DET
RAN
OTE

DETRA NOTE 2023-5

WASSERSTEIN BOOSTING TREES ALGORITHM FOR COUNT DATA

WITH APPLICATION TO CLAIM FREQUENCIES IN MOTOR INSURANCE

Michel Denuit, Julien Trufin and Harrison Verelst

Detralytics

DISCLAIMER

The content of the Detra Notes for a pedagogical use only. Each business case is so specific that a careful analysis of the situation is needed before implementing a possible solution. Therefore, Detralytics does not accept any liability for any commercial use of the present document. Of course, the entire team remain available if the techniques presented in this Detra Note required your attention.

Detralytics
Avenue du Boulevard 21 Box 5 - 1210 Bruxelles (Belgium)
1-7 Cour Valmy, Le Belvédère - 92800 Puteaux (France)
www.detralytics.com
info@detralytics.eu

ABSTRACT

This paper proposes a variant of the well-known boosting trees algorithm to estimate conditional distributions. Since regression trees partition observations into subgroups, the corresponding empirical distributions can be used to define the splitting criterion. Precisely, the parametric approach using Poisson deviance is replaced with a non-parametric one maximizing probabilistic distances between empirical distributions in child nodes. Proceeding in this way, the actuary obtains an estimated conditional distribution for the response, from which a conditional mean can be derived as well as any other quantity of interest in risk management. The numerical performances of the proposed method are assessed with simu-lated data while a case study demonstrates its usefulness for insurance applications.

Keywords: Wasserstein distance, regression trees, boosting, conditional distribution, count data.

1 Introduction

The present paper proposes to use probabilistic distances in the splitting criterion to grow regression trees, in a boosting approach. Boosting emerged from the field of machine learning and became rapidly popular among actuaries. Broadly speaking, boosting is an iterative procedure building the score sequentially, adding terms that weakly improve on the estimations obtained from the preceding step. Regression trees with limited depth are often used as weak learners in insurance studies, following Friedman (2001). We refer the reader to Breiman et al. (1984) for a general introduction to trees.

Boosting requires the specification of a loss function to be optimized for estimating the score. In that respect, actuaries generally adopt a parametric approach by selecting loss functions corresponding to negative log-likelihood, or deviance functions. For count data, Poisson or Negative Binomial distributions are often used to study numbers of claims reported by policyholders to their insurer. We refer the readers to Denuit et al. (2020) for an extensive treatment of boosting in the context of insurance.

Since trees partition the database in groups of observations corresponding to terminal leaves, many other quantities of interest can be computed from their empirical distribution, not only conditional means. Also, considering that trees produce estimated distributions in each terminal leave opens the door to new partition rules based on distances between the resulting empirical distributions. Chapter 9 in Denuit et al. (2005) offers a general presentation of the topic with applications to actuarial science. Probability metrics have been extensively used in risk theory, to assess the quality of the approximation of the individual model by its collective counterpart. Here, the idea is to split a group in two subgroups to make the corresponding empirical distributions the furthest apart according to some relevant probabilistic distance.

Probabilistic distances have already been used in combination with regression trees, considering random forests. Du et al. (2021) proposed to adapt Breiman's original splitting criterion in terms of Wasserstein distance between empirical distributions. Their Wasserstein random forest algorithm then produces an estimate of the conditional distribution function, allowing the actuary to derive a variety of quantities of interest, not just the conditional mean. Given the excellent performances of boosting algorithms in insurance studies, the present paper proposes a Wasserstein boosting trees algorithm. Instead of averaging (in the sense of probabilistic mixtures) estimation obtained from a large number of trees, the conditional distribution is learned sequentially, by updating estimation at previous step.

The remainder of this paper is organized as follows. Section 2 recalls probabilistic distances, with special emphasis on the Wasserstein distance that has been successfully used to grow trees by Du et al. (2021). Section 3 then proposes a new boosting algorithm adopting this distance in the splitting criterion. Section 4 illustrates the numerical performances of this new approach on simulated data. A case study with motor insurance claim frequencies is performed in Section 5. The final Section 6 briefly concludes the paper, summarizing the main findings and discussing possible extensions.

2 Wasserstein distance

Probability metrics, including the Wasserstein one, have successfully been used in actuarial science to assess the accuracy of the approximation of the individual model by its collective counterpart. Given two random variables V and W with respective probability distributions π_V and π_W, recall that the Wasserstein distance $\mathcal{W}$, also known as the Dudley or Kantorovitch distance, is defined as

$$\mathcal{W}(\pi_V, \pi_W) = \int_{-\infty}^{\infty} \left|\overline{F}_V(t) - \overline{F}_W(t)\right| \mathrm{d}t = \int_0^1 \left|F_V^{-1}(u) - F_W^{-1}(u)\right| \mathrm{d}u$$

where $\overline{F}_V = 1 - F_V$ denotes the excess function corresponding to the distribution function F_V of V and where the quantile function F_V^{-1} is defined for a probability level u as

$$F_V^{-1}(u) = \inf\{v \in \mathbb{R} | F_V(v) \geq u\}.$$

Since the representation $\mathcal{W}(\pi_V, \pi_W) = \mathrm{E}[|F_W^{-1}(U) - F_V^{-1}(U)|]$ holds true with U uniformly distributed over the unit interval, $\mathcal{W}(\pi_V, \pi_W)$ appears to be the lower bound on $\mathrm{E}[|K - L|]$ over all random couples (K, L) with marginal distribution functions F_V and F_W (attained when K and L are perfectly positively dependent, or comonotonic).

For two counting random variables M and N with respective probability distributions π_M and π_N, the Wasserstein distance $\mathcal{W}$ can be simply rewritten as

$$\mathcal{W}(\pi_M, \pi_N) = \sum_{j=0}^{\infty} \left|\overline{F}_M(j) - \overline{F}_N(j)\right|.$$

It can easily be computed when M and N only assume a few positive values, as it is the case for claim frequencies at policy-level in personal lines. The series appearing in $\mathcal{W}(\pi_M, \pi_N)$ then reduces to just a few terms and is fast to compute.

Property 9.6.3 in Denuit et al. (2005) shows that $\mathcal{W}$ reduces to the difference of expectations under stochastic dominance. Recall that N is smaller than M in stochastic dominance, denoted as $N \preceq_{\mathrm{ST}} M$, when the inequality $\mathrm{P}[N > t] \leq \mathrm{P}[M > t]$ holds for all real t. It is then easy to see that

$$N \preceq_{\mathrm{ST}} M \quad \Rightarrow \quad \mathcal{W}(\pi_M, \pi_N) = \int_0^1 \left(F_M^{-1}(u) - F_N^{-1}(u)\right) \mathrm{d}u = \mathrm{E}[M] - \mathrm{E}[N].$$

Wasserstein distance thus reduces to the difference in pure premiums in that case. For instance, if N obeys the Poisson distribution with mean λ_1 and M obeys the Poisson distribution with mean λ_2,

$$\lambda_1 < \lambda_2 \Rightarrow N \preceq_{\mathrm{ST}} M \Rightarrow \mathcal{W}(\pi_M, \pi_N) = \lambda_2 - \lambda_1.$$

Considering claim counts M and N, the Wasserstein distance is the sum, over each possible claims number value j, of the difference in probability to report strictly more than j claims. In a frequency context, one would like to be able to distinguish policyholders with low probability to have claims or low probability to have a large number of claims with policyholders with high probability to have one or more than one claim. Using the Wasserstein distance in this context thus seems to be appealing. Since Poisson distributions are ordered with respect to $\preceq_{\mathrm{ST}}$ in their mean parameter, maximizing Wasserstein distance amounts to maximize differences in expected claim frequencies.

3 Wasserstein boosting trees

3.1 Setting

Supervised learning aims to provide an estimation of the conditional expectation $\mu(\boldsymbol{x}) = \mathrm{E}\left[Y|\boldsymbol{X}=\boldsymbol{x}\right]$ for a response Y and a set of features $\boldsymbol{X} = (X_1,\ldots,X_p) \in \chi \subseteq \mathbb{R}^p$. In many applications of practical relevance, the additional information encoded in the conditional distribution function $F_Y(\cdot|\boldsymbol{x})$ defined as $F_Y(y|\boldsymbol{x}) = \mathrm{P}[Y \leq y|\boldsymbol{X}=\boldsymbol{x}]$ is also of great interest. Du et al. (2021) propose natural variants of random forests to estimate $F_Y(\cdot|\boldsymbol{x})$, called Wasserstein random forests. In this paper, we present a variant of boosting trees to estimate $F_Y(\cdot|\boldsymbol{x})$ for count data. Precisely, the response Y is assumed to be discrete with values in $\mathbb{N} = \{0,1,2,\ldots\}$. We denote by $\pi(\boldsymbol{x},\cdot)$ the probability mass function associated with the conditional distribution function $F_Y(\cdot|\boldsymbol{x})$, i.e. $\pi(\boldsymbol{x},k) = \mathrm{P}[Y=k|\boldsymbol{X}=\boldsymbol{x}]$, $k \in \mathbb{N}$. We aim to estimate the conditional probabilities $\pi(\boldsymbol{x},k)$ using a boosting approach given a data set $\mathcal{D} = \{(y_i,\boldsymbol{x}_i), i=1,\ldots,n\}$.

We denote by M the number of trees in the ensemble and for $1 \leq m \leq M$, let Θ_m be the canonical random variable which fully captures the randomness of the m-th tree. Each decision tree is built on a random sample of the training set $\mathcal{D}$, denoted $\mathcal{D}^*(\Theta_m)$, taken without replacement. The fraction of training set used at each iteration to produce the random samples is the bagging fraction and is denoted by α. A typical value for α is 0.5.

We denote by $\chi_{j(\boldsymbol{x})}(\Theta_m)$ the subspace of the feature space χ induced by the m-th tree that contains $\boldsymbol{x}$ and by $N(\boldsymbol{x};\Theta_m)$ the number of observations $(y_i,\boldsymbol{x}_i)$ in $\mathcal{D}^*(\Theta_m)$ such that $\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_m)$. As proposed in Du et al. (2021), the estimate $\widehat{\pi}(\boldsymbol{x},k;\Theta_m)$ of the conditional probability $\pi(\boldsymbol{x},k)$ at point $\boldsymbol{x}$ given by the m-th tree is the empirical measure associated with the observations $(y_i,\boldsymbol{x}_i)$ that fall into the same terminal node as $\boldsymbol{x}$ (i.e. such that $\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_m)$), that is

$$\widehat{\pi}(\boldsymbol{x},k;\Theta_m) = \sum_{(y_i,\boldsymbol{x}_i)\in\mathcal{D}^*(\Theta_m)} \frac{\mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_m), y_i = k\right]}{N(\boldsymbol{x};\Theta_m)}, \tag{3.1}$$

where $\mathrm{I}[\cdot]$ denotes the indicator function, equal to 1 if the event appearing in the brackets is realized and to 0 otherwise.

3.2 Boosting algorithm

We initialize the boosting algoritm with the estimate

$$\widehat{\pi}(\boldsymbol{x},k;\Theta_0) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}\left[y_i = k\right] \tag{3.2}$$

corresponding to the empirical distribution of the response. We thus start with the same probability mass function for every $\boldsymbol{x}$. The starting point (3.2) corresponds to the null model comprising only an intercept initializing the boosting procedure.

From the initial estimate $\widehat{\pi}(\boldsymbol{x},k;\Theta_0)$, we build the first tree and we compute the estimate $\widehat{\pi}(\boldsymbol{x},k;\Theta_1)$, as given by (3.1) with $m=1$. The initial estimate $\widehat{\pi}(\boldsymbol{x},k;\Theta_0)$ and the estimate from the first tree $\widehat{\pi}(\boldsymbol{x}, k; \Theta_1)$ are then combined to produce the estimate of the conditional probability $\pi(\boldsymbol{x}, k)$ after the first tree of the boosting algorithm. We denote this estimate as $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]})$. The latter is given by

$$\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]}) = \widehat{c}(\boldsymbol{x}; \Theta_1) \times \widehat{\beta}(\boldsymbol{x}, k; \Theta_1) \times \widehat{\pi}(\boldsymbol{x}, k; \Theta_0), \tag{3.3}$$

where $\widehat{\beta}(\boldsymbol{x}, k; \Theta_1)$ satisfies

$$\frac{1}{N(\boldsymbol{x}; \Theta_1)} \sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_1)} \mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_1)\right] \left(\widehat{\pi}(\boldsymbol{x}_i, k; \Theta_0) \times \widehat{\beta}(\boldsymbol{x}, k; \Theta_1)\right) = \widehat{\pi}(\boldsymbol{x}, k; \Theta_1), \tag{3.4}$$

that is,

$$\widehat{\beta}(\boldsymbol{x}, k; \Theta_1) \quad = \quad \frac{N(\boldsymbol{x}; \Theta_1)\, \widehat{\pi}(\boldsymbol{x}, k; \Theta_1)}{\sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_1)} \mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_1)\right] \widehat{\pi}(\boldsymbol{x}_i, k; \Theta_0)}, \tag{3.5}$$

and where the normalizing constant

$$\widehat{c}(\boldsymbol{x}; \Theta_1) \quad = \quad \left(\sum_{k=0}^{\infty} \widehat{\beta}(\boldsymbol{x}, k; \Theta_1) \times \widehat{\pi}(\boldsymbol{x}, k; \Theta_0) \right)^{-1} \tag{3.6}$$

guarantees that $\sum_{k=0}^{\infty} \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]}) = 1$. Similarly to how the score in the classical Boosting algorithm is updated at each step, the function $\widehat{\beta}(\boldsymbol{x}, k; \Theta_1)$ defined in (3.5) corrects the initial estimate $\widehat{\pi}(\boldsymbol{x}, k; \Theta_0)$ with the information $\widehat{\pi}(\boldsymbol{x}, k; \Theta_1)$ gathered from the first tree to produce the next estimate $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]})$.

From the estimate after the first tree $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]})$, we build the second tree and compute from this second tree the estimate $\widehat{\pi}(\boldsymbol{x}, k; \Theta_2)$. Similarly to the first step, $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]})$ is updated by $\widehat{\pi}(\boldsymbol{x}, k; \Theta_2)$ to produce the estimate of the conditional probability $\pi(\boldsymbol{x}, k)$ resulting from the first two trees. This estimate is denoted as $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[2]})$ and is given by

$$\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[2]}) = \widehat{c}(\boldsymbol{x}; \Theta_2) \times \widehat{\beta}(\boldsymbol{x}, k; \Theta_2) \times \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]}), \tag{3.7}$$

where $\widehat{\beta}(\boldsymbol{x}, k; \Theta_2)$ satisfies

$$\frac{1}{N(\boldsymbol{x}; \Theta_2)} \sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_2)} \mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_2)\right] \left(\widehat{\pi}(\boldsymbol{x}_i, k; \boldsymbol{\Theta}_{[1]}) \times \widehat{\beta}(\boldsymbol{x}, k; \Theta_2)\right) = \widehat{\pi}(\boldsymbol{x}, k; \Theta_2), \tag{3.8}$$

that is,

$$\widehat{\beta}(\boldsymbol{x}, k; \Theta_2) \quad = \quad \frac{N(\boldsymbol{x}; \Theta_2)\, \widehat{\pi}(\boldsymbol{x}, k; \Theta_2)}{\sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_2)} \mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_2)\right] \widehat{\pi}(\boldsymbol{x}_i, k; \boldsymbol{\Theta}_{[1]})}, \tag{3.9}$$

and where the normalizing constant

$$\widehat{c}(\boldsymbol{x}; \Theta_2) \quad = \quad \left(\sum_{k=0}^{\infty} \widehat{\beta}(\boldsymbol{x}, k; \Theta_2) \times \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[1]}) \right)^{-1} \tag{3.10}$$

guarantees that $\sum_{k=0}^{\infty} \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[2]}) = 1$.

In general, let $\boldsymbol{\Theta}_{[m]} = (\Theta_1, \ldots, \Theta_m)$, $m = 1, \ldots, M$ and $\boldsymbol{\Theta}_{[0]} = \Theta_0$. The estimate of the conditional probability $\pi(\boldsymbol{x}, k)$ produced by the first m trees of the boosting algorithm, denoted $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]})$, is given by

$$\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]}) = \widehat{c}(\boldsymbol{x}; \Theta_m) \times \widehat{\beta}(\boldsymbol{x}, k; \Theta_m) \times \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m-1]}), \tag{3.11}$$

where $\widehat{\beta}(\boldsymbol{x}, k; \Theta_m)$ satisfies

$$\frac{1}{N(\boldsymbol{x}; \Theta_m)} \sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_m)} \mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_m)\right] \left(\widehat{\pi}(\boldsymbol{x}_i, k; \boldsymbol{\Theta}_{[m-1]}) \times \widehat{\beta}(\boldsymbol{x}, k; \Theta_m)\right) = \widehat{\pi}(\boldsymbol{x}, k; \Theta_m), \tag{3.12}$$

that is,

$$\widehat{\beta}(\boldsymbol{x}, k; \Theta_m) \quad = \quad \frac{N(\boldsymbol{x}; \Theta_m)\, \widehat{\pi}(\boldsymbol{x}, k; \Theta_m)}{\sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_m)} \mathrm{I}\left[\boldsymbol{x}_i \in \chi_{j(\boldsymbol{x})}(\Theta_m)\right] \widehat{\pi}(\boldsymbol{x}_i, k; \boldsymbol{\Theta}_{[m-1]})}, \tag{3.13}$$

and where the normalizing constant

$$\widehat{c}(\boldsymbol{x}; \Theta_m) \quad = \quad \left(\sum_{k=0}^{\infty} \widehat{\beta}(\boldsymbol{x}, k; \Theta_m) \times \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m-1]})\right)^{-1} \tag{3.14}$$

guarantees that $\sum_{k=0}^{\infty} \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]}) = 1$.

At iteration m, the first $m-1$ trees produce the estimate $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m-1]})$ of the conditional probability $\pi(\boldsymbol{x}, k)$ and the m-th tree provides the estimate $\widehat{\pi}(\boldsymbol{x}, k; \Theta_m)$ of $\pi(\boldsymbol{x}, k)$. These two estimates are then combined as described in (3.11) and (3.12) to produce the new estimate $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]})$ of $\pi(\boldsymbol{x}, k)$ resulting from the first m trees. Contrary to $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m-1]})$, the new estimate $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]})$ guarantees that its average over the observations that fall into a terminal node j of the m-th tree corresponds to the empirical estimate $\widehat{\pi}(\boldsymbol{x}, k; \Theta_m)$ associated with that terminal node j.

3.3 Splitting rule

The splitting criterion that is maximized at each node of the m-th tree is based on the Wasserstein distance. Suppose that the node under consideration in the m-th tree consists in a subset ϕ of the feature space χ. Hence, any standardized binary split defines a partition $\phi_L \cup \phi_R$ of ϕ. Let N (resp. N_L and N_R) be the number of observations in $\mathcal{D}^*(\Theta_m)$ such that $\boldsymbol{x}_i \in \phi$ (resp. $\boldsymbol{x}_i \in \phi_L$ and $\boldsymbol{x}_i \in \phi_R$). The splitting criterion we consider here takes the form

$$L(\phi_L, \phi_R) = \frac{N_L}{N} \mathcal{W}(\widehat{\pi}_L, \widehat{\pi}_L^{[m-1]}) + \frac{N_R}{N} \mathcal{W}(\widehat{\pi}_R, \widehat{\pi}_R^{[m-1]}), \tag{3.15}$$

where

$$\widehat{\pi}_L^{[m-1]}(k) = \frac{1}{N_L} \sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^*(\Theta_m)} \mathrm{I}\left[\boldsymbol{x}_i \in \phi_L\right] \widehat{\pi}(\boldsymbol{x}_i, k; \boldsymbol{\Theta}_{[m-1]}) \tag{3.16}$$

$$\widehat{\pi}_R^{[m-1]}(k) = \frac{1}{N_R} \sum_{(y_i,\boldsymbol{x}_i)\in\mathcal{D}^*(\Theta_m)} \mathrm{I}\left[\boldsymbol{x}_i \in \phi_R\right] \widehat{\pi}(\boldsymbol{x}_i, k; \boldsymbol{\Theta}_{[m-1]}) \tag{3.17}$$

and

$$\widehat{\pi}_L(k) = \frac{1}{N_L} \sum_{(y_i,\boldsymbol{x}_i)\in\mathcal{D}^*(\Theta_m)} \mathrm{I}\left[\boldsymbol{x}_i \in \phi_L, y_i = k\right] \tag{3.18}$$

$$\widehat{\pi}_R(k) = \frac{1}{N_R} \sum_{(y_i,\boldsymbol{x}_i)\in\mathcal{D}^*(\Theta_m)} \mathrm{I}\left[\boldsymbol{x}_i \in \phi_R, y_i = k\right]. \tag{3.19}$$

At each node of the m-th tree, the best split maximizes the Wasserstein distances between the probability distributions already estimated from the first $m-1$ trees and the empirical distributions resulting from the split. The partition $\phi_L \cup \phi_R$ of ϕ obtained from our splitting criterion is such that the current estimates $\widehat{\pi}_L^{[m-1]}$ and $\widehat{\pi}_R^{[m-1]}$ are "the most different" from the empirical estimates $\widehat{\pi}_L$ and $\widehat{\pi}_R$, respectively, as measured by the Wasserstein distance. The m-th tree is thus built in a way that the resulting partition $\chi_j(\Theta_m)$ of the feature space highlights subsets of the feature space for which the current estimates obtained after $m-1$ iterations are not satisfactory and hence need to be readjusted.

3.4 Shrinkage parameters

In boosting, it is often helpful to slow down the learning speed by adding only a small amount of the fit of the best-performing base-learner to the current additive score. This is achieved by multiplying the new effect entering the score with a shrinkage coefficient (a typical value is 0.1). This also appears to be useful in the Wasserstein boosting trees algorithm, as explained next.

The idea is to decrease $\widehat{\beta}(\boldsymbol{x}, k; \Theta_m)$ to slow down the learning process. Formally, let τ_1 and τ_2 be such that $0 \leq \tau_1 < 1$ and $\tau_2 > 1$. Define

$$\widehat{\beta}^{\,\text{shrink}}(\boldsymbol{x}, k; \Theta_m) = \min\left(\max\left(\widehat{\beta}(\boldsymbol{x}, k; \Theta_m), \tau_1\right), \tau_2\right), \tag{3.20}$$

where $\widehat{\beta}(\boldsymbol{x}, k; \Theta_m)$ is given in (3.13). The updating procedure then becomes

$$\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]}) = \widehat{c}(\boldsymbol{x}; \Theta_m) \times \widehat{\beta}^{\,\text{shrink}}(\boldsymbol{x}, k; \Theta_m) \times \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m-1]}), \tag{3.21}$$

with

$$\widehat{c}(\boldsymbol{x}; \Theta_m) \;\; = \;\; \left(\sum_{k=0}^{\infty} \widehat{\beta}^{\,\text{shrink}}(\boldsymbol{x}, k; \Theta_m) \times \widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m-1]})\right)^{-1}. \tag{3.22}$$

With the proposed shrinkage procedure, we have $\tau_1 \leq \widehat{\beta}^{\,\text{shrink}}(\boldsymbol{x}, k; \Theta_m) \leq \tau_2$, so that we slow down the learning rate of the boosting procedure.

4 Numerical illustrations with simulated data sets

In this section, we illustrate the newly proposed Wasserstein boosting trees on several examples. In the first two examples, we show based on simple distribution functions that the estimated distributions obtained from Wasserstein boosting trees converge towards the true distributions. Then, we compare Wasserstein boosting trees with gradient boosting trees on two examples. In the first example, we simulate the numbers of claims from Poisson distributions and compare the performances of the Wasserstein boosting trees algorithm with the gradient boosting trees algorithm using the Poisson deviance as loss function. The second example demonstrates the advantage of the Wasserstein Boosting trees algorithm over the gradient boosting trees algorithm by simulating the numbers of claims from Negative Binomial distributions with the same mean.

4.1 Example 1

In this very simple example, three categorical features $\boldsymbol{X} = (X_1, X_2, X_3)$ are supposed to be available:

- X_1 = Age: policyholder's seniority with two levels, labeled as young (y) or old (o), with 50% of the portfolio in each level;

- X_2 = Gender: policyholder's gender with two levels, labeled as female (f) or male (m), with 50% of the portfolio in each level;

- X_3 = Sport: whether the policyholder's car is a sports car or not with two levels, labeled as yes or no, with 50% of the portfolio in each level.

The eight risk classes are depicted in Table 1. The variables X_1, X_2 and X_3 are assumed to be mutually independent.

The response Y is supposed to be the number of claims. The conditional probabilities $\pi(\boldsymbol{x}, k)$ are given in Table 1. The eight risk classes have the same possible values but have different associated probabilities. The associated probabilities are arbitrarily chosen. In this example, the true conditional probabilities $\pi(\boldsymbol{x}, k)$ are thus known and we can simulate realizations of the random vector $(Y, \boldsymbol{X})$. Specifically, we generate $n = 10000$ independent realizations $(y_1, \boldsymbol{x}_1), (y_2, \boldsymbol{x}_2), \ldots$ of $(Y, \boldsymbol{X})$. A simulation represents a policy that has been observed during a whole year.

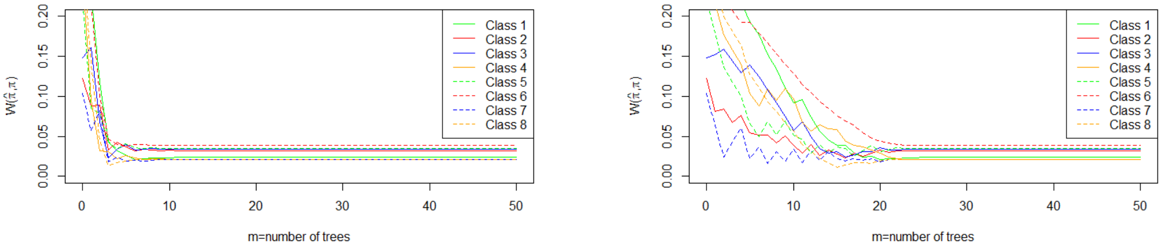
For each tree, a sequence of standardized binary splits is made recursively by maximizing the splitting rule discussed in Section 3.3. The size of the trees is controlled by the interaction depth `ID`. Trees with an interaction depth equal to `ID` correspond to trees with `ID` non-terminal nodes. By setting $\texttt{ID} = 1$, only single-split regression trees are produced, which allows capturing only the main effects of the features in the score. For $\texttt{ID} = 2$, two-way interactions are permitted, and for $\texttt{ID} = 3$, three-way interactions are allowed, and so on. A larger value of `ID` allows the model to learn deeper patterns in the data. In this way the m-th tree partitions the feature space χ into disjoint subsets $\{\chi_j(\Theta_m), j = 1, \ldots, \texttt{ID} + 1\}$.

We run the Wasserstein boosting trees algorithm with $\texttt{ID} \in \{1, 2, 3\}$ and shrinkage parameters $\tau_1 = 1 - \tau$ and $\tau_2 = 1 + \tau$ with $\tau \in \{0.5, 0.05\}$. The results are depicted in Figure

	$\pi(\boldsymbol{x},k)$					
	$\boldsymbol{x}$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
Class 1	$X_1 = y,\ X_2 = m,\ X_3 = yes$	0.400	0.400	0.100	0.050	0.050
Class 2	$X_1 = y,\ X_2 = m,\ X_3 = no$	0.650	0.150	0.100	0.050	0.050
Class 3	$X_1 = o,\ X_2 = m,\ X_3 = yes$	0.600	0.300	0.050	0.025	0.025
Class 4	$X_1 = o,\ X_2 = m,\ X_3 = no$	0.800	0.100	0.050	0.025	0.025
Class 5	$X_1 = y,\ X_2 = f,\ X_3 = yes$	0.500	0.300	0.050	0.100	0.050
Class 6	$X_1 = y,\ X_2 = f,\ X_3 = no$	0.750	0.050	0.050	0.100	0.050
Class 7	$X_1 = o,\ X_2 = f,\ X_3 = yes$	0.700	0.175	0.050	0.050	0.025
Class 8	$X_1 = o,\ X_2 = f,\ X_3 = no$	0.850	0.050	0.025	0.050	0.025

Table 1: $\pi(\boldsymbol{x},k)$ for the eight risk classes.

1 (for ID $=1$), Figure 2 (for ID $=2$) and Figure 3 (for ID $=3$). For each risk profile $\boldsymbol{x}$, we show the Wasserstein distance between the true conditional probabilities $\pi(\boldsymbol{x},k)$ and the estimated ones $\widehat{\pi}(\boldsymbol{x},k;\boldsymbol{\Theta}_{[m]})$ with respect to the number of iterations m. With ID $\in\{1,2\}$, one sees that for any risk class, the estimate $\widehat{\pi}(\boldsymbol{x},k;\boldsymbol{\Theta}_{[m]})$ does not allow to recover the true conditional probability $\pi(\boldsymbol{x},k)$ whatever the number of iteration m. For ID $=3$, one sees that for any risk class, the estimate $\widehat{\pi}(\boldsymbol{x},k;\boldsymbol{\Theta}_{[m]})$ tends to the conditional probability $\pi(\boldsymbol{x},k)$ in the sense that $\mathcal{W}(\pi(\boldsymbol{x},.),\widehat{\pi}(\boldsymbol{x},.;\boldsymbol{\Theta}_{[m]}))$ tends to 0 after 10 iterations for $\tau=0.5$ and after 30 iterations for $\tau=0.05$. This means that for ID $=3$, starting from the empirical distribution of the response for every $\boldsymbol{x}$ and correcting after each fitted tree the previously estimated conditional distribution $\widehat{\pi}(\boldsymbol{x},.;\boldsymbol{\Theta}_{[m-1]})$ with the information coming from the new fitted tree $\widehat{\pi}(\boldsymbol{x},.;\Theta_m)$, we recover the true conditional probability $\pi(\boldsymbol{x},k)$. The smaller the shrinkage parameter is, the slowest the learning rate and the smoothest the convergence as it prevents large corrections at each step.



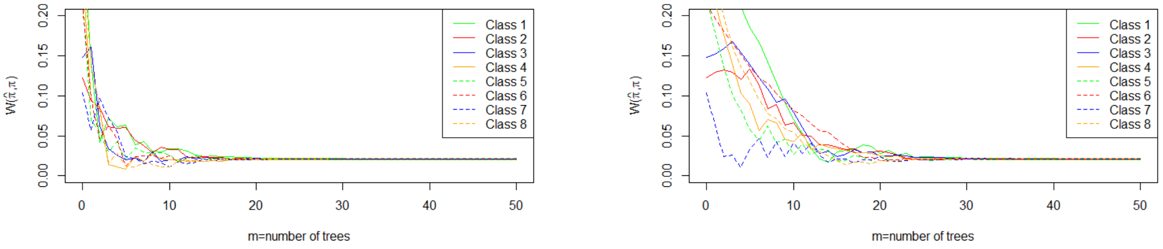
(a) $\tau=0.5$

(b) $\tau=0.05$

Figure 1: Wasserstein distance between $\pi(\boldsymbol{x},.)$ and $\widehat{\pi}(\boldsymbol{x},.;\boldsymbol{\Theta}_{[m]})$ (obtained with ID $=1$) with respect to the number of trees m for the eight risk classes.

4.2 Example 2

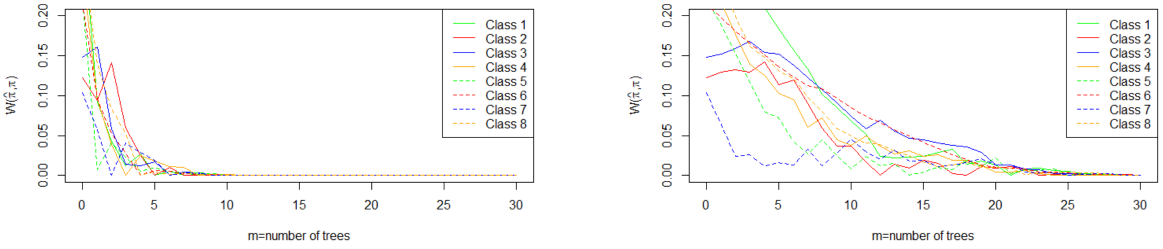
We consider the same example as in the previous section, except that the conditional probabilities $\pi(\boldsymbol{x},k)$ for risk classes 7 and 8 are now given in Table 2. Compared to Table 1,



(a) $\tau = 0.5$

(b) $\tau = 0.05$

Figure 2: Wasserstein distance between $\pi(\boldsymbol{x}, .)$ and $\widehat{\pi}(\boldsymbol{x}, .; \boldsymbol{\Theta}_{[m]})$ (obtained with $\texttt{ID} = 2$) with respect to the number of trees m for the eight risk classes.



(a) $\tau = 0.5$

(b) $\tau = 0.05$

Figure 3: Wasserstein distance between $\pi(\boldsymbol{x}, .)$ and $\widehat{\pi}(\boldsymbol{x}, .; \boldsymbol{\Theta}_{[m]})$ (obtained with $\texttt{ID} = 3$) with respect to the number of trees m for the eight risk classes.

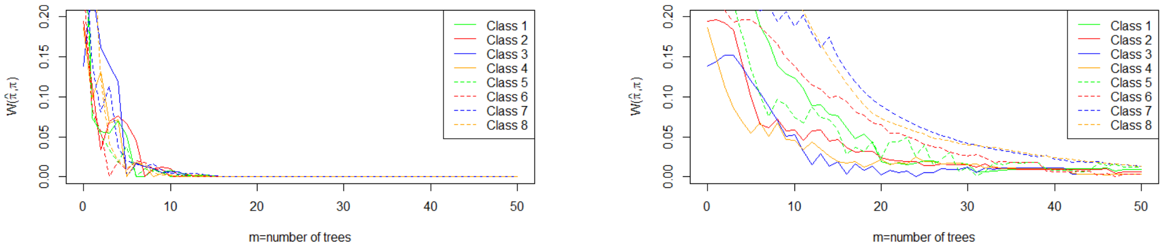
only the values 0 and 1 are allowed for Y for risk classes 7 and 8 in Table 2. The results are shown in Figure 4 for $\texttt{ID} = 3$. Again, one sees that the Wasserstein boosting trees algorithm enables to recover the true conditional probabilities $\pi(\boldsymbol{x}, k)$ for all risk classes.

$\pi(\boldsymbol{x}, k)$						
	$\boldsymbol{x}$	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
Class 7	$X_1 = o,\ X_2 = f,\ X_3 = yes$	0.700	0.300	0.000	0.000	0.000
Class 8	$X_1 = o,\ X_2 = f,\ X_3 = no$	0.950	0.050	0.000	0.000	0.000

Table 2: $\pi(\boldsymbol{x}, k)$ for risk classes 7 and 8.

4.3 Comparison with the gradient boosting tree algorithm

The Wasserstein boosting trees algorithm enables to estimate the conditional probabilities $\pi(\boldsymbol{x}, k)$ in a non-parametric way. In particular, it allows the actuary to estimate the conditional expectation $\mu(\boldsymbol{x})$ from the corresponding conditional distribution. In insurance pricing, actuaries generally apply learning procedures like the gradient boosting trees algorithm to estimate the conditional expectation $\mu(\boldsymbol{x})$, assuming that the responses Y follows



(a) $\tau = 0.5$ (b) $\tau = 0.05$

Figure 4: Wasserstein distance between $\pi(\boldsymbol{x}, k)$ and $\widehat{\pi}(\boldsymbol{x}, k; \boldsymbol{\Theta}_{[m]})$ (obtained with $\texttt{ID} = 3$) with respect to the number of trees m for the eight risk classes.

a counting distribution such as Poisson or Negative Binomial. In this section, we compare the results obtained with the Wasserstein boosting trees algorithm proposed in this paper to the ones derived with the gradient boosting trees algorithm using the Poisson deviance as loss function.

4.3.1 Example 1 : response obeying Poisson distribution

We consider an example in car insurance with the following four features:

- $X_1 =$ Gender: policyholder's gender with two levels, labeled as female (f) or male (m), with 50% of the portfolio in each level;
- $X_2 =$ Sport: whether the policyholder's car is a sports car or not, with two levels (yes or no) and 50% of the portfolio in each level;
- $X_3 =$ Split: whether the policyholder splits its annual premium or not, with two levels (yes or no) and 50% of the portfolio in each level;
- $X_4 =$ Age: policyholder's age (integer values from 18 to 65) with the same proportion 1/48 of the portfolio in each value.

The features X_1, X_2, X_3 and X_4 are assumed to be mutually independent.

The response Y is supposed to be the number of claims. Given $\boldsymbol{X} = \boldsymbol{x}$, Y is assumed to be Poisson distributed with expected claim frequency given by

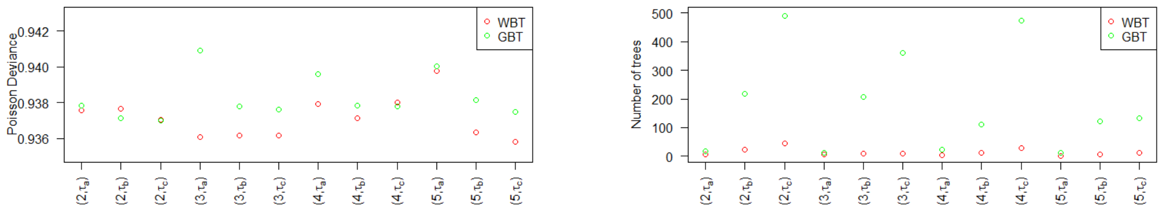
$$\begin{aligned}\mu(\boldsymbol{x}) &= 0.3 \times (1 + 0.1\mathrm{I}\left[x_1 = male\right]) \times \left(1 + \frac{1}{\sqrt{x_4 - 17}}\right) \\ &\quad \times (1 + 0.3\,\mathrm{I}\left[x_2 = yes\right]\mathrm{I}\left[18 \le x_4 < 35\right] - 0.3\,\mathrm{I}\left[x_2 = yes\right]\mathrm{I}\left[45 \le x_4 \le 65\right]) .\end{aligned}$$

The true model $\mu(\boldsymbol{x})$ is known and we can simulate realizations of $(Y, \boldsymbol{X})$. Specifically, we generate $n = 30\,000$ independent realizations of $(Y, \boldsymbol{X})$. We use 80% of the observations for training the models (training set) and 20% for assessing the performance of the models (validation set).

We run the Wasserstein boosting trees (henceforth abbreviated as WBT) algorithm and the gradient boosting trees (henceforth referred to as GBT) algorithm with $\texttt{ID} \in \{2,3,4,5\}$, a bag fraction $\alpha = 0.5$ and shrinkage parameters $\tau_1 = 1-\tau$ and $\tau_2 = 1+\tau$ with $\tau \in \{\tau_a, \tau_b, \tau_c\} = \{0.5, 0.1, 0.05\}$ for the WBT algorithm and $\tau \in \{\tau_a, \tau_b, \tau_c\} = \{1, 0.1, 0.05\}$ for the GBT algorithm. GBT is fitted using the Poisson deviance loss.

Out-of-sample Poisson deviance

To compare the predictive accuracy of both algorithms, we compute the Poisson deviance on the validation set. Figures 5a and 5b display out-of-sample Poisson deviance for GBT and WBT. We observe that in most cases WBT outperforms GBT in the sense that the out-of-sample deviance for WBT is smaller than the one for the corresponding GBT. Figure 5b shows that the number of trees required for WBT can be much smaller than the number of trees for GBT. The lower the shrinkage parameter, the greater the difference of the number of trees between both algorithms. As expected, for both methods, more trees are needed when the shrinkage parameter is decreased.



(a) Poisson Deviance

(b) Number of trees

Figure 5: Out-of-sample Poisson deviance and number of trees for GBT (in green) and WBT (in red) for each set of parameters $(\texttt{ID}, \tau_\ell)$, $\ell = a, b, c$.

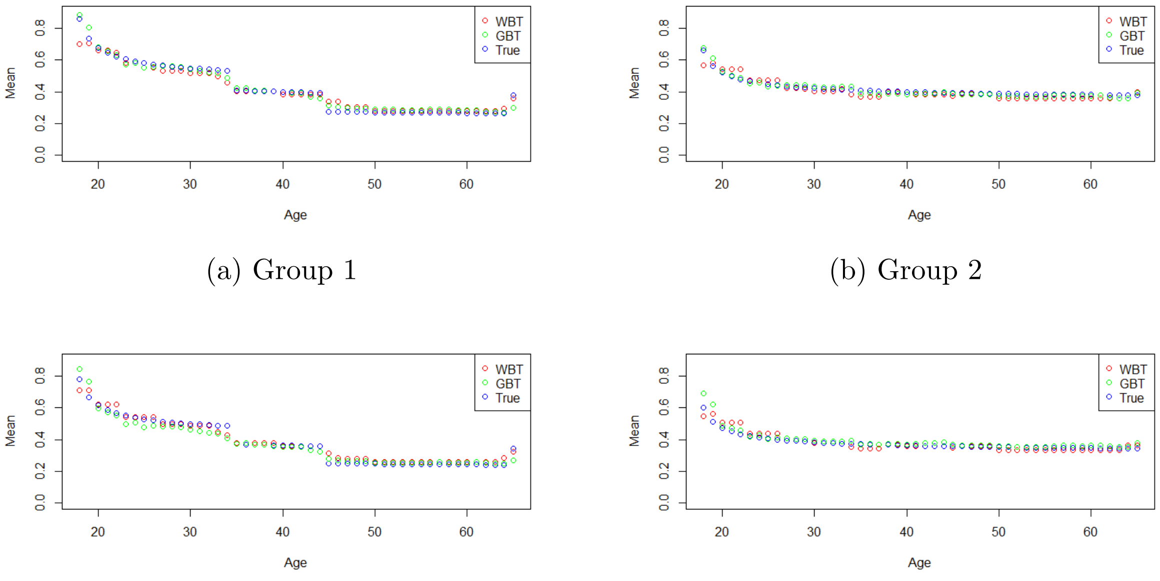
Estimated means

We compare the estimated conditional mean $\widehat{\mu}(\boldsymbol{x})$ to the true one $\mu(\boldsymbol{x})$ for each risk class. For this comparison, we consider the best WBT and GBT models according to the out-of-sample Poisson deviance, namely WBT with $\texttt{ID} = 5$, $\tau = 0.05$ and $M = 13$ and GBT with $\texttt{ID} = 2$, $\tau_3 = 0.05$ and $M = 490$.

Since x_3 does not influence $\mu(\boldsymbol{x})$ in our example, we consider the following four groups:

- Group 1: Male driving a sports car;
- Group 2 : Male driving a regular car;
- Group 3 : Female driving a sports car;
- Group 4 : Female driving a regular car.

For each group, Figures 6a to 6d display the estimated means with respect to the age of the policyholder for the best WBT and GBT together with the true means. One sees that both models accurately estimate the conditional means $\mu(\boldsymbol{x})$. For most values of $\boldsymbol{x}$, both estimated conditional means are closed to each other and closed to the true conditional mean $\mu(\boldsymbol{x})$.



(a) Group 1

(b) Group 2

(c) Group 3

(d) Group 4

Figure 6: Estimated means with respect to the age for each group for WBT (in red) and GBT (in green) together with the true means (in blue).

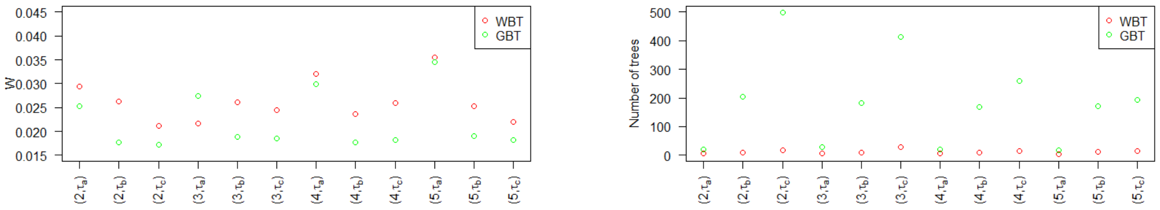
Estimated conditional distributions

The GBT and WBT algorithms perform similarly in this example when it comes to estimating the conditional mean $\mu(\boldsymbol{x})$. However, WBT does not directly estimate the conditional mean $\mu(\boldsymbol{x})$, it rather focuses on the estimation of the conditional distribution $\pi(\boldsymbol{x}, .)$ without imposing any rigid parametric assumption for $\pi(\boldsymbol{x}, .)$ whereas GBT assumes that the response is Poisson distributed.

Let us now compare the estimated conditional distributions $\widehat{\pi}_{\text{WBT}}(\boldsymbol{x}, .)$ obtained from the WBT algorithm with the true Poisson distribution $\pi(\boldsymbol{x}, .)$ on the one hand, and the estimated conditional distribution $\widehat{\pi}_{\text{GBT}}(\boldsymbol{x}, .)$ obtained from the GBT algorithm (which assumes that $\pi(\boldsymbol{x}, .)$ is Poisson) with the true Poisson distribution $\pi(\boldsymbol{x}, .)$ on the other hand. To do so, we use the "average Wasserstein distance" computed on the validation set. For each observation $(y_i, \boldsymbol{x}_i)$ of the validation set $\mathcal{D}^V$, we compute the Wasserstein distance between its estimated conditional distribution $\widehat{\pi}(\boldsymbol{x}_i, .)$ and its true conditional distribution $\pi(\boldsymbol{x}_i, .)$, namely $\mathcal{W}_1(\widehat{\pi}(\boldsymbol{x}_i, .), \pi(\boldsymbol{x}_i, .))$. Then, the average Wasserstein distance is computed as

$$\mathcal{AW} = \frac{1}{|\mathcal{D}^V|} \sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^V} \mathcal{W}(\widehat{\pi}(\boldsymbol{x}_i, .), \pi(\boldsymbol{x}_i, .)), \tag{4.1}$$

where $|\mathcal{D}^V|$ is the number of observations in $\mathcal{D}^V$.



(a) Average Wasserstein Distance $\mathcal{AW}$

(b) Number of trees

Figure 7: Comparison of minimum average Wasserstein Distance $\mathcal{AW}$ and corresponding number of trees between WBT and GBT for each set of parameter (ID, τ_ℓ), $\ell = a, b, c$.

As expected, we see in Figure 7a that GBT with a small shrinkage parameter outperforms WBT since the true conditional distribution is Poisson (as assumed by the GBT algorithm). However, the WBT algorithm, which is in essence non-parametric, also succeeds in estimating the true Poisson distributions. Indeed, one sees that the average Wasserstein distance is smaller than 0.03 in most cases, that is,

$$\mathcal{AW} = \frac{1}{|\mathcal{D}^V|} \sum_{(y_i, \boldsymbol{x}_i) \in \mathcal{D}^V} \sum_{j=0}^{\infty} \left| F_{\widehat{\pi}_{\mathrm{WBT}}}(j) - F_{\pi}(j) \right| \leq 0.03 \text{ in most cases.}$$

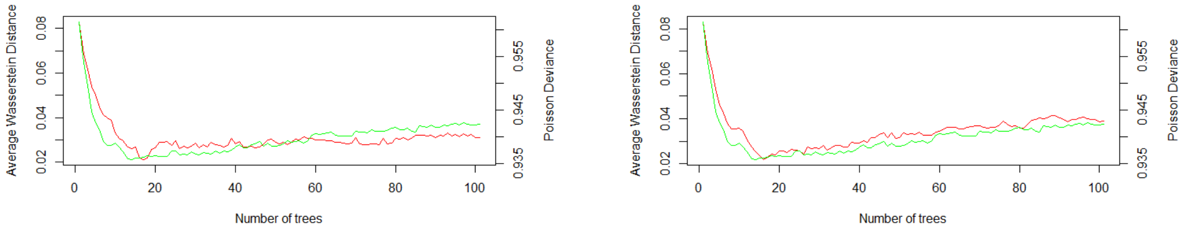
The estimation is good in the sense that on average, the cumulative distribution function estimated by WBT is close to the true cumulative distribution function.

Moreover, it is interesting to notice that the average Wasserstein distances and the out-of-sample Poisson deviances reach their minimum values for a similar number of trees. This can be seen in Figures 8a and 8b where we compare the out-of-sample Poisson deviance and the average Wasserstein distance for two WBT models with `ID` $= 2$, $\tau = 0.05$ and WBT with `ID` $= 5$, $\tau = 0.05$.

4.3.2 Example 2 : response obeying Negative Binomial distribution

Let us assume that conditionally on $\boldsymbol{X}$, the response is no more Poisson distributed but follows Negative Binomial distribution. This time, two different values $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of $\boldsymbol{X}$ such that $\mu(\boldsymbol{x}_1) = \mu(\boldsymbol{x}_2)$ may have different associated distributions $\pi(\boldsymbol{x}_1, .)$ and $\pi(\boldsymbol{x}_2, .)$ since the Negative Binomial distribution is governed by two parameters (the mean and the dispersion parameter).

Consider a simple example with two categorical features, each one with only two levels, say policyholder's age (X_1) either "young" or "old" and policyholder's gender (X_2) either



(a) `ID` $= 2$ and $\tau = 0.05$

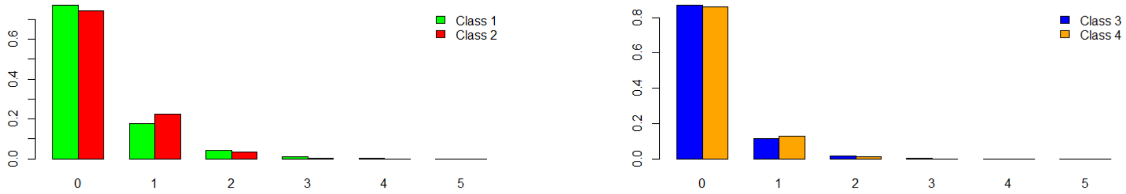
(b) `ID` $= 5$ and $\tau = 0.05$

Figure 8: Comparison of the average Wasserstein distance (in red) and the out-of-sample Poisson deviance (in green) for two models.

"male" or "female". For each of the four risk classes, we assume that the corresponding distribution $\pi(\boldsymbol{x}, .)$ follows a Negative Binomial (NB) distribution with the parameters given in Table 3. The probability mass functions for each class are given on Figure 9. Under Poisson assumption, risk classes 1-2 and 3-4 cannot be distinguished because mean responses are equal. These risk classes nevertheless differ in their residual heterogeneity.

	$\boldsymbol{x}$	$\pi(\boldsymbol{x}, .)$
Class 1	$X_1 = y$, $X_2 = m$	$Y\|\boldsymbol{X} = \boldsymbol{x} \sim NB\left(1, \frac{1}{1+0.3}\right)$, such that $\mathrm{E}[Y\|\boldsymbol{X} = \boldsymbol{x}] = 0.3$
Class 2	$X_1 = o$, $X_2 = m$	$Y\|\boldsymbol{X} = \boldsymbol{x} \sim NB\left(1, \frac{500}{500+0.3}\right)$, such that $\mathrm{E}[Y\|\boldsymbol{X} = \boldsymbol{x}] = 0.3$
Class 3	$X_1 = y$, $X_2 = f$	$Y\|\boldsymbol{X} = \boldsymbol{x} \sim NB\left(1, \frac{1}{1+0.015}\right)$, such that $\mathrm{E}[Y\|\boldsymbol{X} = \boldsymbol{x}] = 0.15$
Class 4	$X_1 = o$, $X_2 = f$	$Y\|\boldsymbol{X} = \boldsymbol{x} \sim NB\left(1, \frac{500}{500+0.015}\right)$, such that $\mathrm{E}[Y\|\boldsymbol{X} = \boldsymbol{x}] = 0.15$

Table 3: $\pi(\boldsymbol{x})$ for the four risk classes.



(a) Class 1 (green) and Class 2 (red)

(b) Class 3 (blue) and Class 4 (orange)

Figure 9: Comparison of the probability mass function between Class 1 and Class 2 and between Class 3 and Class 4.

Figure 10 shows the Wasserstein distance between the estimated conditional distribution $\widehat{\pi}(\boldsymbol{x}, .)$ and the true Negative Binomial distribution $\pi(\boldsymbol{x}, .)$ for the four risk classes. We see that the WBT algorithm enables to accurately estimate the true conditional distribution for any risk class (after 20 iterations, $\mathcal{W}\left(\widehat{\pi}(\boldsymbol{x}, .), \pi(\boldsymbol{x}, .)\right) = 0$ for all $\boldsymbol{x}$). If we use the GBT algorithm with the typical Poisson assumption for the condition distribution $\pi(\boldsymbol{x}, .)$, we cannot distinguish risk profiles of Class 1 (resp. Class 3) from risk profiles of Class 2 (resp. Class 4), as shown in Figure 11.

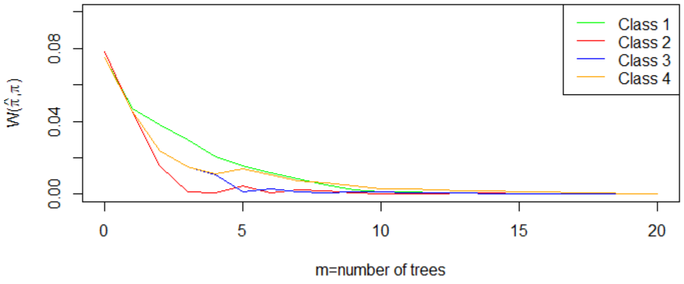


Figure 10: Wasserstein distance between the estimated distribution $\widehat{\pi}(\boldsymbol{x},.)$ and the true distribution $\pi(\boldsymbol{x},.)$ for the four risk classes with `ID` $= 2$ and $\tau_2 = 0.01$.

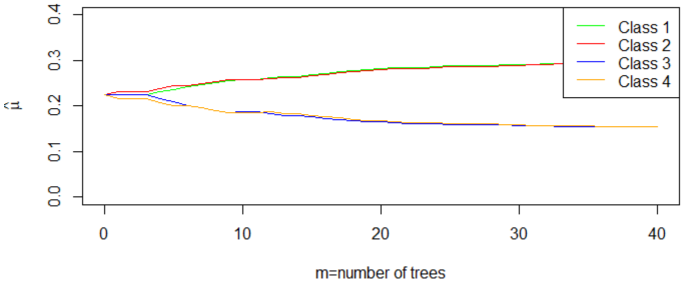


Figure 11: Estimated mean $\widehat{\mu}(\boldsymbol{x})$ with the GBT algorithm for the four risk classes with `ID` $= 2$ and $\tau_2 = 0.01$.

5 Case study with claim frequencies in motor insurance

5.1 MTPL data set

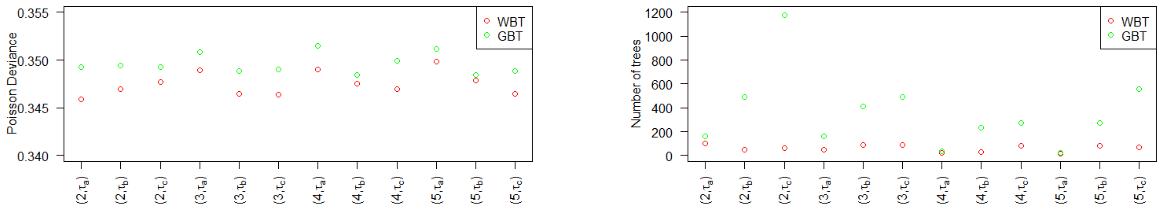
Let us consider the `freMTPL2freq` data set comprised in the `CASdatasets` package in `R`. It contains 168 125 observations (observed over exactly one year) of the number of claims (response Y) in a French motor third-party liability insurance portfolio, together with nine features ($\boldsymbol{X} = (X_1, \ldots, X_9)$). The latter correspond to several characteristics related to the policyholder (age, density of inhabitants in the home city, region, area, bonus-malus) and his or her vehicle (power, age, brand, fuel type). We refer to Noll et al. (2018) for an accurate description of the data set.

The training set comprising 80% of the data is used to train the models. The validation set with the remaining 20% of the data allows us to compare the different models on data that have not been used to train the models.

We run the WBT and GBT algorithms with $\texttt{ID} \in \{2,3,4,5\}$, a bag fraction $\alpha = 0.5$ and shrinkage parameters $\tau_1 = 1 - \tau$ and $\tau_2 = 1 + \tau$ with $\tau \in \{\tau_a, \tau_b, \tau_c\} = \{0.5, 0.1, 0.05\}$ for WBT and $\tau \in \{\tau_a, \tau_b, \tau_c\} = \{1, 0.1, 0.05\}$ for GBT. Moreover, GBT is fitted with the Poisson deviance loss.

5.2 Out-of-sample Poisson deviance

Figures 12a and 12b show out-of-sample Poisson deviance for GBT and WBT. One sees that the WBT algorithm outperforms the GBT algorithm in any case. This demonstrates the outstanding performances of the newly proposed WBT approach which achieves lower out-of-sample Poisson deviance compared to GBT algorithm whereas the latter minimizes Poisson deviance while the former maximizes Wasserstein distance. In addition, we observe again that the number of trees for WBT is much smaller than the number of trees for GBT.



(a) Poisson Deviance

(b) Number of trees

Figure 12: Out-of-sample Poisson deviance and number of trees for GBT (in green) and WBT (in red) for each set of parameter $(\texttt{ID}, \tau_\ell)$, $\ell = a, b, c$.

5.3 Overdispersion

The WBT algorithm does not make any assumption on the underlying conditional distribution of the number of claims whereas GBT assumes that claim counts are Poisson distributed. Let us now test whether the observations in the final nodes of the Wasserstein boosting trees are Poisson distributed. To this end, we use the Poisson dispersion test which compares the sample mean to the sample variance. If the sample mean is very different from the sample variance, then the number of claims in the terminal node under consideration does not follow a Poisson distribution. The Poisson dispersion test statistics is defined as

$$P_n = \sum_{i=1}^{n} \frac{(y_i - \bar{y})^2}{\bar{y}}, \tag{5.1}$$

where $\bar{y}$ is the sample mean and n is the sample size. This test statistics approximately obeys the Chi-Square distribution with $n-1$ degrees of freedom. The null hypothesis that the number of claims in the node under consideration is Poisson distributed is rejected for large values of P_n.

Table 4 displays rejection frequencies for each model (at level 5%) in the terminal nodes of the trees of the ensemble. It is interesting to notice that the smaller the trees (i.e. the smaller the interaction depth), the larger the frequency of rejection of the null hypothesis. This means that the more data we have in a terminal node, the more the Poisson hypothesis is rejected. With a 5% confidence level, when we have an $ID = 5$ or 6 terminal nodes, the Poisson distribution is rejected one time out of three. This probability doubles when we only have 3 terminal nodes. At a global level, the Poisson hypothesis for the claim count distribution does not seem to be valid. This is less obvious at a local level, when less data are available.

Model	Poisson rejection frequency
$\mathtt{ID} = 2$, $\tau_1 = 0.5$	63%
$\mathtt{ID} = 2$, $\tau_2 = 0.1$	62%
$\mathtt{ID} = 2$, $\tau_3 = 0.05$	63%
$\mathtt{ID} = 3$, $\tau_1 = 0.5$	48%
$\mathtt{ID} = 3$, $\tau_2 = 0.1$	52%
$\mathtt{ID} = 3$, $\tau_3 = 0.05$	54%
$\mathtt{ID} = 4$, $\tau_1 = 0.5$	39%
$\mathtt{ID} = 4$, $\tau_2 = 0.1$	45%
$\mathtt{ID} = 4$, $\tau_3 = 0.05$	43%
$\mathtt{ID} = 5$, $\tau_1 = 0.5$	32%
$\mathtt{ID} = 5$, $\tau_2 = 0.1$	31%
$\mathtt{ID} = 5$, $\tau_3 = 0.05$	31%

Table 4: Frequency of rejection for the null hypothesis in the terminal nodes of the trees of the ensemble.

6 Conclusion

This paper proposes a new boosting algorithm for claim counts, based on Wasserstein distance. Compared to existing procedures targeting conditional means, WBT produces estimated conditional distributions which allow the actuary to derive any estimation of interest. When applied to simulated data and a classical industry data base, WBT shows excellent performances compared to GBT. The non-parametric nature of the newly proposed WBT allows the actuary to recover patterns present in the data beyond conditional means, such as residual heterogeneity compared to Poisson distribution for instance. In our examples, WBT even outperforms GBT in terms of Poisson deviance despite GBT precisely optimizes that metric.

The Poisson distribution has been extended in different ways to accommodate the specific aspects of claim frequency distributions: Overdispersed Poisson or Poisson mixtures, including Negative Binomial, have been proposed to account for overdispersion, Zero-inflated Poisson or Hurdle models to account for the probability mass at zero, etc. Machine learning tools have been developed for each model but the approach remains parametric and thus exposes actuaries to model risk. The latter has been extensively studied for the conditional mean, showing that Poisson regression delivers consistent estimates even if responses do not obey the Poisson distribution as long as the conditional mean is correctly specified. However, this is not necessarily true for other parameters of interest, like the no-claim probability extensively used in underwriting. Being non-parametric by nature, the WBT algorithm proposed in this paper avoids this pitfall and can be used to challenge the conclusions obtained from parametric regression.

In this paper, we restricted ourselves to integral probability metrics because they turn out to be computed more rapidly compared to metrics defined as supremum (like Kolmogorov distance, for instance, corresponding to the largest absolute difference between respective distribution functions). Also, we paid particular attention to their expressions with count data. Of course, there are other candidates for being used as distance in splitting criterion. For counting random variables M and N, the Wasserstein distance is closely related to the total-variation distance $\mathcal{TV}(\pi_M, \pi_N)$ given by

$$\mathcal{TV}(\pi_M, \pi_N) = \sum_{k=0}^{\infty} \big|\mathrm{P}[M = k] - \mathrm{P}[N = k]\big|.$$

When M and N refer to claim frequencies at policy-level in personal lines, the series reduces to just a few terms and is thus easy to compute. Proposition 9.6.5 in Denuit et al. (2005) shows that $\mathcal{TV}(\pi_M, \pi_N) \leq 2\mathcal{W}(\pi_M, \pi_N)$.

Besides $\mathcal{W}$ and $\mathcal{TV}$, the integrated stop-loss distance is another candidate to assess the proximity of two distributions in insurance studies. The integrated difference in stop-loss premiums has traditionally been used to measure the distance between two risks with finite variances in the actuarial literature. Given two random variables V and W, the integrated stop-loss distance $\mathcal{ISL}$ is given by

$$\mathcal{ISL}(\pi_V, \pi_W) = \int_{-\infty}^{\infty} \big|\mathrm{E}[(V - t)_+] - \mathrm{E}[(W - t)_+]\big| \mathrm{d}t,$$

where the integrand is the absolute difference of the stop-loss transforms of V and of W. This distance is easy to compute under stop-loss or convex order since Proposition 9.8.2 in Denuit et al. (2005) shows that

(i) If $\mathrm{E}[(V - t)_+] \leq \mathrm{E}[(W - t)_+]$ holds for all t then $\mathcal{ISL}(\pi_V, \pi_W) = \frac{1}{2}\big(\mathrm{E}[W^2] - \mathrm{E}[V^2]\big)$.

(ii) If $\mathrm{E}[(V - t)_+] \leq \mathrm{E}[(W - t)_+]$ holds for all t and $\mathrm{E}[V] = \mathrm{E}[W]$ then $\mathcal{ISL}(\pi_V, \pi_W) = \frac{1}{2}\big(\mathrm{Var}[W] - \mathrm{Var}[V]\big)$.

Considering Poisson mixtures, these results apply when mixing distributions satisfy the conditions appearing in (i)-(ii). Under (ii), we see that $\mathcal{ISL}$ coincides with the classical splitting criterion used in regression trees.

This paper thus opens a new approach to statistical learning in insurance, replacing deviance with probabilistic distance. Depending on the application, the actuary is free to select the most appropriate distance. Considering insurance ratemaking, Wasserstein distance appears to be a natural candidate complying with intuition.

References

- Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). Classification and Regression Trees. Wadsworth Statistics/Probability Series.

- Denuit, M., Dhaene, J., Goovaerts, M.J., Kaas, R. (2005). Actuarial Theory for Dependent Risks: Measures, Orders and Models. Wiley, New York.

- Denuit, M., Hainaut, D., Trufin, J. (2020). Effective Statistical Effective Statistical Learning Methods for Actuaries II: Tree-based Methods and Extensions. Springer Actuarial Lecture Notes Series.

- Du, Q., Biau, G., Petit, F., Porcher, R. (2021). Wasserstein Random Forests and applications in heterogeneous treatment effects. International Conference on Artificial Intelligence and Statistics, 1729-1737.

- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics 29, 1189-232.

- Noll, A., Salzmann, R., Wüthrich, M. (2018). Case study: French motor third-party liability claims. Available at SSRN: https://ssrn.com/abstract=3164764.

Detralytics

People drive actuarial innovation

www.detralytics.com - info@detralytics.eu