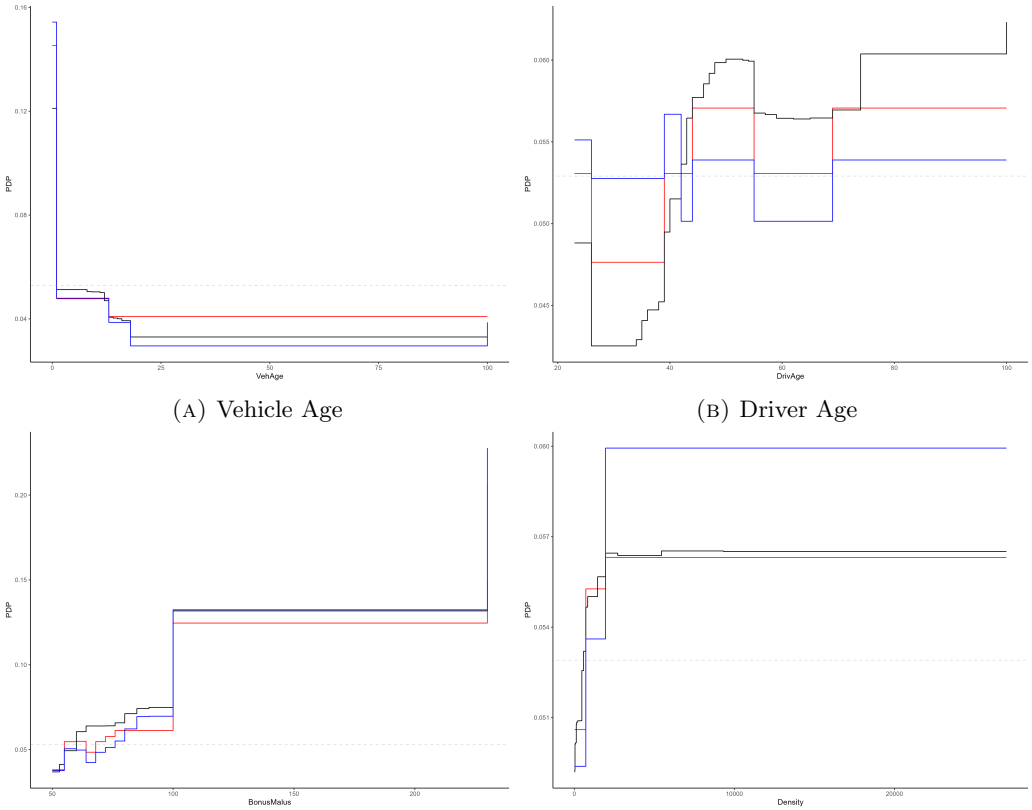
Continuing, in order to compare the predictive performance of the guided categorical GLM and the reference GBM, we calculate the out-of-sample relative difference in Poisson deviance, ΔD_{Pois}, defined according to

$$\Delta D_{\text{Pois}} := \frac{D_{\text{Pois}}(y; \widehat{\mu}^{\text{GLM}^\star}) - D_{\text{Pois}}(y; \widehat{\mu}^{\text{GBM}})}{D_{\text{Pois}}(y; \widehat{\mu}^{\text{GBM}})}, \tag{23}$$

where $D_{\text{Pois}}(y;\mu)$ is given by (22), and where the gudied GLM is denoted by $\text{GLM}^\star$. From Table 1 it is seen that the ΔD_{Pois} values for the different data sets are very small indicating that the guided categorical GLMs tends to track the performance of the initial GBMs closely. One can also note that the guided categorical GLM in fact outperforms the corresponding GBMs for the `beMTPL` and `auspriv` data sets, although these results could be due to random fluctuations. Further, by comparing with the surrogate model from Henckaerts et al. (2022), see their Table 4, their relative Poisson deviances are comparable to those in Table 1. It is, however, worth noting that the guided categorical GLMs with the lowest fidelity, `beMTPL` and `freMTPL`, are the ones that also differ the most compared with Henckaerts et al. (2022), in favour for the current guided GLM. Still, as commented on above, the observed differences could, at least partly, be due to not using the same seed or other model differences. Continuing, the relative Poisson deviance values provide

Data	No. of parameters	ΔD_{Pois}	Fidelity
`norauto`	2	0.11%	100%
`beMTPL`	90	-0.29%	88%
`auspriv`	2	-0.04%	98%
`freMTPL`	49	0.58%	89%

TABLE 1. Summary statistics for the different data sets, where ΔD_{Pois} is defined in (23), and where fidelity refers to the correlation between the GBM predictor and the corresponding guided categorical GLM.

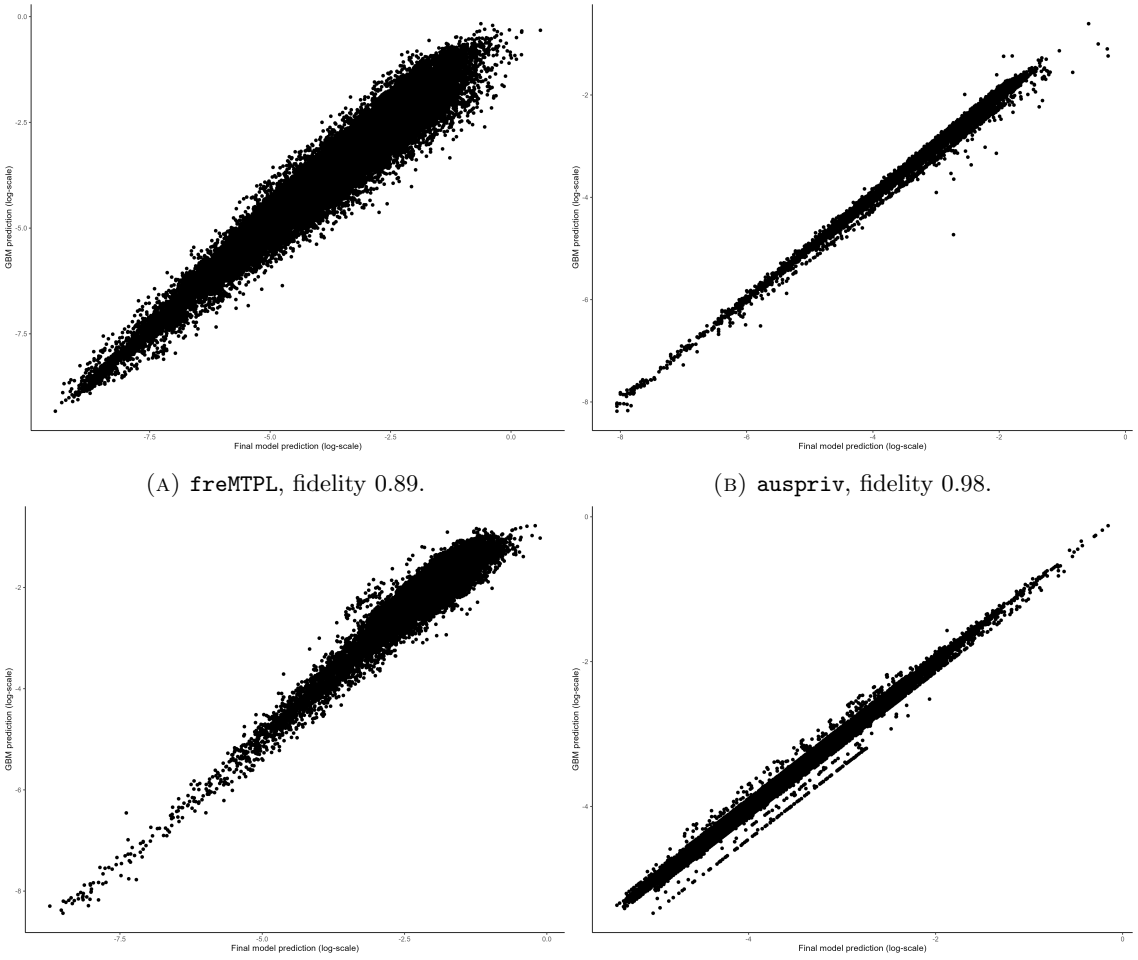


(A) Vehicle Age (B) Driver Age

(C) Bonus Malus (D) Density

FIGURE 1. Comparison of model factor effects (PDPs) for the `freMTPL` data between initial GBM-model (black lines), guided categorical GLM including final lasso (L^1) step (red lines) and a model including all levels found by the tree-calibration (blue lines).

a summary of the overall out-of-sample performance. In order to asses local performance of the mean predictors, we use concentration curves, see e.g. Denuit et al. (2019), see Figure 3. From Figure 3 it is again seen that the local performance of the mean predictors of the guided categorical GLMs are comparable to the corresponding GBMs' performance.



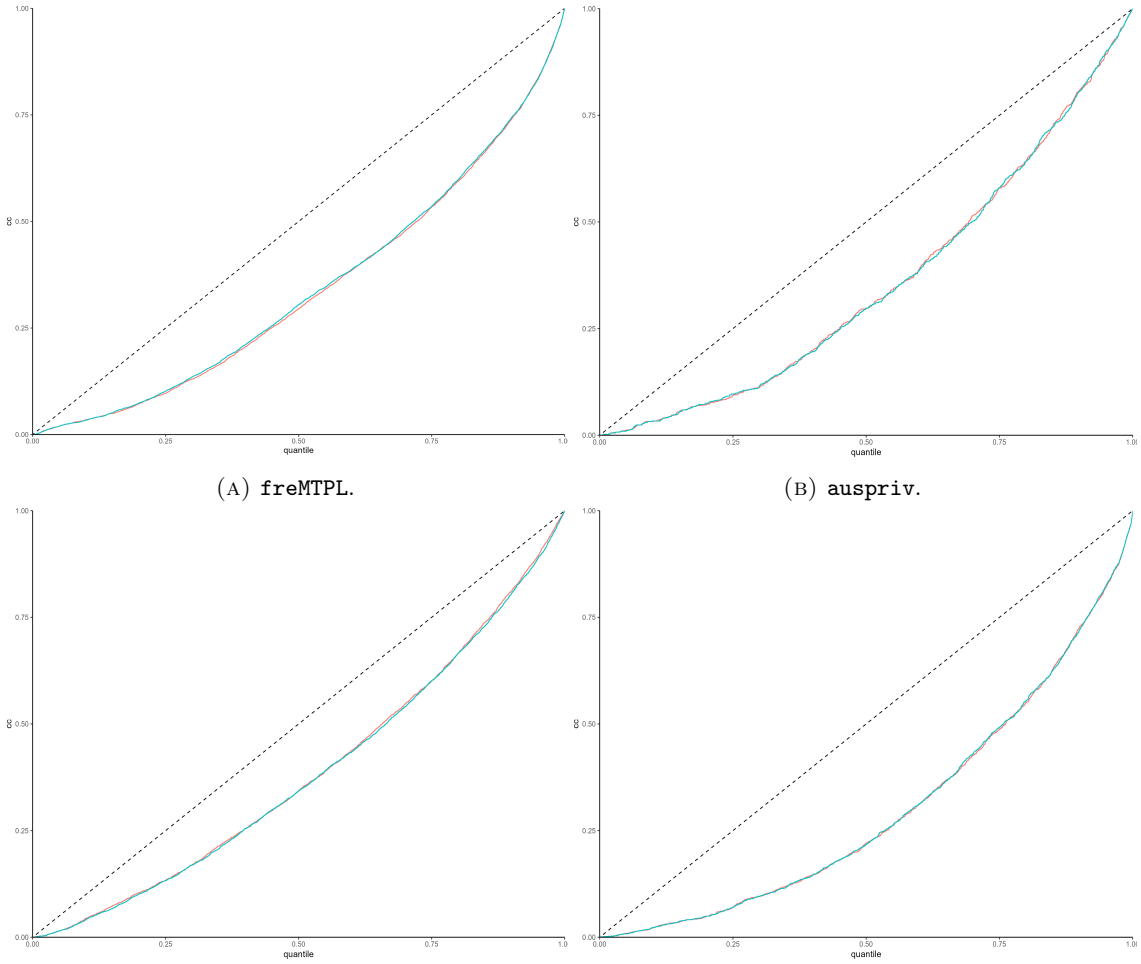
(A) `freMTPL`, fidelity 0.89.

(B) `auspriv`, fidelity 0.98.

(C) `beMTPL`, fidelity 0.88.

(D) `norauto`, fidelity 1.00.

FIGURE 2. Scatter plots for different `CASDatasets` data, comparing the original GBM models and the corresponding guided categorical GLMs. Fidelity corresponds to the correlation between the two predictors.



(A) `freMTPL`.

(B) `auspriv`.

(C) `beMTPL`.

(D) `norauto`.

FIGURE 3. Concentration curves for different `CASDatasets` data comparing the original GBM models (red lines) and the corresponding guided categorical GLM (blue lines).

5. Concluding remarks

In the current short note we introduce a simple procedure for constructing a categorical GLM making use of implicit covariate engineering within a black-box model, see Algorithm 1. The resulting model is referred to as a guided categorical GLM. The central part of the modelling aims at identifying how single covariates (and interactions) impact the response. This is here done using PD-functions together with a marginal auto-calibration step in order to construct covariate partitions. The rationale behind this procedure is as follows: The PD-functions are used to assess the impact of a covariate w.r.t. the initial black-box *predictor* and in this way generate candidate covariate partions. Given a partition, by using marginal auto-calibration only the parts in the candidate partition that have an impact on the *response* will remain, regardless of the underlying black-box model. Consequently, as long as the PD-functions are able to differentiate between covariate values, the actual *level* of the PD-functions are not important, and the PD-functions can be replaced with any other meaningful covariate effect measures, such as ALEs. Further, note that if the PD-functions, or equivalent effect measures, are applied to numerical or ordinal covariates, and the resulting function is monotone, the suggested procedure could just as well be replaced by binning the covariates based on, e.g., their quantile values, see Remark 2(a).

The above procedure is closely related to the modelling approach introduced in Henckaerts et al. (2022), where the main difference is that they aim for fidelity w.r.t. the (PD-function) behaviour of the original black-box predictor. The guided categorical GLM, on the other hand, focuses on high predictive accuracy. Although the two approaches will be close if the PD-functions are monotone, the numerical illustrations show situations where the guided categorical GLMs reduction in fidelity coincides with an increase in predictive performance. This also connects to the wider discussion on the use of auto-calibration and (complex) black-box predictors in non-life insurance pricing, see e.g. Lindholm et al. (2023), Wüthrich & Ziegel (2023). In these references it is noted that a low signal to noise ratio, which is common in non-life insurance data, may result in complex predictors that are spuriously smooth. In their examples, by applying the auto-calibration techniques in Lindholm et al. (2023), Wüthrich & Ziegel (2023) to a complex predictor, the resulting auto-calibrated predictor only has a few unique *predictions*; in the examples around 100 unique predictions. This is still considerably less than the current guided GLMs' predictors that use up to 90 *parameters*, see Table 1 in Section 4 above. Consequently, if the number of parameters in the guided categorical GLM is not too large it may be possible to construct a new interpretable categorical GLM that is auto-calibrated by using the techniques from, e.g., Lindholm et al. (2023), Wüthrich & Merz (2023).

Acknowledgments

M. Lindholm gratefully acknowledges financial support from Stiftelsen Länsförsäkringsgruppens Forsknings- och Utvecklingsfond [project P9/20 "Machine learning methods in non-life insurance"]. J. Palmquist gratefully acknowledges the support from Länsförsäkringar Alliance. All views expressed in the paper are the authors own opinions and not necessarily those of Länsförsäkringar Alliance.

References

Apley, D. W. & Zhu, J. (2020), 'Visualizing the effects of predictor variables in black box supervised learning models', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **82**(4), 1059–1086.

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

Denuit, M., Hainaut, D. & Trufin, J. (2020), *Effective Statistical Learning Methods for Actuaries II Tree-Based Methods and Extensions*, 1st ed. 2020. edn, Springer International Publishing, Cham.

Denuit, M., Sznajder, D. & Trufin, J. (2019), 'Model selection based on lorenz and concentration curves, gini indices and convex order', *Insurance: Mathematics and Economics* **89**, 128–139.

Dutang, C. & Charpentier, A. (2020), 'Software package `CASdatasets`'.
URL: *http://cas.uqam.ca/pub/web/CASdatasets-manual.pdf*

Friedman, J. H. & Popescu, B. E. (2008), 'Greedy function approximation: a gradient boosting machine', *Predictive Learning via Rule Ensembles* **2**(3), 916–954.

Hastie, T., Tibshirani, R. & Friedman, J. H. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer.

Hastie, T., Tibshirani, R. & Wainwright, M. (2015), *Statistical learning with sparsity: the lasso and generalizations*, CRC press.

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.

Hinton, G., Vinyals, O. & Dean, J. (2015), 'Distilling the knowledge in a neural network', *arXiv preprint, arXiv:1503.02531* .

Jørgensen, B. & Paes De Souza, M. C. (1994), 'Fitting tweedie's compound poisson model to insurance claims data', *Scandinavian Actuarial Journal* **1994**(1), 69–93.

Krüger, F. & Ziegel, J. F. (2021), 'Generic conditions for forecast dominance', *Journal of Business & Economic Statistics* **39**(4), 972–983.

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

Lindholm, M. & Nazar, T. (2023), 'On duration effects in non-life insurance pricing', *SSRN preprint, 4474908* .

Ohlsson, E. & Johansson, B. (2010), *Non-Life Insurance Pricing with Generalized Linear Models*, EAA Series, Springer Berlin Heidelberg.

Wüthrich, M. V. & Merz, M. (2023), *Statistical foundations of actuarial learning and its applications*, Springer Nature.

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

Zhao, Q. & Hastie, T. (2021), 'Causal interpretations of black-box models', *Journal of Business & Economic Statistics* **39**(1), 272–281.