Grabit: Gradient Tree-Boosted Tobit Models for Default Prediction

Fabio Sigrist*
Lucerne University of Applied Sciences and Arts
and
Christoph Hirnschall
Advanon

August 16, 2018

Abstract

In this article, a novel binary classification approach for data with (i) class imbalance between a minority and a majority class and (ii) a sample size that is small in relation to the complexity of the decision boundary is presented. We introduce a novel model – the Grabit model – which is obtained by applying gradient tree boosting to the Tobit model and show how this model can leverage auxiliary data to better learn the decision function for imbalanced data and, thus, obtain increased predictive accuracy. We apply the Grabit model for predicting defaults on loans made to Swiss small and medium-sized enterprises (SME) and obtain a large and significant improvement in predictive performance compared to other state-of-the-art approaches.

Keywords: Classification, class imbalance, credit scoring, bankruptcy prediction, censored regression

*Corresponding author. Email: fabio.sigrist@hslu.ch. Address: Lucerne School of Business, Grafenauweg 10, 6302 Zug, Switzerland.
1 Introduction

Prediction of corporate failures is important since bankruptcies can result in significant economic losses for investors and even cause economic downturns and recessions. As a result, default prediction has been of major interest to both researchers and practitioners in the financial sector for almost a century [FitzPatrick, 1932, Winakor and Smith, 1935, Merwin et al., 1942]. Statistical modeling of default began in the late 1960s with the seminal papers of Beaver [1966] and Altman [1968]. See Altman [2002] and Bellovary et al. [2007] for surveys on default prediction. Statistical default prediction is traditionally approached using binary classification models; see, e.g., Bauer and Agarwal [2014] and Tian et al. [2015] for recent examples. A common problem is that default events are relatively rare and the number of defaulted companies is typically much lower than the number of non-defaults. In general, class imbalance [Japkowicz and Stephen, 2002], i.e., the situation where a majority class is much more frequent than a minority class, is a problem not just in the area of default prediction but also for various other classification tasks such as fraud detection or predictive maintenance. The problem of class imbalance is aggravated if the size of the data set is small.

In this article, we introduce a novel and flexible classification model, the Grabit model, which is obtained by applying gradient tree boosting to the Tobit model. The Grabit model allows for alleviating the class imbalance and/or small data problem and for obtaining increased predictive accuracy if auxiliary data, which is related to the underlying decision function, is available. For instance for default prediction, this auxiliary data can be in the form of stock returns, a distance-to-default measure, or credit spreads for companies with publicly traded stocks or bonds, and number of days of delay until repayment, number of outstanding payments, or amount in arrears for non-public companies.

In our application, we show that the Grabit model considerably outperforms other state-of-the-art approaches for default prediction of loans made to Swiss small and medium-sized enterprises (SMEs). In an extensive simulation study, we show how the increased predictive accuracy of the Grabit model compared to other classifiers depends on (i) the correlation between the auxiliary data and the latent decision function, (ii) the class imbalance ratio, (iii) the sample size, and (iv) the complexity of the decision function. In particular, our
results show that if the auxiliary data is independent of the decision function, the Grabit model still performs as well as the best competing model. This means that one loses no predictive accuracy when using the Grabit model in cases where the auxiliary data contains no additional information. Further, we observe that the Grabit model outperforms other models also in cases of moderately sized to large data sets if the decision function is sufficiently complex, for example, having strong non-linearities or interactions among predictors.

In addition, the Grabit model offers several important advantages over existing approaches. In contrast to the Logit and Tobit models, the Grabit model can learn non-linearities, discontinuities, and complex interactions. Since the Grabit model uses trees as base learners, it is robust against outliers in predictor variables and scale invariant to monotonic transformation for the predictor variables. This means that no transformation of the predictor variables is needed which is an important advantage in practice. Other advantages are that missing values in the predictors can be automatically accommodated and do not need to be imputed \cite{Elith2008}, and that the predictive performance is not impaired by the problem of multicollinearity.

The Grabit algorithm is implemented in Python and openly available on GitHub as a fork of scikit-learn\footnote{It is available in the branch ‘grabit’ on https://github.com/fabsig/scikit-learn.git. This can be installed, for instance, using the command ‘pip install git+https://github.com/fabsig/scikit-learn.git@grabit’. One then needs to choose the option ‘loss=tobit’ and specify the lower and upper limits ‘yl’ and ‘yu’ for the function ‘GradientBoostingRegressor’ accordingly. Alternatively, if reinstalling scikit-learn is not desirable, one can simply download the file ‘https://github.com/fabsig/scikit-learn/blob/grabit/sklearn/ensemble/gradient_boosting.py’ and then import it from a local directory using import ‘gradient_boosting as gb’.}

1.1 Short review of default prediction

In early proposals for statistical default prediction, the focus was primarily on linear classification models such as linear discriminant analysis (LDA) or logistic regression where the covariates enter the model in a linear combination. See \cite{Altman1968, Zmijewski1984, Lau1987, Shumway2001, Altman2007, Ding2012} for exam-
ples of this approach. Recently, non-linear methods such as generalized additive models, neural networks, classification trees, and ensemble methods have been proposed for default prediction. See, e.g., Brown and Mues [2012], Lessmann et al. [2015], Jones et al. [2017] for a comparison of these approaches for default prediction. In particular, it has also been proposed to use standard boosting algorithms for default prediction [Alfaro et al., 2008, Zieba et al., 2016, Xia et al., 2017]. In contrast to these previous studies, the focus of this article is not on comparing existing machine learning algorithms, but to introduce a novel boosting algorithm that can leverage auxiliary data to obtain increased predictive accuracy. Moffatt [2005] uses the Tobit model and variants of it for default prediction with a different goal than ours, though. Whereas we propose to leverage auxiliary data for the non-default cases to obtain a classifier with higher accuracy for default prediction, Moffatt [2005] has the goal to jointly model the default probability and the extent of default. A solution used by practitioners to overcome the limitations of binary classifiers for imbalanced default data is to model migrations across multiple discrete states. See, e.g., Vahid and Ahmadi [2016] for a recent example of an application with a machine learning model. In our application, we compare this approach to the Grabit model and observe that the Grabit model performs substantially and significantly better than a multi-state classifier.

The remainder of this article is structured as follows. In Section 2 we briefly review the Tobit model and gradient boosting and then introduce the Grabit model. In Section 3 we investigate the performance of the Grabit model in a simulation study. We then apply the model to the prediction of defaults of Swiss small and medium-sized enterprises (SMEs) in Section 4. We conclude in Section 5 and mention possible directions for future research.

2 Gradient Tree-Boosted Tobit Model

2.1 The Tobit model

As we show in the following, auxiliary data for the majority class can be combined with the binary data by the use of censored regression models. The Tobit model [Tobin, 1958] is one of the most widely used censored regression models. In the following, we briefly present the two-sided version of the Tobit model [Rosett and Nelson, 1975]. The one-sided
versions are simply obtained as special cases of the two-sided one by letting one of the boundaries converge to plus or minus infinity. We refer to Maddala [1986] or Amemiya [1985] for additional information on the Tobit model.

It is assumed that there exists a latent variable \( Y^* \) which follows, conditional on some covariates \( X = (X_1, \ldots, X_p)^T \in \mathbb{R}^p \), a Gaussian distribution:

\[
Y^*|X \sim N(\mu(X), \sigma^2).
\]  

(1)

The mean \( \mu(X) \) is assumed to depend linearly on the covariates \( X \) through

\[
\mu(X) = X^T \beta,
\]  

(2)

where \( \beta \in \mathbb{R}^p \) denotes a set of coefficients. This latent variable \( Y^* \) is observed only if it lies in an interval \([y_l, y_u]\). Otherwise, one observes \( y_l \) or \( y_u \) depending on whether the latent variable is below the lower threshold \( y_l \) or above the upper threshold \( y_u \), respectively. Denoting \( Y \) as the observed variable, we can express this as

\[
Y = \begin{cases} 
  y_l, & \text{if } Y^* \leq y_l, \\
  Y^*, & \text{if } y_l < Y^* < y_u, \\
  y_u, & \text{if } y_u \leq Y^*. 
\end{cases}
\]  

(3)

It it easily seen that the distribution of \( Y \) is a mixture of a discrete and a continuous distribution. We denote by \( \delta_a(y), a \in \mathbb{R} \), the Dirac measure which is the probability measure of a discrete random variable which equals \( a \) with probability one. The density of \( Y \) with respect to the sum of the Lebesgue measure and the Dirac measures \( \delta_{y_l}(y) \) and \( \delta_{y_u}(y) \) is then given by

\[
f_{\mu(x), \sigma}(y) = \Phi \left( \frac{y_l - \mu(x)}{\sigma} \right) 1_{y_l}(y) + \frac{1}{\sigma} \phi \left( \frac{y - \mu(x)}{\sigma} \right) 1_{\{y_l < y < y_u\}} \\
+ \left( 1 - \Phi \left( \frac{y_u - \mu(x)}{\sigma} \right) \right) 1_{y_u}(y),
\]  

(4)

where \( 1_A(y), A \subset \mathbb{R} \), denotes the indicator function which equals one if \( y \in A \) and zero otherwise, \( \phi(x) \) and \( \Phi(x) \) are the standard normal density and cumulative distribution.

Although the Tobit model is defined by the censoring mechanism in Equation (3), it can be applied not only to truly censored data but to many other cases where the data
consists of a continuous part and a discrete point mass at the borders. This includes for instance fractional response data, corner solution response data, rainfall data [Papke and Wooldridge, 1996, Sanso and Guenni, 1999, Wooldridge, 2005, Papke and Wooldridge, 2008, Sigrist et al., 2012], or default data as in this article. The latent variable $Y^*$ can often be interpreted as a potential which indicates how likely the event under consideration occurs. For instance, for rainfall or default prediction, $Y^*$ can be interpreted as a rainfall or default potential. Rainfall or default, respectively, occurs if the potential $Y^*$ exceeds a certain threshold. In our application, default events correspond to the case $Y^* \geq y_u$, i.e., for defaults, the observed data equals $Y = y_u$. The non-default cases correspond to $Y^* < y_u$, and the auxiliary data for this majority class is then identified with the observed, and potentially lower censored, variable $Y$. As we show in the simulation study in Section 3, gains in predictive accuracy can be obtained with the Grabit model also in cases where this auxiliary data and the binary default data is not generated by the same model as in (1) and (3).

2.2 Gradient boosting

A rather restrictive assumption of the Tobit model is the functional form in Equation (2) which relates a set of covariates to a linear predictor. In this article, we relax this assumption by applying gradient tree boosting to the Tobit model. We denote the resulting model as 'Grabit' model. Boosting enjoys large popularity in many areas mainly due to its excellent predictive accuracy on a wide range of data sets; see, e.g., [Chen and Guestrin, 2016] or [Yang et al., 2017]. It is an ensemble technique which additively combines multiple relatively simple models, so-called base learners which often consist of regression trees. Boosting was first introduced in machine learning for classification [Freund and Schapire, 1995]. Important contributions to the topic, in particular the statistical view of boosting as stagewise optimization of a risk functional include [Breiman, 1998, Friedman et al., 2000, Friedman, 2001]. See [Bühlmann and Hothorn, 2007, Mayr et al., 2014a, and Mayr et al, 2014b] for recent reviews on boosting algorithms.

In the following, we briefly present the idea of gradient boosting as introduced by [Friedman, 2001]. We assume that there is a response variable $Y$ and a vector of covariates...
$X \in \mathbb{R}^p$, and that we observe data $(y_i, x_i), i = 1, \ldots, n$. The goal of boosting is to find a minimizer $F^*(\cdot)$ of the empirical risk $R^e(F)$

$$F^*(\cdot) = \arg\min_{F(\cdot) \in \Omega_S} R^e(F)$$

$$= \arg\min_{F(\cdot) \in \Omega_S} \sum_{i=1}^{n} L(y_i, F(x_i)),$$

where $F(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}$ are functions that map $X$ to $Y$ and where $L$ is an appropriately chosen loss function such as, e.g., the squared error $L(y, F) = (y - F)^2$ or the negative Tobit log-likelihood in our case. One restricts the functions $F(\cdot)$ to lie in the span $\Omega_S = \text{span}(S)$ of a set $S$ of so-called base learners $h(x; a[m])$. I.e.,

$$F(x) = F[0] + \sum_{m=1}^{M} \rho[m] h(x; a[m]),$$

where we assume that $h(x; a[m])$ are regression trees [Breiman et al., 1984] with parameters $a[m]$, $F[0]$ is a constant, and $M$ denotes the number of boosting iterations or trees.

The gradient boosting approach of Friedman [2001] iteratively finds $F^*(\cdot)$ using a functional gradient descent algorithm. Denoting the current estimate for $F^*(\cdot)$ by $F^{[m-1]}(\cdot)$, an update from $F^{[m-1]}(\cdot)$ to $F^{[m]}(\cdot)$ is obtained by first calculating the negative gradient

$$-\frac{\partial L(y_i, F(x_i))}{\partial F} \bigg|_{F=F^{[m-1]}},$$

and then approximating this gradient with a base learner $h(x; a[m])$. Additionally, one can then perform a line search over $R^e(F^{[m-1]}(x) + \rho[m] h(x; a[m]))$ to find $\rho[m]$. If $h(x; a[m])$ are regression trees [Breiman et al., 1984], this line search is usually done separately for the coefficients $\gamma_j^{[m]}$ of each partition; see Section 2.2 for more details.

In addition, a shrinkage factor $\nu$, $0 < \nu \leq 1$ is typically used for the update step:

$$F^{[m]}(x) = F^{[m-1]}(x) + \nu \rho^{[m]} h(x; a[m]).$$

This parameter $\nu$ acts as a regularization parameter. It has been empirically observed that the introduction of a shrinkage factor slows down overfitting and results into increased predictive performance.

### 2.3 The Grabbit model

In this section, we introduce the Grabbit model which is obtained by extending the Tobit model using gradient boosting with trees as base learners. While non-linearities and
interactions can be modeled in various ways, boosting with trees provides a very flexible approach which relies on few assumptions and which in particular shows very good predictive performance on a wide range of data sets [Chen and Guestrin, 2016].

Instead of assuming a linear functional form as in Equation (2), the Grabbit model uses a flexible function \( F(\cdot) : \mathbb{R}^p \to \mathbb{R} \) which is assumed to be an ensemble of regression trees obtained through boosting. I.e., for the mean \( \mu(X) \) of the latent variable \( Y^* \) in the Tobit model, we assume

\[
\mu(X) = F(X).
\]

An estimate for this function is found by applying boosting with regression trees as base learners. To be specific, we use the negative log-likelihood of the Tobit model as loss function \( L(y, F) \):

\[
L(y, F) = -\log \left( f_{F,\sigma}(y) \right),
\]

where the density \( f_{F,\sigma}(y) \) of the Tobit model is given in Equation (4). The explicit expression for the corresponding loss \( L(y_i, F(x_i)) \) for one observation \((y_i, x_i)\) can be found in the appendix in Equation (10). For the boosting algorithm, we consider \( \sigma \) as a known parameter. The parameter \( \sigma \) can be chosen by cross-validation or estimated using a profile-likelihood approach; see Section 2.4.

The gradient boosting approach of Friedman [2001] then works by iteratively fitting a regression tree \( h(x_i, a^{[m]}) \) to the so-called pseudoresponses \( \tilde{y}_i \) which are given by the negative gradient

\[
\tilde{y}_i = -\frac{\partial L(y_i, F)}{\partial F} \bigg|_{F=F^{[m-1]}(x_i)}.
\]

The regression tree \( h(x_i, a^{[m]}) \) is fitted to the pseudoresponse data \((\tilde{y}_i, x_i), i = 1, \ldots n, \) by assuming a squared error.

Next, the optimal \( \rho^{[m]} \) is found by minimizing the empirical risk

\[
\rho^{[m]} = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, F^{[m-1]}(x_i) + \rho h(x_i, a^{[m]})).
\]

Since trees are used as base learners, instead of finding one global step size \( \rho^{[m]} \), we can find the optimal step size for each partition separately by recalculating the values of the terminal nodes of the tree \( h(x_i, a^{[m]}) \). Denoting by \( \{R_{jm}\}_{j=1}^{J} \) the partition of the tree \( h(x_i, a^{[m]}) \) with
J terminal nodes, the optimal coefficients \( \gamma_j^{[m]} \) of these partitions can be found by doing a line search

\[
\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F^{[m-1]}(x_i) + \gamma).
\]

However, for the Tobit loss function, this line search cannot be done in closed form and, consequently, an approximate minimization method has to be used. Following Friedman [2001], we use a second order Taylor approximation for \( \sum_{x_i \in R_{jm}} L(y_i, F^{[m-1]}(x_i) + \gamma) \) around \( F^{[m-1]}(x) \), and find \( \gamma_j^{[m]} \) such that the approximation is minimized. This corresponds to performing a single Newton-Raphson step as follows:

\[
\gamma_{jm} = -\sum_{x_i \in R_{jm}} \frac{\partial L(y_i, F)}{\partial F} \bigg|_{F = F^{[m-1]}(x_i)} \sum_{x_i \in R_{jm}} \frac{\partial^2 L(y_i, F)}{\partial^2 F} \bigg|_{F = F^{[m-1]}(x_i)}.
\]

For the boosting algorithm described above, we need to be able to evaluate both the gradient \( \frac{\partial L(y_i, F)}{\partial F} \) and the second derivative \( \frac{\partial^2 L(y_i, F)}{\partial^2 F} \). These can be calculated explicitly; see Equations (11) and (12) in the appendix. In summary, we use a gradient descent step to find the structure of the trees, i.e., the partition of the space, and a Newton update step to learn the leaves. We then obtain the following algorithm.

**Algorithm 1: Grabit (gradient tree-boosted Tobit model)**

1. Initialize \( F^{[0]}(x) = \bar{y} \).
2. for \( i = 1 \) to \( M \) do
3. Compute the pseudoresponses using (11): \( \tilde{y}_i = -\frac{\partial L(y_i, F)}{\partial F} \big|_{F = F^{[m-1]}(x_i)} \).
4. Fit a J-terminal node regression tree to \((\tilde{y}_i, x_i), i = 1 \ldots n\), and obtain a partition \( R_{jm}^j, j = 1 \ldots J \).
5. Update the terminal nodes of the tree using (12):
6. Update \( F^{[m]}(x) = F^{[m-1]}(x) + \nu \sum_{j=1}^{J} \gamma_j^{[m]} 1_{R_j^{[m]}(x)}, j = 1 \ldots J \).
7. end
8. Return \( F^{[M]}(x) \).

The Grabit algorithm is implemented in Python and it is openly available on GitHub as a fork of scikit-learn; see Section 1 for more information.
2.4 Choice of tuning parameters

The Grabit algorithm has several tuning parameters. Concerning the boosting updates and the trees as base learners, we consider the following tuning parameters: the number of trees $M$, the shrinkage factor $\nu$, and the size of the trees $L$. In addition, the Tobit model has the parameter $\sigma$ which is the standard deviation of the latent variable $Y^*$ in Equation (1). In the following, we discuss how these parameters can be chosen.

The shrinkage factor $\nu$ and the number of trees $M$ control the amount of regularization. Past research [Friedman, 2001; Bühlmann and Hothorn, 2007] has shown that the predictive accuracy of boosting algorithms is generally superior when choosing smaller values for $\nu$. The size of the trees $L$ controls the amount of interaction among the covariates $X$. The parameters $\nu$, $M$, and $L$ are can be chosen by cross-validation or using an information criterion. Cross-validation refers to the procedure of splitting the data into a training and a validation data set. The model is then estimated using various choices of tuning parameters and evaluated on the validation data. If the computational costs are not too large, one can repeat this several times. One then chooses the combinations of tuning parameters that shows the best predictive performance on the left out-of-sample validation data. Note that if subsequently model comparison is performed, an additional test data is usually left aside in order to compare different models.

The parameter $\sigma$ can be chosen by either maximizing the profile likelihood or also by cross-validation. The profile log-likelihood function for $\sigma$ is given by the negative empirical risk

$$\ell(\sigma) = -R(\hat{F}_\sigma, \sigma)$$

as a function of $\sigma$, where $\hat{F}_\sigma(\cdot)$ is obtained as outline above in Section 2.3 for a fixed $\sigma$. The maximum

$$\hat{\sigma} = \arg\min_{\sigma} \ell(\sigma)$$

can be found by using a general purpose optimizer in the form of, e.g., a quasi-Newton method such as the Broyden-Fletcher-Goldfarb-Shanno algorithm. In order to avoid problems with negative values, one can reparametrize $\sigma$ by $\phi = \log(\sigma) \in \mathbb{R}$, find $\hat{\phi} = \arg\min_{\phi} \ell(e^{\phi})$, and set $\hat{\sigma} = e^{\hat{\phi}}$. A computationally faster but potentially less accurate alternative is to do a grid search over a grid $\{\phi_1, \ldots, \phi_K\}$, where $K$ is the number of grid points.
2.5 Model interpretation

Linear models such as logistic regression have the advantage that they can be easily interpreted. The Grabit model and, in general, gradient boosted trees are not as easily interpretable as linear models due to their generality and flexibility. However, compared to other non-linear models such as neural networks or support vector machines, gradient boosted trees can be relatively well interpreted. The two main tools are variable importance measures and partial dependence plots.

2.5.1 Variable importance

In many applications, one has a large set of predictor variables and not all of them are equally important for the prediction of $Y$. Often, only relatively few variables have a substantial influence on the predictor variable $Y$ [Friedman et al., 2001]. It is thus often of interest to know which variables are the most relevant ones.

For a single $J$-terminal node tree $T^{[m]}$, Breiman et al. [1984] proposed the following measure of importance for variable $X_I$

$$
\mathcal{I}_{X_I} (T^{[m]}) = \sum_{j=1}^{J-1} \hat{\tau}_j^2 \mathbb{1}_{v_j}(X_j),
$$

where the sum is over all nonterminal nodes $j$, $v_j$ is the splitting variable selected in node $j$, $\mathbb{1}_{v_j}(X_j)$ is an indicator function that equals one if $X_j$ is the splitting variable in node $j$ and zero otherwise, and $\hat{\tau}_j^2$ denotes the reduction in squared error due to split $j$.

Friedman [2001] generalized this measure to an ensemble of $M$ trees obtained by boosting by simply taking the average of the measures of all single trees

$$
\mathcal{I}_{X_I} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_{X_I} (T^{[m]}).
$$

Note that despite their popularity $\mathcal{I}_{X_I} (T^{[m]})$ and $\mathcal{I}_{X_I}$ can be biased [Breiman et al., 1984] in the sense that a variable $X_I$ that is independent of $Y$ might still be selected for a split in a tree, and hence the variable importance measure $\mathcal{I}_{X_I}$ might not be zero. A bias correction can be obtained along the ideas presented in Sandri and Zuccolotto [2008].
2.5.2 Partial dependence plots

Visualization is a powerful tool for interpreting models. For gradient boosted trees, one can visualize main effects and second-order interactions using partial dependence plots [Friedman, 2001]. For this, we partition the predictor variables $X$ in two non-overlapping subsets $X_s$ and $X_{\setminus s}$, where $X_{\setminus s}$ is the complement of $X_s$. For main effects and second-order interactions, $X_s$ will simply consist of one or two variables.

Given a model $\hat{F}(\cdot)$ and data $(y_i, x_i)$, an estimate for the average partial dependence of $\hat{F}(\cdot)$ on $X_s$ can be obtained as

$$\hat{F}_s(X_s) = \sum_{i=1}^{n} \hat{F}(X_s, x_{\setminus s,i}).$$

A partial dependence plot is then obtained by plotting $\hat{F}_s(X_s)$ versus $X_s$.

Note that this approach is limited to the visualization of low order interactions. Depending on the highest interaction order, which is controlled by the tree depth parameter $L$, higher order interactions can be important. In such cases, main effects and second-order interactions give only an incomplete picture of the full model $\hat{F}(\cdot)$.

2.6 Explaining predictions

Variable importance is a summary measure for the overall impact of a variable $X_l$ on the target variable $Y$. In practice, it is often desirable to know which features have the largest influence for a specific prediction

$$\hat{y} = \hat{F}(x_{\text{new}}),$$

where $x_{\text{new}}$ is a new set of predictor variables for which predictions should be made. For instance, in a credit scoring model, a user would like to know which features are the main drivers for a certain score. For this, we propose the following approach.

For each variable $X_l$, one calculates the predicted value two times: once for the data $x_{\text{new}}$ and once for $(x_{l,\text{med}}, x_{\setminus l,\text{new}})$ where one replaces the value for variable $X_l$ with the median $x_{l,\text{med}} = \text{median}\{x_{il}, i = 1, \ldots, n\}$ of the past data. Instead of the median, one can also use another appropriate summary measure or simply shift the value locally around
\[ I_{X_l}(x_{\text{new}}) = \left| \hat{F}(x_{\text{new}}) - \hat{F}(x_{l,\text{med}}, x_{\setminus l,\text{new}}) \right|, \]

where \( x_{\setminus l,\text{new}} \) contains data for all variables from \( x_{\text{new}} \) except for \( X_j \). We take \( I_{X_l}(x_{\text{new}}) \) as a measure of importance for variable \( l \) for the specific data \( x_{\text{new}} \). In general, this approach can be used for any non-linear model for which such a justification is desirable, and it can be easily extended to quantify the joint influence of more than one variable by simply replacing the values of a group of variables with their medians.

3 Simulation study: imbalanced classification with auxiliary data

In the following, we compare the Grabit model with other state-of-the-art approaches in a simulation study. We consider the task of binary classification in a class imbalance setting where the minority class has relatively few observations but auxiliary data, which is correlated with the decision function, is available for the majority class. The setting is inspired by the default prediction application in Section 4, where default cases are rare but one has auxiliary information for the non-default cases such as the number of days in delay, stock price returns, or distance to default measures.

We first investigate the impact of the correlation between the latent decision function and the auxiliary variable for the majority class on the performance of the Grabit model. In Sections 3.3 and 3.2, we then investigate how the results change when varying the class imbalance ratio and the sample size. In addition, we also consider other types of decision functions and a larger data set in Section 3.4.

3.1 The impact of correlation between the auxiliary data and the latent decision function

In the following, we investigate how the correlation between the auxiliary variable and the latent decision function is related to the gain in predictive accuracy of the Grabit model with respect to other models. We consider four different levels of (Pearson) correlations:
0.75, 0.5, 0.25, and 0 between the latent decision function and the auxiliary data. We use a simulation setting that mimics the situation of our application in Section 4. In particular, the class imbalance ratio, the sample size \(n\), and the number of predictor variables \(p\) are chosen to approximately correspond to the characteristics of the data in our application in Section 4.

We assume the following model for simulating data:

\[
Y^* = F(X) + \epsilon, \quad X = (X_1, \ldots, X_p)^T \in \mathbb{R}^p, \quad \epsilon \sim N(0,1),
\]

\[
C = \mathbb{1}_{\{y^* \geq y_u\}}(Y^*),
\]

\[
Y_a = C \cdot y_u + (1 - C) \cdot (F(X) + \epsilon_a), \quad \epsilon_a \sim N(\mu_a, \sigma_a^2).
\]

(6)

Here, \(C\) denotes the observed binary variable. The variable \(C\) equals one if the latent variable \(Y^*\) is above a threshold \(y_u\) and zero otherwise. For the mean function \(F(\cdot)\) of the latent variable \(Y^*\), we use a non-linear function which simply consists of \(p = 50\) quadratic terms:

\[
F(X) = 1.5 \sum_{k=1}^{p} X_k^2, \quad X_k \overset{iid}{\sim} \text{Unif}(-1,1).
\]

(7)

The threshold \(y_u\) is chosen such that we obtain a class imbalance ratio of approximately 95% to 5%, i.e., \(P(C = 1) \approx 5\%\). For the above specified mean function, this corresponds to \(y_u = 2.98\).

We assume that besides the binary variable \(C\), we additionally observe the auxiliary variable \(Y_a\). The variable \(Y_a\) equals \(F\) plus a Gaussian error term \(\epsilon_a\) with standard deviation \(\sigma_a\) for the majority class \(C = 0\). This means that for the majority class, we have auxiliary data, which is correlated with the latent decision function \(F\). The standard deviation \(\sigma_a\) of the noise term \(\epsilon_a\) determines the correlation between \(Y_a\) and \(F\). We choose different values for \(\sigma_a\) in order to compare the performance of the Grabit model under different correlation levels. Specifically, we use \(\sigma_a = 0.53, 1.05,\) and \(2.35\) which corresponds to Pearson correlations between \(Y_a\) and \(F\) of approximately \(0.75, 0.5,\) and \(0.25\). The means \(\mu_a\) of \(\epsilon_a\) are simply set to a value such that the auxiliary data is below the threshold \(y_u = 2.98\). We use \(\mu_a = -1.3, -3.3,\) and \(-9\) for \(\sigma_a = 0.53, 1.05,\) and \(2.35\). In addition, we also consider the case of zero correlation between the auxiliary variable and the decision function by simulating according to \(Y_a = C \cdot y_u + (1 - C) \cdot \epsilon_a, \quad \epsilon_a \sim N(-4,1)\).
We compare the one-sided Grabit model with two other classification methods which only use data for the binary variable $C$ and with the Tobit model which also uses the auxiliary data. As classification methods, we use a logistic regression model denoted as 'Logit' model as well as a tree boosted classifier for a Bernoulli likelihood with a logistic link function denoted as 'boosted Logit' model\footnote{For simplicity and brevity, we restrict ourselves to these two classifiers. However, other state-of-the-art classifiers such as random forest do not perform better than the boosted Logit model in our simulations (results non reported).}. The Logit model and the boosted Logit model are fitted using the open-source scikit-learn library \cite{scikit-learn}. For estimating the Grabit model, we use the algorithm presented in Section \ref{sec:alg}. For the Tobit model, we numerically minimize the negative log-likelihood in Equation (10) using a quasi-Newton method.

In each simulation iteration, we simulate $n = 500$ data points for use as training data for estimating the models and additional 500 data points as test data for comparing the different models. In total, the simulations are repeated 100 times. Tuning parameters are selected on an independent validation data set of the same size for all models. We use the area under the receiver operating characteristic (AUROC) as a measure of the predictive accuracy for choosing tuning parameters. For the boosted Logit model, we have the following tuning parameters: the number of trees $M$, the learning rate $\nu$, and the size of the trees $L$. These are chosen among the following combinations of tuning parameters $M \in \{10, 100, 1000\}$, $\nu \in \{0.1, 0.01, 0.001\}$, and $L \in \{3, 5, 10\}$. For the Grabit model, we additionally select $\sigma$ from $\{0.01, 0.1, 1, 10, 100\}$.

In Figure \ref{fig:results} we show the results for the four different correlation levels 0.75, 0.5, 0.25, and 0. We use the receiver operating characteristic (ROC) and the area under ROC (AUROC) to compare the different approaches. The AUROC values shown in the plots are sample means of the 100 simulations runs. In addition, the 95\% confidence intervals for the AUROCs are obtained by calculating 2.5\% and 97.5\% quantiles. ROC curves and 95\% confidence bands (shaded areas) are point-wise means as well as point-wise 2.5\% and 97.5\% quantiles of the 100 simulated ROC curves. Before calculating means and quantile, the sample ROC curves are first linearly interpolated over an equally spaced grid with 100 grid points between 0 and 1.
Figure 1: Comparison of models using receiver operating characteristic (ROC) and area under ROC (AUROC) for different levels of correlation between the auxiliary data and the latent decision function when simulating from the model in (6) and (7).

The plots show that if the correlation between the auxiliary variable and the latent decision function is larger than zero, the Grabit model clearly outperforms the other approaches. I.e., the Grabit model can use the information in the auxiliary data in order to increase the predictive performance. Due to the non-linear nature of the function $F$ in (7), the performances of both the Logit and the Tobit model are essentially equivalent to
random guessing. In the case where the correlation is zero, the predictive accuracy of the Grabit model is essentially equivalent to the one of the best performing competitor, i.e., the boosted Logit model. This means that in the case where the auxiliary data contains no information, we lose no predictive accuracy when using the Grabit model. Intuitively, the reason for this is that in the case of unrelated auxiliary data, which is independent of the decision function, the estimated decision function of the Grabit model has no systematic structure in non-default areas and, consequently, we obtain very similar binary predictions for varying thresholds as in the boosted Logit model.

In the following, we investigate how these results depend on the sample size, on the degree of class imbalance, and on the functional form of the latent decision function $F$.

### 3.2 The impact of the sample size

Here, we investigate how the sample size impacts the performance of the Grabit model and the other classifiers. We use the same simulation setting as in Section 3.1 with a correlation of 0.5 between the auxiliary variable and the latent decision function. In addition to the sample size of $n = 500$ from above, we consider the cases $n = 200, 1000, 2000, \text{ and } 10000$. The results are shown in Figure 2. The figure shows that the smaller the sample size, the larger is the performance gain of the Grabit model compared to the other models. For the largest sample, the performance of the boosted Logit model is almost as good as the one of the Grabit model. Again, due to the non-linearity, both logistic regression and the Tobit model perform poorly. As expected, the smaller the sample size, the wider are the confidence bands for all methods.

### 3.3 The impact of the class imbalance ratio

In this section, we explore the relation between the class imbalance ratio and the performance of the Grabit model and the other approaches. Previously, we have assumed that the minority class occurs in approximately 5% of all cases. Here, we additionally consider the following proportions of the minority class: 1%, 2%, 10%, and 20%. This corresponds to using the following thresholds $y_u$: 3.82, 3.47, 2.53, and 2. Apart from this, we use the same simulation setting as in Section 3.1 with a correlation of 0.5 between the auxiliary
Figure 2: Results from a simulation study for investigating the impact of sample size on the performance of the Grabit model and other approaches.

variable and the latent decision function.

The results are shown in Figure 3. We observe that the increase in predictive accuracy of the Grabit model compared to the boosted Logit model is larger the larger the class imbalance ratio or the lower the proportion of the minority class. Not surprisingly, the lower the fraction of the minority class, the wider are the confidence bands due to the small number of the minority class observations in the test data. Further, we also observe
that the performance of the Grabit model is even slightly better, the smaller the fraction of the minority class. This is because it can be advantageous to directly observe a noisy version of the latent decision function compared to observing only a random binary variable with mean given by the decision function.

Figure 3: Results from a simulation study for investigating the impact of the class imbalance ratio on the performance of the Grabit model and other approaches.
### 3.4 Other decision functions and sample size

In this section, we consider two other choices for the latent decision function $F$: a linear function and a highly non-linear function in combination with a larger sample size. For the linear function, we replace (7) with

$$F(X) = 0.25 \sum_{k=1}^{50} X_k, \quad X_k \sim \text{Unif}(-1, 1).$$  \hfill (8)

As non-linear function, we use the following choice

$$F(X) = 2 \cos \left(4\pi \sqrt{\sum_{k=1}^{20} X_k^2} \right), \quad X_k \sim \text{Unif}(-1, 1).$$  \hfill (9)

In both cases, $y_u$ is chosen such that approximately 5% of all cases are in the minority class and $\sigma_a$ is chosen such that the correlation between the auxiliary variable and the latent decision function $F$ is approximately 0.5, and $\mu_a$ such that all simulated auxiliary data is below $y_u$. For the linear model, we simulate $n = 500$ data points and for the non-linear function, we simulate $n = 10000$ data points in each simulation iteration.

The results from this are shown in Figure 4. The figure shows that in the linear case, the Tobit model performs best and the tree based Grabit model performs worse than the Tobit model. Similarly, the tree-boosted Logit model performs worse than the linear Logit model. It is not surprising that linear models perform best in situations where the decision function is linear.

For the highly non-linear function in (9) with $n = 10000$ simulated data points, we observe that the Grabit model still outperforms the other three approaches by a large margin. This example shows that the Grabit model can also provide increased predictive accuracy for moderately to large sized data sets when the decision function is sufficiently complex, e.g., having strong non-linearities, interactions, or many predictor variables.

In summary, we have found the following results in this simulation study. First, the larger the correlation between the auxiliary data and the latent decision function, the larger is the performance gain of the Grabit model compared to other classification methods, which only use binary data and neglect the auxiliary data. In the case of zero correlation, the Grabit model can perform as good as it second best competitor, the boosted Logit
model. Further, the larger the class imbalance or the smaller the sample size, the larger is the performance gain of the Grabit model. Not surprisingly, in linear situations, the linear Tobit model outperforms the Grabit model. Finally, the Grabit model outperforms other models also in cases of moderately to large-sized data sets if the decision function is complex such as, e.g., showing strong non-linearities or interactions among predictors. In general, it is likely that the more complex the decision function, e.g., the stronger the non-linearities or the higher the number of predictor variables, the larger is the performance gain of the Grabit model compared to other approaches that ignore the auxiliary data.

4 Application to SME default prediction

In this section, we apply the Grabit model to default prediction of loans made to small and medium-sized enterprises (SME) in Switzerland. The data is provided by Advanon, a Swiss start-up company, which operates a platform on which SMEs can obtain short-term loans by pre-financing invoices. The goal of this application is to predict whether a loan will be repaid or not once an SME requests a new loan.
On Advanon’s platform, a loan can be repaid with some delay without immediately being classified as a default. This means that in addition to the information whether a company defaulted on a loan or not, we know for each loan whether it was repaid in due time or, if not, the number of days of delay by which it was repaid. The maximum number of days in arrear is 60. In case a loan is overdue more than 60 days, the loan is automatically classified as a default event by Advanon. A loan can obviously also result in a default prior to this 60 day grace period.

We use the Grabbit model for jointly modeling the auxiliary delay days and the binary default events. In the Grabbit model, our observed variable $Y$ is a censored version of a latent variable $Y^*$ that can be interpreted as a default potential or a credit score. Both lower and upper censoring occur at $y_l = 0$ and $y_u = 60$, respectively. If the latent variable exceeds the upper threshold, $Y^* \geq 60$, a default occurs. I.e., all default events correspond to $Y = 60$. If the latent variable is below 60, this corresponds to the number of days of delay. Lower censoring at $y_l = 0$ is introduced in order to account for the large number of loans that did repay without any delay at all.

4.1 Data

The data consist of 850 loans made to 141 different Swiss SMEs between 2016 and 2017. In total, 36 loans were not repaid due to defaults of 14 different SMEs. Note that a company can request more than one loan at the same time, and the lifespan of different loans for the same company can be overlapping. In Figure 5 we illustrate the default events and the number of delay days by which loans were repaid. The point mass at 60 represents the fraction of default events and the remaining part of the histogram below 60 represents delay days.

For each loan, we have data for approximately 50 different predictor variables. These covariates include financial ratios calculated from balance sheets and income statements, SME characteristics such as variables that reflect the repayment history of the SME on the platform or the age of the company, loan characteristics such as the loan amount or time until maturity, ratings for the SME from social media platforms, data from several external rating agencies, and data about online user behavior such as log-in and click data.
We transform variables that are highly skewed such as loan amounts or several balance sheet summaries by applying the logarithm. We do this in order to mitigate the influence of single data points when applying a logistic regression model for comparison below. For tree based models such as the Gribit model, this is not needed since they are invariant under monotone transformation.

For confidentiality reasons, we cannot fully disclose all predictor variables, and the data used here consists of a random subsample of all loans made on Advanon’s platform to Swiss SMEs. The subsample contains all default events but only a random selection of all non-defaulted loans. This is done in order to not disclose the actual default rate on Advanon’s platform, which is different from the one in the random subsample used here. However, the results shown in the following change only marginally when using the original data set.

Figure 5: Histogram of delay days and default events. The bar at 60 does not correspond to delay days but represents the fraction of default events. The remaining part of the histogram below 60 represents the number of delay days.
We compare the performance of the Grabit model with several alternatives: logistic regression (“Logit”), a classification tree, a random forest, tree-boosted logistic regression (“boosted Logit”), a neural network, the Tobit model, and a tree-boosted multinomial logistic regression model (“boosted multiclass Logit”). Apart from the Tobit model and the boosted multiclass Logit model, the methods are classification techniques that only use the binary information whether a loan was repaid or not. Similarly as the Grabit model, the Tobit model also uses the number of delay days as auxiliary data but assumes a linear decision function. The boosted multiclass Logit model uses the auxiliary data as follows. We created four states: ”no delay”, delay between 1 and 30 days, delay between 31 and 60 days, and default. We then use a boosted multinomial softmax classifier as described in [Friedman 2001] to model these four discrete states.

For evaluating and comparing the different models, we use temporal cross-validation. For each loan, we make a prediction and compare the prediction with the actual outcome. In doing so, the models are estimated based on data of past loans only, i.e., loans that are either repaid or have been classified as default events at the time when making the prediction. To avoid a temporal censoring issue, we only use data for loans for which the repayment date is at least 61 days due at the time of estimation. In addition, we require at least 100 past data points in order to train a model and to make a prediction. In total, there are 610 loans, out of which 28 are default cases, for which we can make this temporal out-of-sample evaluation. For a relatively small fraction of the data, there are missing values in some of the predictor variables. These are simply interpolated by using the median of all past data available at that point in time.

As in the simulation study, all models are fitted in Python. Except for the Grabit model and the Tobit model, the open-source scikit-learn library [Pedregosa et al. 2011] is used for fitting the models. For estimating the Grabit model, we use the algorithm presented in Section 2.3 and implemented in Python. For the Tobit model, we numerically minimize the negative log-likelihood in Equation (10) using a quasi-Newton method. Tuning parameters for the boosting approaches (boosted Logit, multiclass boosted Logit, Grabit) and the random forest are chosen using the same temporal cross-validation scheme with the area under
the receiver operating characteristic (AUROC) as a measure of fit. As in the simulation study, we have the following tuning parameters for the tree-boosted methods: the number of trees $M$, the learning rate $\nu$, and the size of the trees $L$. These are chosen among the following combinations of tuning parameters $M \in \{10, 100, 1000\}$, $\nu \in \{0.1, 0.01, 0.001\}$, and $L \in \{3, 5, 10\}$. For the Grabit model, we additionally select $\sigma$ from $\{0.01, 0.1, 1, 10, 100\}$. For the random forest algorithm, we consider the following tuning parameters: the number of trees $M \in \{10, 100, 1000\}$, the fraction of the number of variables considered for making a split when growing trees $\rho \in \{50\%, 75\%, 100\\%\}$, and the tree size $L \in \{3, 5, 10\}^3$. For the neural network, we use two hidden layers with five and two nodes, respectively, and rectified linear units (ReLU) as activation functions.$^4$

We evaluate the predictions and compare the different models using the receiver operating characteristic (ROC) curve and the area under ROC (AUROC). The ROC curve is obtained by plotting the true positive rate versus the false positive rate for varying thresholds. The AUROC is the area under this curve. Figure 6 shows our results. As the figure shows, the Grabit model clearly outperforms all other approaches considered. For most of the thresholds, the ROC curve of the Grabit model is above the ROC curves of all other models. Further, the AUROC is considerably larger compared to all other alternative models considered. In particular, the AUROC of the Grabit model is significantly higher at the 5% level compared to all other models when using the DeLong test [DeLong et al., 1988]. The corresponding p-values are reported in Table 1. The figure also shows that the Tobit model performs poorly. Similarly as in the simulation study in Section 3, this is likely due to the presence of non-linearity and interactions in the auxiliary data. Further, the poor performance of the classification tree is due to the fact that the unpruned tree overfits the data.

$^3$We obtain the following choices for the tuning parameters. Boosted Logit: $\nu = 0.1$, $M = 1000$, $L = 3$. Boosted multiclass Logit: $\nu = 0.01$, $M = 1000$, $L = 5$. Grabit: $\nu = 0.1$, $M = 100$, $L = 3$, $\sigma = 1$. Random forest: $M = 1000$, $L = 10$, $\rho = 1$.

$^4$No attempt is made to search for an optimal network structure of the neural network. As we mention below in the text, the sample size is too small to have an additional validation set, which we could use for choosing the potentially many tuning parameters of a neural network (e.g. number of hidden layers, number of nodes per layer, different activation functions, several regularization options) without resulting in in-sample overfitting.
Due to the small sample size, we use the same data for choosing the tuning parameters and for comparing the different models, i.e., we do not distinguish between a validation and a test data. This comes at the potential risk that the results of methods for which tuning parameters are chosen in this way are too optimistic. In order to investigate whether this is an issue, we also consider other non-optimal choices of tuning parameters. In particular, in Figure 9 in the appendix, we report the results when using the second best choice of
Table 1: Comparison of AUROC of the Grabit model with alternative approaches using the DeLong test.

| Model                  | p-Value  |
|------------------------|----------|
| Boosted Logit          | 0.0371   |
| Boosted multiclass Logit| 0.0379   |
| Classification tree    | 1.3e-14  |
| Logit                  | 0.002    |
| Neural network         | 2.83e-06 |
| Random forest          | 0.00128  |
| Tobit                  | 3.27e-07 |

The performance of the Grabit model is essentially the same whereas some of the other methods perform marginally worse. We obtain similar results when using other non-optimal, but reasonable, tuning parameters (results not reported). These results show that our findings are robust to different choices of tuning parameters. In addition, concerning the use of parameters optimally tuned on the test data for boosting methods, Bühlmann and Yu [2003] state that ”the effect of using the optimal number of boosting iterations instead of an estimated [e.g. by cross-validation] number is typically small”.

An alternative approach for dealing with imbalanced binary data is over- or under-sampling as well as synthetic minority over-sampling technique (SMOTE) [Chawla et al., 2002]. In our case, these approaches do not improve the accuracy of the binary classifiers that we consider (results not reported). Presumably, the reason for this is that with the Grabit model, one can learn additional structure from the auxiliary data that is not present in the binary data. In contrast, over- and under-sampling or also SMOTE, cannot use the auxiliary data and, consequently, cannot learn this structure.

Note that we implicitly assume that multiple loans from the same borrower are independent conditional on the predictor variables. We do account for potential borrower

---

5We obtain the following choices for the second best tuning parameters. Boosted Logit: \( \nu = 0.1, M = 10, L = 3 \). Boosted multiclass Logit: \( \nu = 0.1, M = 1000, L = 5 \). Grabit: \( \nu = 0.01, M = 1000, L = 3, \sigma = 1 \). Random forest: \( M = 1000, L = 5, \rho = 0.5 \).
specific dependence by including the repayment history of a company, such as the number of loans repaid, total amount repaid, maximum past delays, and others, in the covariates. Even if there is a certain amount of additional residual dependence, we believe that this is not a problem in our case for the following reasons. First, given that in linear regression models, coefficient estimates are unbiased also in the presence of correlation, it is likely that the learned mean function in the Grabit model is also unbiased and not impacted by correlation. Further, our goal in this application is univariate default prediction. If the goal is the prediction of aggregate default rates of, e.g., loan portfolios, over a longer time period this assumption might not be applicable. In this case, a simple solution for prediction is to assume implicit cross default provisions and declare all loans as nonperforming once a borrower is predicted to default on one of its loans. Finally, the reason why we use loans and not borrowers as observation units is that, given the small sample size, it is not possible to aggregate to the borrower level at, e.g., yearly frequency. However, the distribution of the number of loans per defaulted borrower is not skewed, and we do not have the problem that a few borrowers with many loans might disproportionately influence the results. Nonetheless, we have also considered loan amount weighed ROCs and AUROCs and obtained similar performance gains with the Grabit model compared to the other approaches (results not reported).

4.3 Model interpretation

In the following, we illustrate the use of partial dependence plots for interpreting a learned Grabit model. Partial dependence plots are calculated as described in Section 2.5.2.
In Figure 7 we show partial dependence plots illustrating main effects for four selected variables with high variable importance. The four variables are a rating from a rating agency (the higher the better is the credit worthiness of a company), a repayment score calculated by Advanon (the higher the better is the repayment history of a company), the age of the accounting data provided by a company, as well as the maximum number of days of delay by which a company did repay its past loans. The results from this are in line with expectations about default probabilities: companies with higher ratings have a lower default probability. Similarly, a better repayment history and the fact that companies provide newer accounting data is related to a lower default probability. Three of the four effects are clearly non-linear. In Figure 8 we additionally show two examples
of two-dimensional partial dependence plots illustrating second-order effects.

Figure 8: Partial dependence plots illustrating two second-order effects.

5 Conclusions

The Grabit model is a flexible, non-linear censored regression model that can be applied to various modeling tasks. In particular, this includes binary classification in situations where there is class imbalance in combination with a relatively small sample size but auxiliary data, which is related to the classification mechanism, is available. We have shown in a simulation study and in our default prediction application that the Grabit model can provide substantial and significant gains in predictive accuracy.

Future research should investigate the performance of the Grabit model on other data sets with class imbalance and/or relatively small sample sizes. As we show in the simulation study, whether the sample size is too small and the class imbalance is too large for traditional classification methods depends on the form of the latent decision function. If the decision function is of sufficiently complex containing, e.g., non-linearities, interactions, or a large number of predictor variables, the Grabit model can also provide increased predictive accuracy for moderately to large sized data.

The Grabit model can be extended in several ways. For instance, one can relax the assumption of a constant $\sigma^2$ by also relating the variance in Equation (1) to an ensemble of regression trees as it is done for the mean. Further, the normality assumption for
the latent variable in Equation (1) can be relaxed by using another density that is twice
differentiable in the parameter that is related to the tree ensemble. In general, one can also
replace the Tobit loss function by another appropriate loss function, for instance, in order
to make the model robust against outliers in the dependent variable. Finally, as we have
mentioned before, we implicitly assume that multiple loans from the same borrower are
independent conditional on the predictor variables. Future research can investigate how
potential correlation can be accounted for in the Grabit model or in other machine learning
type models in general.

Acknowledgements

We gratefully acknowledge funding by the Commission for Technology and Innovation
(CTI) under grant number '25746.1 PFES-ES'. Further, we thank the associate editor as
well as two anonymous reviewers for constructive comments and suggestions that helped
to improve the paper. We also would like to thank Benjamin Christoffersen for interesting
discussions and Dirk Tasche for feedback on the article.

References

E. Alfaro, N. García, M. Gámez, and D. Elizondo. Bankruptcy forecasting: An empirical
comparison of adaboost and neural networks. Decision Support Systems, 45(1):110–122,
2008.

E. I. Altman. Financial ratios, discriminant analysis and the prediction of corporate
bankruptcy. The Journal of Finance, 23(4):589–609, 1968.

E. I. Altman. Bankruptcy, credit risk, and high yield junk bonds. Wiley-Blackwell, 2002.

E. I. Altman and G. Sabato. Modelling credit risk for smes: Evidence from the us market.
Abacus, 43(3):332–357, 2007.

T. Amemiya. Advanced econometrics. Harvard university press, 1985.
J. Bauer and V. Agarwal. Are hazard models superior to traditional bankruptcy prediction approaches? a comprehensive test. *Journal of Banking & Finance*, 40:432–442, 2014.

W. H. Beaver. Financial ratios as predictors of failure. *Journal of Accounting Research*, pages 71–111, 1966.

J. L. Bellovary, D. E. Giacomino, and M. D. Akers. A review of bankruptcy prediction studies: 1930 to present. *Journal of Financial education*, pages 1–42, 2007.

L. Breiman. Arcing classifiers. *Annals of Statistics*, pages 801–824, 1998.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

I. Brown and C. Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446–3453, 2012.

P. Bühlmann and T. Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007.

P. Bühlmann and B. Yu. Boosting with the l 2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845, 1988.
A. A. Ding, S. Tian, Y. Yu, and H. Guo. A class of discrete transformation survival models with application to default probability prediction. *Journal of the American Statistical Association*, 107(499):990–1003, 2012.

J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.

P. J. FitzPatrick. A comparison of the ratios of successful industrial enterprises with those of failed companies. *Certified Public Accountant*, 2:598–605, 1932.

Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

S. Jones, D. Johnstone, and R. Wilson. Predicting corporate bankruptcy: An evaluation of alternative statistical frameworks. *Journal of Business Finance & Accounting*, 44(1-2):3–34, 2017.

A. H.-L. Lau. A five-state financial distress prediction model. *Journal of Accounting Research*, pages 127–138, 1987.

S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.
G. S. Maddala. *Limited-dependent and qualitative variables in econometrics*. Number 3. Cambridge university press, 1986.

A. Mayr, H. Binder, O. Gefeller, and M. Schmid. The evolution of boosting algorithms. *Methods of information in medicine*, 53(06):419–427, 2014a.

A. Mayr, H. Binder, O. Gefeller, and M. Schmid. Extending statistical boosting. *Methods of information in medicine*, 53(06):428–435, 2014b.

C. L. Merwin et al. Financing small corporations in five manufacturing industries, 1926-36. *NBER Books*, 1942.

P. G. Moffatt. Hurdle models of loan default. *Journal of the Operational Research Society*, 56(9):1063–1071, 2005.

L. E. Papke and J. M. Wooldridge. Econometric methods for fractional response variables with an application to 401 (k) plan participation rates. *Journal of Applied Econometrics*, 11(6):619–632, 1996.

L. E. Papke and J. M. Wooldridge. Panel data methods for fractional response variables with an application to test pass rates. *Journal of Econometrics*, 145(1):121–133, 2008.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

R. N. Rosett and F. D. Nelson. Estimation of the two-limit probit regression model. *Econometrica: Journal of the Econometric Society*, pages 141–146, 1975.

M. Sandri and P. Zuccolotto. A bias correction algorithm for the gini variable importance measure in classification trees. *Journal of Computational and Graphical Statistics*, 17(3):611–628, 2008.

B. Sanso and L. Guenni. Venezuelan rainfall data analysed by using a bayesian space–time model. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):345–362, 1999.
T. Shumway. Forecasting bankruptcy more accurately: A simple hazard model. *The Journal of Business*, 74(1):101–124, 2001.

F. Sigrist, H. R. Künsch, and W. A. Stahel. A dynamic nonstationary spatio-temporal model for short term prediction of precipitation. *The Annals of Applied Statistics*, pages 1452–1477, 2012.

S. Tian, Y. Yu, and H. Guo. Variable selection and corporate bankruptcy forecasts. *Journal of Banking & Finance*, 52:89–100, 2015.

J. Tobin. Estimation of relationships for limited dependent variables. *Econometrica: journal of the Econometric Society*, pages 24–36, 1958.

P. R. Vahid and A. Ahmadi. Modeling corporate customers’ credit risk considering the ensemble approaches in multiclass classification: evidence from iranian corporate credits. *Journal of Credit Risk*, 12(3):71–95, 2016.

A. Winakor and R. Smith. Changes in the financial structure of unsuccessful industrial corporations. *Bureau of Business Research Urbana, Bulletin*, 51, 1935.

J. M. Wooldridge. Simple solutions to the initial conditions problem in dynamic, nonlinear panel data models with unobserved heterogeneity. *Journal of Applied Econometrics*, 20 (1):39–54, 2005.

Y. Xia, C. Liu, Y. Li, and N. Liu. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241, 2017.

Y. Yang, W. Qian, and H. Zou. Insurance premium prediction via gradient tree-boosted tweedie compound poisson models. *Journal of Business & Economic Statistics*, pages 1–15, 2017.

M. Zieba, S. K. Tomczak, and J. M. Tomczak. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, 58:93–101, 2016.
Appendix

Tobit loss as well as first and second derivatives

The negative log-likelihood $L(y_i, F(x_i))$ of the Tobit model, i.e., the loss used in the Grabit model, for one observation $(y_i, x_i)$ is given by

$$L(y_i, F(x_i)) = - \log \left( \Phi \left( \frac{y_l - F(x_i)}{\sigma} \right) \right) \mathbb{I}_{y_l}(y_i)$$
$$+ \left( \frac{(y_i - F(x_i))^2}{2\sigma^2} + \log(\sigma) + 0.5 \log(2\pi) \right) \mathbb{1}_{\{y_i < y_u\}}$$
$$- \log \left( 1 - \Phi \left( \frac{y_u - F(x_i)}{\sigma} \right) \right) \mathbb{I}_{y_u}(y_i).$$

The gradient $\frac{\partial L(y_i, F)}{\partial F}$ and the second derivative $\frac{\partial^2 L(y_i, F)}{\partial^2 F}$ of the Tobit loss are given by:

$$\frac{\partial L(y_i, F)}{\partial F} = \frac{\Phi \left( \frac{y_l - F(x_i)}{\sigma} \right)}{\sigma \Phi \left( \frac{y_l - F(x_i)}{\sigma} \right)} \mathbb{I}_{y_l}(y_i) - \frac{y_i - F(x_i)}{\sigma^2} \mathbb{1}_{\{y_i < y_u\}}$$
$$- \frac{\Phi \left( \frac{y_u - F(x_i)}{\sigma} \right)}{\sigma \left( 1 - \Phi \left( \frac{y_u - F(x_i)}{\sigma} \right) \right)} \mathbb{I}_{y_u}(y_i).$$ (11)

and

$$\frac{\partial^2 L(y_i, F)}{\partial^2 F} = \frac{\Phi \left( \frac{y_l - F(x_i)}{\sigma} \right)}{\sigma^2 \Phi^2 \left( \frac{y_l - F(x_i)}{\sigma} \right)} \left( \frac{y_l - F(x_i)}{\sigma} \right) \Phi \left( \frac{y_l - F(x_i)}{\sigma} \right) + \frac{\Phi \left( \frac{y_l - F(x_i)}{\sigma} \right)}{\sigma^2} \mathbb{I}_{y_l}(y_i)$$
$$+ \frac{1}{\sigma^2} \mathbb{1}_{\{y_l < y_u\}}$$
$$- \frac{\Phi \left( \frac{y_u - F(x_i)}{\sigma} \right)}{\sigma^2 \left( 1 - \Phi \left( \frac{y_u - F(x_i)}{\sigma} \right) \right)^2} \left( 1 - \Phi \left( \frac{y_u - F(x_i)}{\sigma} \right) \right) \Phi \left( \frac{y_u - F(x_i)}{\sigma} \right) - \frac{\Phi \left( \frac{y_u - F(x_i)}{\sigma} \right)}{\sigma^2} \mathbb{1}_{y_u}(y_i).$$ (12)
For calculating the gradient and the second derivative, one can use the following relations:

\[
\frac{\partial \Phi \left( \frac{y_i - F(x_i)}{\sigma} \right)}{\partial F} = -\frac{1}{\sigma} \phi \left( \frac{y_i - F(x_i)}{\sigma} \right)
\]

and

\[
\frac{\partial \phi \left( \frac{y_i - F(x_i)}{\sigma} \right)}{\partial F} = \left( \frac{y_i - F(x_i)}{\sigma^2} \right) \phi \left( \frac{y_i - F(x_i)}{\sigma} \right).
\]
Results for non-optimal tuning parameters

Figure 9: Comparison of different models using receiver operating characteristic (ROC) and area under ROC (AUROC) when using non-optimal tuning parameters for the tree-boosted methods (boosted Logit, boosted multiclass Logit, Grabit) and the random forest.