Gradient boosted mixture models and their applications in insurance loss prediction

Guangyuan Gao Jiahong Li

October 22, 2021

Abstract

The Expectation-Maximization (EM) algorithm is often used to estimate parameters in mixture of regression models. In this paper, we propose a gradient boosting algorithm for mixture of models, which combines the EM algorithm and functional gradient decent techniques. The proposed algorithm fully explores the predictive power of covariates and performs variable selection during model fitting. We illustrate those advantages in two simulated examples and a real data set: a simulated mixture of Gaussians, a simulated zero-inflated Poisson, and a real insurance claims amount data.

1 Introduction

Insurance loss data usually cannot be well modelled by a single distribution. For claims counts data, there may be an excess of zero claims, so a Poisson distribution is not the best option. For claims amount data, there may be a heavy tail, so a gamma distribution is not enough to describe the entire data. One solution to such issues is to apply mixture models. For claims counts data, we can utilize zero-modified Poisson distribution. For claims amount data, we can utilize mixture of gamma distribution and heavy-tailed distribution such as Pareto distribution. When the individual risk factors are available, we can extend mixture of distributions to mixture of regressions to address the risk heterogeneity in the portfolio.

Parameter estimation in mixture models is challenging since each component distribution/regression parameters are related to each other. The expectation-maximization algorithm is an iterative method to estimate component parameters and (hidden) component indicator variable. Variable selection and component selection in mixture models is also challenging. And we always performs variable selection and component selection after parameter estimation.

In this paper, we propose an expectation-boosting (EB) algorithm, which replaces the maximization step by an overfitting-sensitive boosting step. There are several advantages of EB algorithm over the EM algorithm. First, boosting algorithm is a flexible non-parametric regression facilitating both non-linear effects and interaction. Second, boosting algorithm is overfitting-sensitive, if we add suitable normalization to boosting step we can perform variable selection simultaneously during the EB algorithm. Third, by investigating the final boosted component, we can select component by removing those components containing very few weak learners.

2 Review of mixed distributions

We review the mixed distributions and the EM algorithm. We mixed K different EDF densities f_k by

$$Y \sim \sum_{k=1}^{K} p_k f_k\left(y; \theta_k, v/\varphi_k\right) = \sum_{k=1}^{K} p_k \exp\left\{\frac{y\theta_k - \kappa_k\left(\theta_k\right)}{\varphi_k/v} + a_k\left(y; v/\varphi_k\right)\right\} \tag{2.1}$$

with canonical parameter θ_k, exposure $v_k > 0$, dispersion parameters $\varphi_k > 0$ and $\sum_{k=1}^{K} p_k = 1$. If we consider the log-likelihood function of n observations

$$\mathcal{D} = \{(Y_1, \boldsymbol{x}_1, v_1), (Y_n, \boldsymbol{x}_2, v_2), \ldots, (Y_n, \boldsymbol{x}_n, v_n)\},$$

the log-likelihood function of mixed distribution (2.1) is given by

$$\ell_{\boldsymbol{Y}}(\boldsymbol{\theta}, \boldsymbol{p}) = \sum_{i=1}^{n} \ell_{Y_i}(\boldsymbol{\theta}, \boldsymbol{p}) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} p_{ki} f_{ki}\left(Y_i; \theta_{ki}, v_i/\varphi_k\right)\right), \tag{2.2}$$

where both canonical parameter $\theta_{ki} = h_k(\mu_k(\boldsymbol{x}_i))$ and $p_{ki} = p_k(\boldsymbol{x}_i)$ are regression functions. Mixed distribution can be defined as (2.1) and also can be defined in a more constructive way using a latent variable Z. Z is a categorical r.v. having probabilities $\Pr[Z = k] = p_k > 0$ for $k = 1, \ldots, K$. We use Z to form the mixed distribution i.e. we first choose $Z \in \{1, \cdots, K\}$ and $Y|Z = k \sim f_k\left(y; \theta_k, v/\varphi_k\right)$. Using one-hot coding, Z can be represent by random vector $\boldsymbol{Z}$:

$$\boldsymbol{Z} = (Z_1, \cdots, Z_K)^\top = (\mathbb{1}_{\{Z=1\}}, \ldots, \mathbb{1}_{\{Z=K\}})^\top. \tag{2.3}$$

Y can also be expressed as

$$Y = \sum_{k=1}^{K} Z_k Y_k^*, \tag{2.4}$$

where the pdf of Y_k^* is denoted by $f_k\left(y; \theta_k, v/\varphi_k\right)$. Therefore, under complete information $(Y, \boldsymbol{Z})$ is given by

$$\begin{aligned}
\ell_{(Y,Z)}(\boldsymbol{\theta}, \boldsymbol{p}) &= \log\left(\prod_{k=1}^{K} \left(p_k f_k\left(Y; \theta_k, v/\varphi_k\right)\right)^{Z_k}\right) \\
&= \log\left(\prod_{k=1}^{K} \left(p_k \exp\left\{\frac{Y\theta_k - \kappa_k\left(\theta_k\right)}{\varphi_k/v} + a_k\left(Y; v/\varphi_k\right)\right\}\right)^{Z_k}\right) \\
&= \sum_{k=1}^{K} Z_k \left(\log\left(p_k\right) + \frac{Y\theta_k - \kappa_k\left(\theta_k\right)}{\varphi_k/v} + a_k\left(Y; v/\varphi_k\right)\right).
\end{aligned} \tag{2.5}$$

In fact, the information of Z is missing, so we can use EM algorithm to estimate the $(\boldsymbol{\theta}, \boldsymbol{p})$ under incomplete information $\boldsymbol{Y}$. The EM algorithm for mixed distribution usually contains two steps: Expectation step and Maximization step.

Expectation step. Using Bayes' rule, we can calculate the posterior probability of $Z_k = 1$ given Y as

$$\mathbb{E}\left(Z_k \mid Y\right) = \mathbb{P}\left[Z_k = 1 \mid Y\right] = \frac{p_k f_k\left(Y; \theta_k, v/\varphi_k\right)}{\sum_{l=1}^{K} p_l f_l\left(Y; \theta_l, v/\varphi_l\right)},$$

which is used as an estimate of latent variable Z_k, denoted by $\hat{Z}_k$.

Expectation step. The joint log-likelihood function for the data set $(Y_i, \hat{Z}_{ki}, \boldsymbol{x}_i)$ is given by

$$\ell(\boldsymbol{\theta}, \boldsymbol{p}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{Z}_{ki} \left(\log\left(p_{ki}\right) + \frac{Y_i \theta_{ki} - \kappa_k\left(\theta_{ki}\right)}{\varphi_k / v_i} + a_k\left(Y_i; v_i/\varphi_k\right) \right). \tag{2.6}$$

Therefore, the MLE of $\boldsymbol{\theta}$ and $\boldsymbol{p}$ can be obtained by maximizing the log-likelihood function (2.6). Iterating the expectation step and maximization step leads to the MLE of $\boldsymbol{\theta}$ and $\boldsymbol{p}$.

3 Special Case: Zero-inflated Poisson model

A special example of mixed distribution is Zero-inflated Poisson distribution, which is given by

$$f_{\mathrm{ZIP}}(N; \lambda, \pi_0) = \begin{cases} \pi_0 + (1-\pi_0)e^{-\lambda} & \text{for} \quad N = 0 \\ (1-\pi_0)\frac{e^{-\lambda}\lambda^N}{N!} & \text{for} \quad N \in \mathbb{N}_+. \end{cases} \tag{3.1}$$

Given the data $(N_0, \boldsymbol{x}_i)_{i=1:n}$ the likelihood is

$$\begin{aligned} L(\beta, \gamma) = &\prod_{i=1}^{n} \left\{ \pi_0(\boldsymbol{x}_i; \beta) + [1 - \pi_0(\boldsymbol{x}_i; \beta)] e^{-\lambda(\boldsymbol{x}_i; \gamma)} \right\} \mathbb{1}_{\{N_i = 0\}} + \\ &[1 - \pi_0(\boldsymbol{x}_i; \beta)] \frac{e^{-\lambda(\boldsymbol{x}_i; \gamma)} \lambda(\boldsymbol{x}_i; \gamma)^{N_i}}{N_i!} \mathbb{1}_{\{N_i > 0\}}. \end{aligned} \tag{3.2}$$

The log-likelihood is then

$$\begin{aligned} l(\beta, \gamma) = &\sum_{i: N_i = 0} \log \left\{ \pi_0(\boldsymbol{x}_i; \beta) + [1 - \pi_0(\boldsymbol{x}_i; \beta)] e^{-\lambda(\boldsymbol{x}_i; \gamma)} \right\} + \\ &\sum_{i: N_i > 0} \log[1 - \pi_0(\boldsymbol{x}_i; \beta)] + N_i \log \lambda(\boldsymbol{x}_i; \gamma) - \lambda(\boldsymbol{x}_i; \gamma). \end{aligned} \tag{3.3}$$

The MLE of β and γ are difficult to obtain based on the log-likelihood (3.3).

Expectation-Maximization algorithm. The ZIP distribution can be interpreted as a mixture of two distributions, the Poisson distribution and a single point measure distribution in $N = 0$, with mixing probability π_0:

$$f_{\mathrm{ZIP}}(N; \lambda, \pi_0) = \pi_0 \mathbb{1}_{\{N=0\}} + (1-\pi_0)\frac{e^{-\lambda}\lambda^N}{N!}.$$

Therefore, we explore the Expectation-Maximization algorithm for the ZIP model. We introduce a latent Bernoulli variable Z with $\Pr(Z = 1) = \pi_0$, which indicates the component distribution, i.e., 1 for a single point measure distribution and 0 for the Poisson distribution. The joint probability mass function of (N, Z) can be written as

$$f_{\mathrm{ZIP}}(N, Z; \lambda, \pi_0) = \left(\pi_0 \mathbb{1}_{\{N=0\}}\right)^Z \left[(1-\pi_0)\frac{e^{-\lambda}\lambda^N}{N!} \right]^{1-Z}. \tag{3.4}$$

Expectation step. When $N > 0$, the latent variable is known as $Z = \hat{Z} = 0$. However, when $N = 0$, Z can be either 0 or 1. The conditional expectation of Z given $N = 0$ is

$$\mathbb{E}(Z|N=0) = \Pr(Z = 1|N = 0) = \frac{\Pr(Z = 1, N = 0)}{\Pr(N = 0)} = \frac{\pi_0}{\pi_0 + (1-\pi_0)e^{-\lambda}},$$

which is used as an estimate of latent variable Z, denoted by $\hat{Z}$.

Maximization step. The joint log-likelihood function for the data set $(N_i, \hat{Z}_i, \boldsymbol{x}_i)_{\{i:N_i=0\}} \cup (N_i, Z_i, \boldsymbol{x}_i)_{\{i:N_i>0\}}$ is given by

$$\begin{aligned} l(\beta,\gamma) = & \sum_{i:N_i=0} \hat{Z}_i \log \pi_0(\boldsymbol{x}_i;\beta) + (1-\hat{Z}_i)\log(1-\pi_0(\boldsymbol{x}_i;\beta)) - (1-\hat{Z}_i)\lambda(\boldsymbol{x}_i,\gamma) + \\ & \sum_{i:N_i>0} \log(1-\pi_0(\boldsymbol{x}_i,\beta)) + N_i \log\lambda(\boldsymbol{x}_i;\gamma) - \lambda(\boldsymbol{x}_i;\gamma) \\ = & \sum_{i=1}^{n} \hat{Z}_i \log \pi_0(\boldsymbol{x}_i;\beta) + \left(n - \sum_{i=1}^{n} \hat{Z}_i\right) \log(1-\pi_0(\boldsymbol{x}_i;\beta)) + \\ & \sum_{i=1}^{n} (1-\hat{Z}_i) \left[N_i \log\lambda(\boldsymbol{x}_i;\gamma) - \lambda(\boldsymbol{x}_i;\gamma)\right]. \end{aligned} \tag{3.5}$$

Therefore, the MLE of parameters β and γ can be obtained by maximizing the following two log-likelihood functions, respectively,

$$l_{\text{ZIP1}}(\beta) = \sum_{i=1}^{n} \hat{Z}_i \log \pi_0(\boldsymbol{x}_i;\beta) + \left(n - \sum_{i=1}^{n} \hat{Z}_i\right) \log(1-\pi_0(\boldsymbol{x}_i;\beta)) \tag{3.6}$$

and

$$l_{\text{ZIP2}}(\gamma) = \sum_{i=1}^{n} (1-\hat{Z}_i) \left(N_i \log\lambda(\boldsymbol{x}_i;\gamma) - \lambda(\boldsymbol{x}_i;\gamma)\right). \tag{3.7}$$

Iterating the expectation step and maximization step leads to the MLE of β and γ in the log-likelihood (3.3).

4 Gradient boosting for mixed EDFs

A cyclic algorithm proposed by Mayr et al. (2012) or a non-cyclic algorithm proposed by Thomas et al. (2018) can be implemented to iteratively estimate π_0 and λ based on the log-likelihood function (3.3). We propose a new algorithm which embeds expectation of latent variable into a non-cyclic boosting algorithm.

Essentially, the boosting algorithms are functional gradient decent techniques. The task is to estimate the function $F : \mathbb{R}^p \to \mathbb{R}$, minimizing an expected cost:

$$\mathbb{E}[C(Y, F(\boldsymbol{x}))], \qquad C : \mathbb{R}^2 \to \mathbb{R},$$

based on data

$$\mathcal{D} = \{(Y_1, \boldsymbol{x}_1, v_1), (Y_n, \boldsymbol{x}_2, v_2), \ldots, (Y_n, \boldsymbol{x}_n, v_n)\},$$

where $\boldsymbol{x}$ denotes a p-dimensional covariate. In statistical modeling, the cost function is usually chosen as the negative log-likelihood.

Let $F_1(\boldsymbol{x}), \ldots, F_{K-1}(\boldsymbol{x})$ be $K-1$ regression functions. The probability of $Z = k$ is given by

$$p_k(\boldsymbol{x}) = \Pr(Z_k = 1|\boldsymbol{x}) = \frac{\exp F_k(\boldsymbol{x})}{1 + \sum_{l=1}^{K-1} \exp F_l(\boldsymbol{x})}, \quad k = 1, \ldots, K-1 \tag{4.1}$$

and

$$p_K(\boldsymbol{x}) = \Pr(Z_K = 1|\boldsymbol{x}) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp F_l(\boldsymbol{x})}. \tag{4.2}$$

We use the negative log-likelihood function corresponds to the cost function

$$\begin{aligned} C_Z(\boldsymbol{Z}, \boldsymbol{F}(\boldsymbol{x})) &= -\sum_{k=1}^{K} Z_k \log p_k \\ &= -\sum_{k=1}^{K-1} Z_k F_k(\boldsymbol{x}) + \log[1 + \sum_{l=1}^{K-1} \exp F_l(\boldsymbol{x})]. \end{aligned} \tag{4.3}$$

The negative gradient of the two cost functions w.r.t. F_k is given by

$$U(Z_k, F_k(\boldsymbol{x})) = -\frac{\partial C_Z(Z_k, F_k(\boldsymbol{x}))}{\partial F_k(\boldsymbol{x})} = Z_k - \frac{\exp F_k(\boldsymbol{x})}{1 + \sum_{l=1}^{K-1} \exp F_l(\boldsymbol{x})}, \quad k = 1, \cdots, K-1 \tag{4.4}$$

Similarly, we can also use negative log-likelihood as cost function for Y_k^*:

$$C_{Y_k^*}(Y, Z_k, G_k(\boldsymbol{x})) = -\frac{v}{\varphi_k} Z_k \left[Y G_k(\boldsymbol{x}) - \kappa_k \left(G_k(\boldsymbol{x}) \right) \right] - a\left(Y; v/\varphi_k \right). \tag{4.5}$$

where $G_k(\boldsymbol{x}) = h_k(\mu_k(\boldsymbol{x}))$.

$$V_k(Y, Z_k, G_k(\boldsymbol{x})) = -\frac{\partial C_{Y_k^*}(Y, Z_k, G_k(\boldsymbol{x}))}{\partial G_k(\boldsymbol{x})} = \frac{v}{\varphi} Z_k \left[Y - \kappa_k' \left(G_k(\boldsymbol{x}) \right) \right]. \tag{4.6}$$

Algorithm 1. Mixed distribution boosting (cyclical)

Step 1 (initialization) Initialize $\hat{p}_1^{[0]}, \ldots, \hat{p}_K^{[0]}, \hat{\mu}_1^{[0]}, \ldots, \hat{\mu}_K^{[0]}, \hat{F}_1^{[0]}, \ldots, \hat{F}_{K-1}^{[0]}$, and $\hat{G}_1^{[0]}, \ldots, \hat{\mu}_K^{[0]}$:

$$\hat{p}_k^{[0]} = \frac{1}{K}, \quad \hat{\mu}_k^{[0]} = \frac{\sum_{i=1}^n Y_i}{\sum_{i=1}^n v_i}, \quad \hat{G}^{[0]} = h_k(\hat{\mu}_k^{[0]})$$

and $\hat{F}_1^{[0]}, \ldots, \hat{F}_{K-1}^{[0]}$ can be gained by solving the equation

$$\frac{1}{K} = \frac{1}{1 + (K-1)\exp F_k}$$

Set $m = 0$.

Step 2 (expectation of latent variable) Set $\hat{Z}_{ki}^{[m]}$ as

$$\hat{Z}_{ki}^{[m]} = \frac{\hat{p}_k^{[m]}(\boldsymbol{x}_i) f_k\left(Y_i; \hat{\mu}_k^{[m-1]}(\boldsymbol{x}_i), v_i/\varphi_k \right)}{\sum_{l=1}^K \hat{p}_k^{[m]}(\boldsymbol{x}_i) f_k\left(Y_i; \hat{\mu}_k^{[m-1]}(\boldsymbol{x}_i), v_i/\varphi_k \right)},$$

Step 3 (projection of gradient to learner). The follwoing two base learners are fitted independently.

Tree 1 ($K-1$ trees) Compute the negative gradient vector $(u_{k1}^{[m]}, \ldots, u_{kn}^{[m]})^\top, k = 1, \ldots, K-1$ in which

$$u_{ki}^{[m]} = U(\hat{Z}_i, \hat{F}_k^{[m]}(\boldsymbol{x}_i)).$$

Then fit the gradient vector to covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ by a K-terminal node regression trees $\hat{f}_k^{[m+1]}\left(\boldsymbol{x}; R^{[m]}, \bar{u}_k^{[m]} \right)$ with L_2 loss

Tree 2 (K trees) Compute the negative gradient vector $(v_1^{[m]}, \ldots, v_n^{[m]})^\top$ in which

$$v_{ki}^{[m]} = V(Y_i, \hat{Z}_{ki}, \hat{G}_k^{[m]}(\boldsymbol{x}_i)), \quad k = 1, \ldots, K$$

Then fit the gradient vector to covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ by a K-terminal node regression trees $\hat{g}_k^{[m+1]}\left(\boldsymbol{x}; S^{[m]}, \bar{v}_k^{[m]}\right)$ with L_2 loss.

Step 4 (line search) Conduct the following two independent one-dimensional line searches for the best step sizes (expansion coefficents).

$$\hat{w}_{1k}^{[m+1]} = \arg\min_w \sum_{i=1}^n C_Z(\hat{Z}_i, \hat{F}_k^{[m]}(\boldsymbol{x}_i) + w\hat{f}_k^{[m+1]}(\boldsymbol{x}_i)), \quad k = 1, \cdots, K-1$$

$$\hat{w}_{2k}^{[m+1]} = \arg\min_w \sum_{i=1}^n C_{Y_k^*}(Y_i, \hat{Z}_i, \hat{G}_k^{[m]}(\boldsymbol{x}_i) + w\hat{g}_k^{[m+1]}(\boldsymbol{x}_i)), \quad k = 1, \cdots, K$$

Compute the proposed updates

$$\hat{F}_k^{[m+1]}(\boldsymbol{x}_i) = \hat{F}_k^{[m]}(\boldsymbol{x}_i) + \hat{w}_{1k}^{[m+1]}\hat{f}_k^{[m+1]}(\boldsymbol{x}_i),$$

$$\hat{G}_k^{[m+1]}(\boldsymbol{x}_i) = \hat{G}_k^{[m]}(\boldsymbol{x}_i) + \hat{w}_{2k}^{[m+1]}\hat{g}_k^{[m+1]}(\boldsymbol{x}_i),$$

$$\hat{p}_k^{[m+1]}(\boldsymbol{x}_i) = \frac{\exp \hat{F}_k^{[m]}(\boldsymbol{x})}{1 + \sum_{l=1}^{K-1} \exp \hat{F}_k^{[m]}(\boldsymbol{x})}, \quad k = 1, \ldots, K-1,$$

$$\hat{p}_K^{[m+1]}(\boldsymbol{x}_i) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp \hat{F}_k^{[m]}(\boldsymbol{x})},$$

$$\hat{\mu}_k^{[m+1]}(\boldsymbol{x}_i) = h_k(\hat{G}_k^{[m+1]}(\boldsymbol{x}_i)).$$

Step 5 (iteration). Increase m by 1, and repeat steps 2-4.

4.1 Special Case: zero-inflated Poisson boosting

A cyclic algorithm proposed by Mayr et al. (2012) or a non-cyclic algorithm proposed by Thomas et al. (2018) can be implemented to iteratively estimate π_0 and λ based on the log-likelihood function (3.3). We propose a new algorithm which embeds expectation of latent variable into a non-cyclic boosting algorithm.

Similar to zero-truncated Poisson model, we assume the logit link function for π_0 and the logarithm link function for λ,

$$\log \frac{\pi_0}{1 - \pi_0} = F \quad \text{and} \quad \log \lambda = G.$$

The two cost functions are given by

$$\begin{aligned} C_{ZIP1}(Z, F(\boldsymbol{x})) &= -Z \log \left(\frac{\exp F(\boldsymbol{x})}{1 + \exp F(\boldsymbol{x})} \right) - (1 - Z) \log \left(1 - \frac{\exp F(\boldsymbol{x})}{1 + \exp F(\boldsymbol{x})} \right) \\ &= -ZF(\boldsymbol{x}) + \log \left(1 + \exp F(\boldsymbol{x}) \right) \end{aligned}$$

and

$$C_{ZIP2}(N, Z, G(\boldsymbol{x})) = (1 - Z) \left(-NG(\boldsymbol{x}) + \exp G(\boldsymbol{x}) \right).$$

The negative gradient of the two cost functions w.r.t. F or G are given by

$$U(Z, F(\boldsymbol{x})) = -\frac{\partial C_{\text{ZIP1}}(Z, F(\boldsymbol{x}))}{\partial F(\boldsymbol{x})} = Z - \frac{\exp F(\boldsymbol{x})}{1 + \exp F(\boldsymbol{x})} \tag{4.7}$$

and

$$V(N, Z, G(\boldsymbol{x})) = -\frac{\partial C_{\text{ZIP2}}(N, Z, G(\boldsymbol{x}))}{\partial G(\boldsymbol{x})} = (1 - Z)(N - \exp G(\boldsymbol{x})).$$

Note that we do a slight abuse of notations F, G, U, V.

Algorithm 2. Zero-inflated Poisson boosting.

Step 1 (initialization) Initialize $\hat{\pi}_0^{[0]}, \hat{\lambda}^{[0]}, \hat{F}^{[0]}$ and $\hat{G}^{[0]}$:

$$\hat{\pi}_0^{[0]} = \frac{\sum_{i=1}^n \mathbb{1}_{\{N_i=0\}}}{n}, \quad \hat{\lambda}^{[0]} = \frac{\sum_{i=1}^n N_i}{n}, \quad \hat{F}^{[0]} = \log \frac{\hat{\pi}_0^{[0]}}{1 - \hat{\pi}_0^{[0]}}, \quad \hat{G}^{[0]} = \log \hat{\lambda}^{[0]}.$$

Set $m = 0$.

Step 2 (expectation of latent variable) For $N_i > 0$, the latent variable is set as $\hat{Z}_i^{[m]} = 0$. For $N_i = 0$, set $\hat{Z}_i^{[m]}$ as

$$\hat{Z}_i^{[m]} = \frac{\hat{\pi}_0^{[m]}(\boldsymbol{x}_i)}{\hat{\pi}_0^{[m]}(\boldsymbol{x}_i) + (1 - \hat{\pi}_0^{[m]}(\boldsymbol{x}_i))e^{-\hat{\lambda}(\boldsymbol{x}_i)}}.$$

Update data $(N_i, \hat{Z}_i^{[m]}, \boldsymbol{x}_i)_{i=1:n}$

Step 3 (projection of gradient to learner). The follwoing two base learners are fitted independently.

Tree 1 Compute the negative gradient vector $(u_1^{[m]}, \ldots, u_n^{[m]})^\top$ in which

$$u_i^{[m]} = U(\hat{Z}_i, \hat{F}^{[m]}(\boldsymbol{x}_i)).$$

Then fit the gradient vector to covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ by a K-terminal node regression tree $\hat{f}^{[m+1]}\left(\boldsymbol{x}; R^{[m]}, \bar{u}^{[m]}\right)$ with L_2 loss

Tree 2 Compute the negative gradient vector $(v_1^{[m]}, \ldots, v_n^{[m]})^\top$ in which

$$v_i^{[m]} = V(N_i, \hat{Z}_i, \hat{G}^{[m]}(\boldsymbol{x}_i)).$$

Then fit the gradient vector to covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ by a K-terminal node regression tree $\hat{g}^{[m+1]}\left(\boldsymbol{x}; S^{[m]}, \bar{v}^{[m]}\right)$ with L_2 loss.

Step 4 (line search) Conduct the following two independent one-dimensional line searches for the best step sizes (expansion coefficents).

$$\hat{w}_1^{[m+1]} = \arg\min_w \sum_{i=1}^n C_{\text{ZIP1}}(\hat{Z}_i, \hat{F}^{[m]}(\boldsymbol{x}_i) + w\hat{f}_{m+1}(\boldsymbol{x}_i)).$$

$$\hat{w}_2^{[m+1]} = \arg\min_w \sum_{i=1}^n C_{\text{ZIP2}}(N_i, \hat{Z}_i, \hat{G}^{[m]}(\boldsymbol{x}_i) + w\hat{g}_{m+1}(\boldsymbol{x}_i)).$$

Compute the proposed updates

$$\hat{F}^*(\boldsymbol{x}_i) = \hat{F}^{[m]}(\boldsymbol{x}_i) + \hat{w}_1^{[m+1]} \hat{f}_{m+1}(\boldsymbol{x}_i),$$

$$\hat{G}^*(\boldsymbol{x}_i) = \hat{G}^{[m]}(\boldsymbol{x}_i) + \hat{w}_2^{[m+1]} \hat{g}_{m+1}(\boldsymbol{x}_i),$$

$$\hat{\pi}^*(\boldsymbol{x}_i) = \frac{\exp \hat{F}^*(\boldsymbol{x}_i)}{1 + \exp \hat{F}^*(\boldsymbol{x}_i)},$$

$$\hat{\lambda}^*(\boldsymbol{x}_i) = \exp \hat{G}^*(\boldsymbol{x}_i)$$

Step 5 (selection of base learner) Calculate the decrease of negative log-likelihood (3.3) due to either $\hat{F}^*(\cdot)$ or $\hat{G}^*(\cdot)$:

$$\Delta_1 = -l(\hat{\pi}^{[m]}, \hat{\lambda}^{[m]}) + l(\hat{\pi}^*, \hat{\lambda}^{[m]}),$$

$$\Delta_2 = -l(\hat{\pi}^{[m]}, \hat{\lambda}^{[m]}) + l(\hat{\pi}^{[m]}, \hat{\lambda}^*).$$

If $\Delta_1 > \Delta_2$, accept the proposed $\hat{F}^*(\cdot), \hat{\pi}^*(\cdot)$ but reject the proposed $\hat{G}^*(\cdot), \hat{\lambda}^*(\cdot)$. Update $\hat{F}^{[m+1]}(\boldsymbol{x}_i) = \hat{F}^*(\boldsymbol{x}_i), \hat{\pi}^{[m+1]}(\boldsymbol{x}_i) = \hat{\pi}^*(\boldsymbol{x}_i), \hat{G}^{[m+1]}(\boldsymbol{x}_i) = \hat{G}^{[m]}(\boldsymbol{x}_i), \hat{\lambda}^{[m+1]}(\boldsymbol{x}_i) = \hat{\lambda}^{[m]}(\boldsymbol{x}_i)$.

If $\Delta_1 < \Delta_2$, accept the proposed $\hat{G}^*(\cdot), \hat{\lambda}^*(\cdot)$ but reject the proposed $\hat{F}^*(\cdot), \hat{\pi}^*(\cdot)$. Update $\hat{F}^{[m+1]}(\boldsymbol{x}_i) = \hat{F}^{[m]}(\boldsymbol{x}_i), \hat{\pi}^{[m+1]}(\boldsymbol{x}_i) = \hat{\pi}^{[m]}(\boldsymbol{x}_i), \hat{G}^{[m+1]}(\boldsymbol{x}_i) = \hat{G}^*(\boldsymbol{x}_i), \hat{\lambda}^{[m+1]}(\boldsymbol{x}_i) = \hat{\lambda}^*(\boldsymbol{x}_i)$.

Step 6 (iteration). Increase m by 1, and repeat steps 2-5.

4.2 Special Case: Gaussian mixture model

We assume that Y_i is mixed Gaussian r.v. with pdf $f_Y(y_i; \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, p)$ i.e.

$$Y_i \sim p \left[\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp \left\{ -\frac{[y_i - \mu_1(\boldsymbol{x}_i)]^2}{2\sigma_1^2} \right\} \right] + (1-p) \left[\frac{1}{\sqrt{2\pi\sigma_2^2}} \exp \left\{ -\frac{[y_i - \mu_2(\boldsymbol{x}_i)]^2}{2\sigma_2^2} \right\} \right],$$

where $\mu_1(\cdot)$ and $\mu_2(\cdot)$ are two regression function. Now we propose a cyclic boosting algorithm for Gaussian mixture model. For the first Gaussian distribution, we use the negative log-likelihood for cost function, that is

$$C_1(Z, Y, \mu_1(\boldsymbol{x}), \sigma_1^2) = Z \left[\frac{1}{2} \log(2\pi\sigma_1^2) + \frac{1}{2\sigma_1^2} (Y - \mu_1(\boldsymbol{x}))^2 \right].$$

The negative gradient of C_1 is given by

$$V_1(Z, Y, \mu_1(\boldsymbol{x}), \sigma_1^2) = -\frac{Z}{\sigma_1^2} (Y - \mu_1(\boldsymbol{x})).$$

Also, the cost function with respect to the second Gaussian distribution is given by

$$C_2(Z, Y, \mu_2(\boldsymbol{x}), \sigma_2^2) = (1-Z) \left[\frac{1}{2} \log(2\pi\sigma_2^2) + \frac{1}{2\sigma_2^2} (Y - \mu_2(\boldsymbol{x}))^2 \right].$$

The negative gradient of C_2 is given by

$$V_2(Z, Y, \mu_2(\boldsymbol{x}), \sigma_2^2) = -\frac{1-Z}{\sigma_2^2} (Y - \mu_2(\boldsymbol{x})).$$

Algorithm 3. Mixed Gaussian boosting.

Step 1 (initialization) Initialize $\hat{p}^{[0]}, \hat{\mu}_1^{[0]}, \hat{\mu}_2^{[0]}, (\hat{\sigma}_1^2)^{[0]}, (\hat{\sigma}_2^2)^{[0]}$:

$$\hat{p}^{[0]} = \frac{1}{2}, \quad \hat{\mu}_1^{[0]} = \hat{\mu}_2^{[0]} = \bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}, \quad (\hat{\sigma}_1^2)^{[0]} = (\hat{\sigma}_2^2)^{[0]} = \frac{1}{n}\sum_{i=1}^n (Y_i - \bar{Y})^2.$$

Set $m = 0$.

Step 2 (expectation of latent variable) Set $\hat{Z}_i^{[m]}$ as

$$\hat{Z}_i^{[m]} = \frac{\frac{\hat{p}^{[m]}}{\sqrt{2\pi(\hat{\sigma}_1^2)^{[m]}}} \exp\left\{-\frac{1}{2(\hat{\sigma}_1^2)^{[m]}}[Y_i - \hat{\mu}_1^{[m]}(\boldsymbol{x}_i)]^2\right\}}{\frac{\hat{p}^{[m]}}{\sqrt{2\pi(\hat{\sigma}_1^2)^{[m]}}} \exp\left\{-\frac{1}{2(\hat{\sigma}_1^2)^{[m]}}[Y_i - \hat{\mu}_1^{[m]}(\boldsymbol{x}_i)]^2\right\} + \frac{1-\hat{p}^{[m]}}{\sqrt{2\pi(\hat{\sigma}_2^2)^{[m]}}} \exp\left\{-\frac{1}{2(\hat{\sigma}_2^2)^{[m]}}[Y_i - \hat{\mu}_2^{[m]}(\boldsymbol{x}_i)]^2\right\}}.$$

Update data $(Y_i, \hat{Z}_i^{[m]}, \boldsymbol{x}_i)_{i=1:n}$

Step 3 (projection of gradient to learner). The follwoing two base learners are fitted independently.

Tree 1 Compute the negative gradient vector $(v_{11}^{[m]}, \ldots, v_{1n}^{[m]})^\top$ in which

$$v_{1i}^{[m]} = V_1(Y_i, \hat{Z}_i^{[m]}, \hat{\mu}_1^{[m]}(\boldsymbol{x}_i), (\hat{\sigma}_1^2)^{[m]}).$$

Then fit the gradient vector to covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ by a K-terminal node regression tree $\hat{g}_1^{[m+1]}\left(\boldsymbol{x}; S_1^{[m]}, \bar{v}_1^{[m]}\right)$ with L_2 loss.

Tree 2 Compute the negative gradient vector $(v_{21}^{[m]}, \ldots, v_{2n}^{[m]})^\top$ in which

$$v_{2i}^{[m]} = V_1(Y_i, \hat{Z}_i^{[m]}, \hat{\mu}_2^{[m]}(\boldsymbol{x}_i), (\hat{\sigma}_2^2)^{[m]}).$$

Then fit the gradient vector to covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ by a K-terminal node regression tree $\hat{g}_2^{[m+1]}\left(\boldsymbol{x}; S_2^{[m]}, \bar{v}_2^{[m]}\right)$ with L_2 loss.

Step 4 (line search) Conduct the following two independent one-dimensional line searches for the best step sizes (expansion coefficents).

$$\hat{w}_1^{[m+1]} = \arg\min_w \sum_{i=1}^n C_1(Y_i, \hat{Z}_i, \hat{\mu}_1^{[m]}(\boldsymbol{x}_i) + w\hat{g}_1^{[m+1]}(\boldsymbol{x}_i), (\sigma_1^2)^{[m]})$$

$$\hat{w}_2^{[m+1]} = \arg\min_w \sum_{i=1}^n C_2(Y_i, \hat{Z}_i, \hat{\mu}_2^{[m]}(\boldsymbol{x}_i) + w\hat{g}_2^{[m+1]}(\boldsymbol{x}_i), (\sigma_2^2)^{[m]})$$

Compute the proposed updates

$$\hat{\mu}_1^{[m+1]}(\boldsymbol{x}_i) = \hat{\mu}_1^{[m]}(\boldsymbol{x}_i) + \hat{w}_1^{[m+1]}\hat{g}_1^{[m+1]}(\boldsymbol{x}_i),$$
$$\hat{\mu}_2^{[m+1]}(\boldsymbol{x}_i) = \hat{\mu}_2^{[m]}(\boldsymbol{x}_i) + \hat{w}_2^{[m+1]}\hat{g}_2^{[m+1]}(\boldsymbol{x}_i),$$
$$\hat{p}^{[m+1]}(\boldsymbol{x}_i) = \frac{\sum_{i=1}^n \hat{Z}_i^{[m]}}{n},$$
$$(\hat{\sigma}_1^2)^{[m+1]} = \frac{\sum_{i=1}^n Z_i^{[m]}[Y_i - \hat{\mu}_1^{[m+1]}(\boldsymbol{x}_i)]^2}{\sum_{i=1}^n Z_i^{[m]}},$$
$$(\hat{\sigma}_2^2)^{[m+1]} = \frac{\sum_{i=1}^n (1 - Z_i^{[m]})[Y_i - \hat{\mu}_2^{[m+1]}(\boldsymbol{x}_i)]^2}{\sum_{i=1}^n (1 - Z_i^{[m]})}.$$

Step 5 (iteration). Increase m by 1, and repeat steps 2-4.

References

Mayr, A., Fenske, N., Hofner, B., Kneib, T. and Schmid, M. (2012). Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **61**(3): 403–427.

Thomas, J., Mayr, A., Bischl, B., Schmid, M., Smith, A. and Hofner, B. (2018). Gradient boosting for distributional regression: faster tuning and improved variable selection via noncyclical updates, *Statistics and Computing* **28**(3): 673–687.

Wüthrich, M. V. and Merz, M. (2021). Statistical foundations of actuarial learning and its applications, *SSRN* (3822407).