Cyclical Gradient Boosting Machines

for multidimensional parameter estimation

Henning Zakrisson
based on joint work with Łukasz Delong and Mathias Lindholm

Department of Mathematics
Stockholm University

June 9, 2023

Disclaimer

Disclaimer:

The following slides present a simplified version of the algorithm. The notation is not consistent with the paper and is only meant to give an intuition of the algorithm.

Gradient Boosting Machines

Gradient Boosting Machines

The goal is to find parameter function

$$\theta(x) : \mathbb{R}^p \to \mathbb{R}$$

that minimizes some loss function

$$L\left((y_i, \theta(x_i))_{i=1}^n\right)$$

on the training data set

$$(y_i, x_i)_{i=1}^n$$

Gradient Boosting Machines

Algorithm: Gradient Boosting Machine

- Initialize $\theta^{(0)}(x) \in \mathbb{R}$
- For $k = 1, \ldots, \kappa$
  - Calculate the point-wise negative derivatives

$$g_i = -\frac{\partial L\left(y_i, \theta(x_i)\right)}{\partial \theta(x_i)}\bigg|_{\theta=\theta^{(k-1)}}$$

  - Fit a regression tree h to the gradients

$$\gamma^{(k)} = \arg\min_{\gamma} \sum_{i=1}^{n} \left(g_i - h(x_i; \gamma)\right)^2$$

  - Update parameter function

$$\theta^{(k)}(x) = \theta^{(k-1)}(x) + \epsilon \cdot h(x; \gamma^{(k)})$$

Early stopping

Gradient Boosting Machines are prone to overfitting.
To avoid this, we can use early stopping, i.e. adjust hyper-parameter κ.
Split data set into

- Training data set: $(y_i, x_i)_{i=1}^{m}$
- Validation data set: $(y_i, x_i)_{i=m+1}^{n}$

Run the algorithm for $k = 1, 2, \ldots$ and choose

$$\kappa = \arg\min_k L\left(\left(y_i, \theta^{(k)}(x_i)\right)_{i=m+1}^{n}\right)$$

Early stopping (example)

Sample $(y_i, x_i)_{i=1}^{n}$ from:

$$X_i \sim \mathcal{N}(0, I), \qquad Y_i \sim \mathcal{N}(\mu_i(x_i), \sigma^2)$$

with parameter function

$$\mu_i(x_i) = x_{i1} + 10 \cdot \mathbb{1}_{\{x_{i2} > 0\}}, \qquad \sigma^2 = 1$$

Create training data set $(y_i, x_i)_{i=1}^{\frac{n}{2}}$ and validation data set $(y_i, x_i)_{i=\frac{n}{2}+1}^{n}$.
Run GBM with early stopping, $n = 10,000$, $\epsilon = 0.1$.

Early stopping (example)

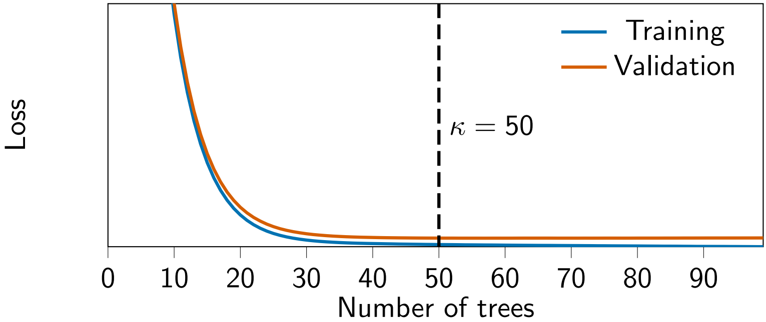

Figure: Early stopping for a GBM

Early stopping (example)

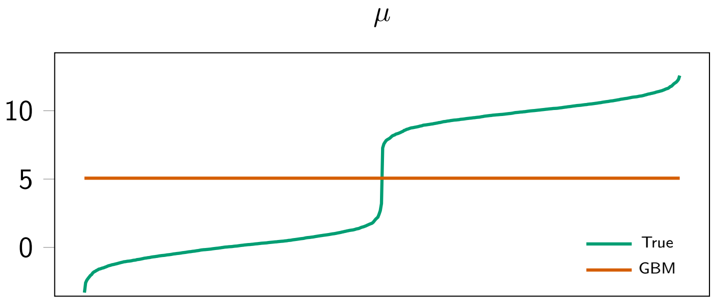

Figure: Parameter estimates, $\kappa = 0$

Early stopping (example)

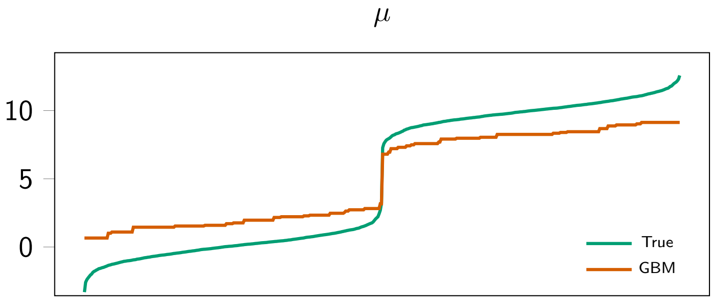

Figure: Parameter estimates, $\kappa = 10$

Early stopping (example)

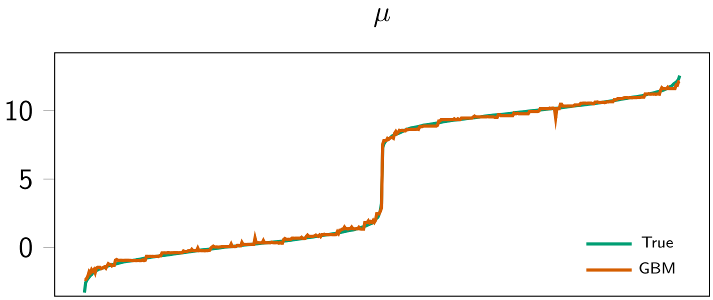

Figure: Parameter estimates, $\kappa = 50$

Cyclical Gradient Boosting Machines

Cyclical Gradient Boosting Machines

The goal is to find parameter function

$$\theta(x) : \mathbb{R}^p \to \mathbb{R}^d$$

for $d \geq 1$ that minimizes some loss function

$$L\left((y_i, \theta(x_i))_{i=1}^n\right)$$

on the training data set

$$(y_i, x_i)_{i=1}^n$$

Cyclical Gradient Boosting Machines

Algorithm: Cyclical Gradient Boosting Machine

- Initialize $\theta^{(0)}(x) = \widehat{\theta}_{\text{MLE}}$
- For $k = 1, \ldots, \kappa$
  - For $j = 1, 2, \ldots, d$
    - Calculate the point-wise negative partial derivatives

$$g_{ij} = -\frac{\partial L\left(y_i, \theta(x_i)\right)}{\partial \theta_j(x_i)}\bigg|_{\theta=\theta^{(k-1)}}$$

    - Fit a regression tree to the gradients

$$\gamma_j^{(k)} = \arg\min_{\gamma} \sum_{i=1}^{n} \left(g_{ij} - h(x_i; \gamma)\right)^2$$

    - Update parameter function

$$\theta_j^{(k)}(x) = \theta_j^{(k-1)}(x) + \epsilon \cdot h(x; \gamma_j^{(k)})$$

Cyclical Gradient Boosting Machines

- Using the univariate early stopping scheme can be problematic!
- The complexity of the parameter function can differ over the different dimensions.
- This might lead to dimension-wise over- or underfitting.

Strategy: individual stopping times for each dimension.
For every boosting step k and every dimension j, calculate the loss contribution ΔL_{jk}. Then, choose

$$\kappa_j = \arg\min_k \{k : \Delta L_{jk} > 0\}$$

Simulated example

Sample $(y_i, x_i)_{i=1}^{n}$ from:

$$X_i \sim \mathcal{N}(0, I), \qquad Y_i \sim \mathcal{N}(\mu_i(x_i), \sigma(x_i)^2)$$

with parameter function

$$\mu_i(x_i) = x_{i1} + 10 \cdot \mathbb{1}_{\{x_{i2} > 0\}}$$

$$\log \sigma(x_i) = 3 - 2 \cdot \mathbb{1}_{\{x_{i1} > -0.2\}}$$

Create training data set $(y_i, x_i)_{i=1}^{\frac{n}{2}}$ and validation data set $(y_i, x_i)_{i=\frac{n}{2}+1}^{n}$.
Run CGBM with (individual) early stopping, $n = 100,000$, $\epsilon = 0.1$.

Simulated example

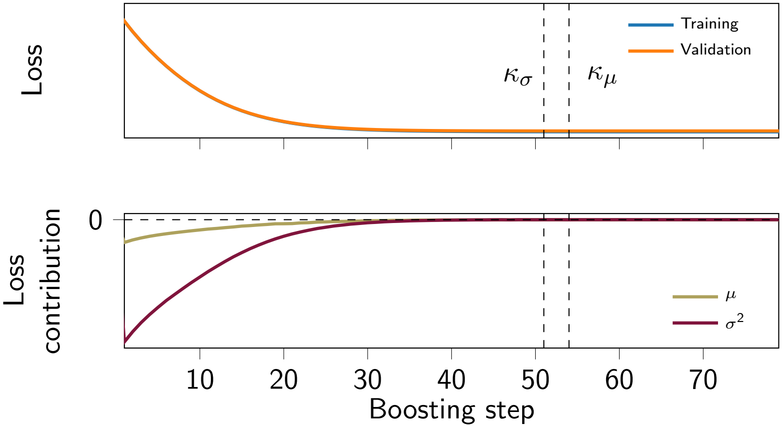

Figure: Early stopping for a CGBM using individual stopping times

Simulated example

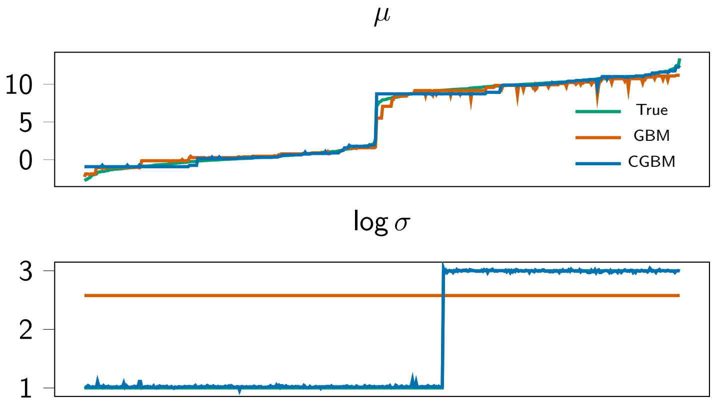

Figure: Parameter estimates

Real data example

The `freMTPL2` data set contains $678,013$ observations of French motor third-party liability claims, of which $34,060$ have at least one claim. The following covariates are available:

Feature	Description	Type
Brand	Brand of car	Categorical (7)
Gas	Gas used by car	Categorical (2)
Density	Population density in car-owners city	Continuous
Area	Area of car	Categorical
Region	Region of car	Categorical
BonusMalus	Bonus/Malus level of driver	Continuous
Power	Power level of car	Ordinal (12)
Vehicle age	Age of the car in years	Continuous
Driver age	Age of driver in years	Continuous

Table: Features used in the real data example.

Real data example

Assume

$$N_i | X_i \sim \text{NegBin}\left(w_i \mu(X_i), w_i \theta(X_i)\right)$$
$$Y_i | X_i, N_i \sim \text{Gamma}\left(N_i m(X_i), \phi(X_i)/N_i\right)$$

where, for contract i,

- N_i is the number of claims
- w_i is the duration of the contract
- Y_i is the total claim amount
- X_i is the vector of covariates

using a mean-dispersion parametrization for the both distribution such that

$$\mathbb{E}\left[N_i | X_i\right] = w_i \mu(X_i), \qquad \text{Var}\left[N_i | X_i\right] = w_i \mu(X_i) \left(1 + \frac{\mu(X_i)}{\theta(X_i)}\right)$$
$$\mathbb{E}\left[Y_i | N_i, X_i\right] = N_i m(X_i), \qquad \text{Var}\left[Y_i | N_i, X_i\right] = N_i m(X_i)^2 \phi(X_i)$$

Real data example

The CGBM produces a slightly lower loss for both data sets.

		Intercept	GBM	CGBM
Claim counts	Train	0.21	0.20	0.20
	Test	0.24	0.24	0.20
Claim amounts	Train	1.25	1.25	1.20
	Test	1.24	1.24	1.20

Table: Average negative log-likelihood for the `freMTPL2` data set.

Real data example

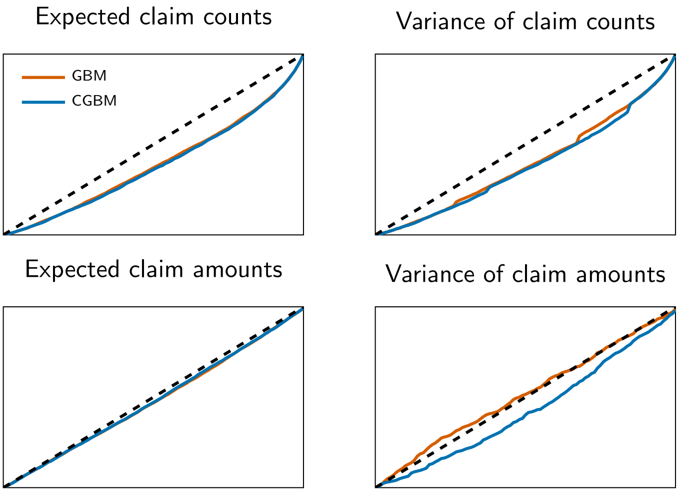

Figure: Concentration curves for the `freMTPL2` data set.

Summary

- Cyclical Gradient Boosting extends Gradient Boosting to the multi-parametric setting.
- A hyperparameter tuning procedure is proposed.
- The algorithm can easily be extended to similar algorithms such as XGBoost.

References

- J. H. Friedman, *Greedy function approximation: a gradient boosting machine*, *Annals of statistics*, pp. 1189–1232, 2001.
- Ł. Delong, M. Lindholm, and H. Zakrisson, *A Note on Multi-Parametric Gradient Boosting Machines with Non-Life Insurance Applications*, *Available at SSRN*, 2023.

Thank you

Thank you for your attention!
Questions?