CEREMADE
UMR CNRS 7534

Dauphine | PSL
UNIVERSITÉ PARIS

AN EXPLICIT SPLIT POINT PROCEDURE IN MODEL-BASED TREES ALLOWING FOR A QUICK FITTING OF GLM TREES AND GLM FORESTS

Quentin Guibert

CEREMADE, Université Paris-Dauphine
Email: guibert@ceremade.dauphine.fr

MLISTRAL Conference
27 September 2022, Marseille, France

Joint work with Christophe Dutang

Outlines

Outlines

Binary trees

Recursive binary partitioning: technique for building decision trees by separating a dataset into different homogeneous subgroups according to partitioning variables.

These models have been widely used in supervised learning for regression and classification for more than 50 years [Loh14].

- Most binary trees comprise two or three steps:
  1. recursively splitting the dataset by selecting the best split until reaching a stopping criterion,
  2. fitting an intercept-only model at each terminal node.
  3. then, the tree may be pruned.
- CART [Bre+84]: the most used, efficient and easy to interpret.
- Drawbacks: selection bias problem induced by an exhaustive search over all possible splits simultaneously, see [Bre+84].
- Possible improvements: selection based on statistical tests, see, FACT [LV88], QUEST [LS97], CRUISE [KL01] or CTREE [HHZ06] algorithms.

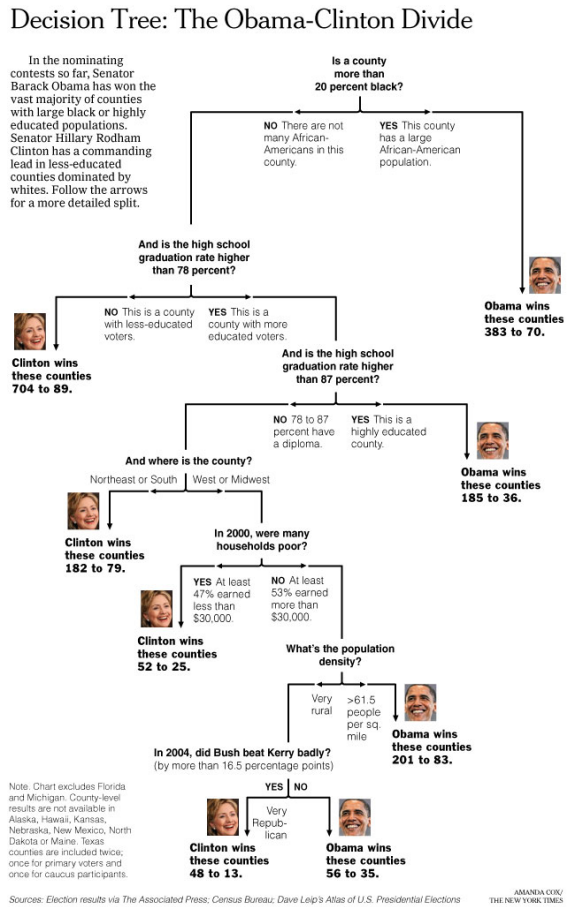


FIGURE 1: Example of decision tree

From trees to forest

For all tree methods, single trees suffer from an instability issue:

$\Rightarrow$ the resulting tree can be significantly affected by small changes in the training data.

They may be less competitive than other approaches in machine learning, like

- neural networks [Law94] or
- support vector machines [CV95] in terms of prediction.

Predictions can be improved by introducing ensemble tree methods based on:

- bagging [Bre96],
- random forest [Bre01]
- or boosting [Fri02],

but this is done at the expense of interpretability.

In this literature, the overwhelming majority of approaches are based on the CART algorithm, although variable selection may be biased.

Aims of the study

MOdel-Based recursive partitioning (MOB)

- Tree algorithm based on a set of partitioning variables and a local parametric model (ex : GLM) fitted on explanatory variables.
- Predictions given by a statistical model adapted to each node
- Computation time significantly longer than simple tree algorithms.
- Tricky to use with ensemble tree methods.

Main aims

- Reduce the computation time using explicit fitted GLM likelihood as an objective function.
- Introduce closed-form estimators for GLM-type trees for any link function in the case of categorical explanatory variables.
- Study the gain in computation speed.
- Explore the interest of a GLM forest algorithm.

Outlines

Generalized linear models (GLM)

[NW72] unify different regression models through the exponential family.

The likelihood L of the response variable Y_i verifies

$$\log L(\boldsymbol{\theta} \mid y_i) = \frac{\lambda_i(\boldsymbol{\theta}) y_i - b\left(\lambda_i(\boldsymbol{\theta})\right)}{a(\phi)} + c(y_i, \phi), \quad y_i \in \mathbb{Y} \subset \mathbb{R}, \tag{1}$$

and $-\infty$ if $y_i \notin \mathbb{Y}$, where $a : \mathbb{R} \to \mathbb{R}$, $b : \Lambda \to \mathbb{R}$ and $c : \mathbb{Y} \times \mathbb{R} \to \mathbb{R}$ are known real-valued measurable functions and ϕ is the dispersion parameter

GLM are defined by assuming that

- $(Y_i)_i$ are independent random variables,
- $Y_i \sim \mathcal{F}_{exp}(\lambda_i, \phi, a, b, c)$,
- the expectation $E(Y_i)$ and variables $\boldsymbol{x}_i$ are linked

$$g(E(Y_i)) = g(b'(\lambda_i(\boldsymbol{\theta}))) = \langle \boldsymbol{x}_i, \boldsymbol{\theta} \rangle = \eta_i, \quad \text{for all } \boldsymbol{\theta} \in \Theta, \tag{2}$$

  where η_i are the linear predictors.

USUAL DISTRIBUTIONS AND FITTING PROCEDURE

Distribution	$\mathbb{Y}$	λ	ϕ	$a(x)$	$b(x)$	$c(x,\phi)$
Bernoulli $\mathcal{B}(p)$	$\{0,1\}$	$\log(\frac{p}{1-p})$	1	x	$\log(1+e^x)$	0
Poisson $\mathcal{P}(\mu)$	$\mathbb{N}$	$\log(\mu)$	1	x	e^x	$-\log(x!)$
Gaussian $\mathcal{N}(\mu,\sigma^2)$	$\mathbb{R}$	μ	σ^2	x		$-\frac{1}{2}\log(2\pi\phi)$
Gamma $\mathcal{G}(\nu,\mu)$	$]0,+\infty[$	$\frac{-1}{\mu}$	$1/\nu$	x	$-\log(-x)$	$\frac{\log(x/\phi)}{\phi} - \log(x)$ $-\log\Gamma(\frac{1}{\phi})$
Inv. Gauss. $\mathcal{IG}(\mu,\sigma^2)$	$]0,+\infty[$	$-1/(2\mu^2)$	$1/\sigma^2$	x	$-\sqrt{-2x}$	$-\frac{1}{2}\log(2\pi\phi x^3)$ $-1/(2\phi x)$

TABLE 1: Usual distributions in the exponential family

- GLM with the gaussian distribution is a linear model and can be fitted by using closed-form formulas.
- In general, GLM are fitted using a numerical procedure which solves the score equations based on an Iteratively re-Weighted Least Square (IWLS) algorithm.
- But, in some situations closed form solutions exists $\Rightarrow$ Potential gain in terms of computation efficiency.

Outlines

Model-based (MOB) partitioning tree – goals and notation

[ZHH08] introduce model-based trees:

- by integrating a parametric model (e.g. GLM or survival regression) fitted at each leaf of a tree
- based on least squares, maximum likelihood or more broadly M-estimation approaches
- selecting variables on M-fluctuation test
- generalizing score-based tests, statistics based on LM statistics.

Notation:

- partitioning variables $\boldsymbol{z}_i = (z_{i,1}, \ldots, x_{i,q}) \in \mathbb{R}^q$,
- explanatory variables $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,p}) \in \mathbb{R}^p$,
- response variable $\boldsymbol{y}$

GLM Trees – overall principle

The GLM-based tree algorithm [RZ13] consists of splitting the dataset recursively based on a set of partitioning variables and of fitting a GLM on a set of explanatory variables to observations in each node.

Main steps are:

1. Fit the GLM on the current sample.
2. Assess parameter stability for each partition. variable.
3. Choose the best splitting point.
4. Repeat this process.

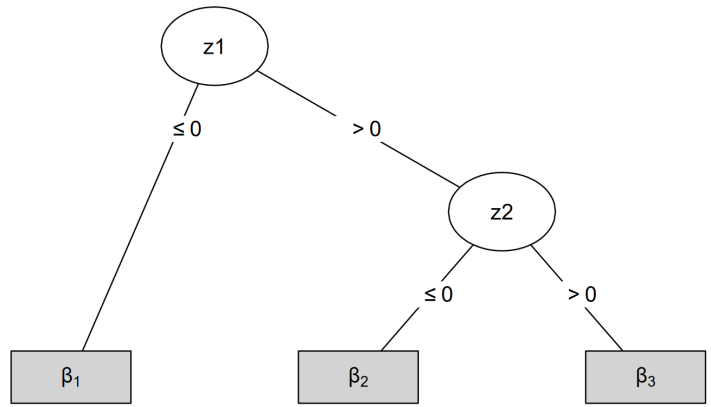


FIGURE 2: Example of GLM tree

Example: GLM tree with 2 partition. variables [SHZ18]:

$$g(E(Y_i)) = \langle \boldsymbol{x}_i, \boldsymbol{\beta}(\boldsymbol{z}) \rangle \text{ with } \begin{cases} \beta_1 & \text{if } z_1 \leq 0 \\ \beta_2 & \text{if } (z_1 > 0) \wedge (z_2 \leq 0) \\ \beta_3 & \text{if } (z_1 > 0) \wedge (z_2 > 0) \end{cases}$$

GLM TREES – A TWO-STEP PROCEDURE AT EACH NODE TO SELECT THE SPLITTING VARIABLE

For a node b, the first two steps are:

1. a GLM is fitted on all observations of the current node b ($i \in b$) possibly with explanatory variables $\boldsymbol{x}_i$.
2. a variable selection is performed based on a M-fluctuation test

$$W_j(t, \hat{\boldsymbol{\theta}}) = \hat{J}^{-1/2} \frac{1}{\sqrt{n_b}} \sum_{i \leq \lfloor t \times \sharp b \rfloor, i \in b} \hat{s}_{\sigma(i),j}, 0 \leq t \leq 1,$$

where $s_{i,j}()$ is the score function

$$s_{i,j}(\boldsymbol{\theta}) = \frac{y_i - \mu_i}{V(\mu_i)} h'(\eta_i) z_{i,j},$$

$\sigma(i)$ is the ordering permutation giving the anti-rank observation of $z_{i,j}$, $\sharp b$ is the cardinality of the set b, $\hat{J} = J(\hat{\boldsymbol{\theta}})$ the fitted covariance matrix, $\mu_i = h(\eta_i)$ and $h = g^{-1}$.

Under the null hypothesis, $W_{j,n}$ converges to a Brownian bridge as $n \to +\infty$, see [ZH07].

Recursive partition algorithm for GLM trees

while *Loop over node b until no significant instability is detected* **do**

0. Compute the observation number: $n_b = \sharp b$ for node b.

if n_b *is too small* **then**

Stop the process for that node.

end

1. Fit the local model:

Fit GLM with $\boldsymbol{x}_i$ for $i \in b$ maximizing (1) $\Rightarrow \hat{\boldsymbol{\theta}}_b$.

2. Assess param. instability of partition. variables with M-fluctuation tests:

for $j = 1, \ldots, q$ **do**

Compute the i-th score contribution as $\hat{s}_{i,j} = s_{i,j}(\hat{\boldsymbol{\theta}}_b)$ for all $i \in b$.

if j *is a numerical variable* **then**

Compute parameter instability as $\lambda_j = \max_{i=\underline{i},\ldots,\bar{i}} \frac{(n_b)^2}{i(n_b-i)} \left\| W_j \left(i/n_b, \hat{\boldsymbol{\theta}}_b \right) \right\|_2^2$,

where $[\underline{i}, \bar{i}]$ is the interval of potential instability.

else

Compute parameter instability as $\lambda_j = \frac{1}{n_b} \sum_{c=1}^{l_j} (\sharp I_{v_{j,c}})^{-1} \left\| \Delta_{v_{j,c}} W_j \left(i/n_b, \hat{\boldsymbol{\theta}}_b \right) \right\|_2^2$,

where $I_{v_{j,c}} = \{ i \in b, z_{i,j} = v_{j,c} \}$ is the set of observation indices in category $v_{j,c}$.

end

end

Compute the p-value of the fluctuation test and assess the significance.

if *there is at least one significant instable variable* **then**

Select the most unstable variable $j^\star = \underset{j \in \{1,\ldots,q\}}{\arg\max} \lambda_j$.

3. Choose the best splitting point s:

if $j^\star$ *is a numerical variable* **then**

Search for the optimal split point $s^\star \in (\min_i z_{i,j^\star}, \max_i z_{i,j^\star})$ maximizing log-likelihood.

else

Search for the optimal set $s^\star \subset \{ v_{j^\star,1}, \ldots, v_{j^\star,l_j} \}$ maximizing log-likelihood.

end

end

end

Algorithm 1: Recursive partition algorithm for GLM Trees

GLM TREES – FINDING THE OPTIMAL SPLIT IN PRACTICE

An exhaustive search is performed.

For instance for a categorical variable z with 4 levels A, B, C, D, the following six GLMs are fitted:

- left node $1_{z \in \{A,B,C\}}$ against right node $1_{z \in \{D\}}$,
- left node $1_{z \in \{A,B,D\}}$ against right node $1_{z \in \{C\}}$,
- left node $1_{z \in \{A,C,D\}}$ against right node $1_{z \in \{B\}}$,
- left node $1_{z \in \{A,B\}}$ against right node $1_{z \in \{C,D\}}$,
- left node $1_{z \in \{A,C\}}$ against right node $1_{z \in \{B,D\}}$,
- left node $1_{z \in \{A,D\}}$ against right node $1_{z \in \{B,C\}}$.

For instance for a continuous variable z, all possible splits are tested in $(\min_i z_i, \max_i z_i)$.

GLM Tree – defining the objective function for optimal split

- Given the best splitting variable, the algorithm searches for the best split point based on

$$O\left(\boldsymbol{y}, \phi, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_B\right) = \sum_{b=1}^{B} \log L(\hat{\boldsymbol{\theta}}_b, \phi, y_i) 1_{\{i \in L_b(j^\star)\}}, \tag{3}$$

  where $L_b(j^\star)$ corresponds to the b-th segment w.r.t. values taken by the variable $j^\star$.
- For binary tree ($B = 2$), only one split point for a continuous variable or one subset for a categorical variable, hereafter noted s, should be exhibited.

- Objective function (3) is generally not explicit since $\hat{\boldsymbol{\theta}}_b$ is estimated by the IWLS algorithm.

- However, explicit objective functions exist when
  - a GLM with no explanatory variable and a set of partitioning variables (continuous and/or categorical).
  - a GLM with a single categorical explanatory variable and a set of partitioning variables (continuous and/or categorical).

For splitting variable $j^\star$, the linear predictor is

$$\eta_i = \theta_L \times 1_{\{i \in L(j^\star, s)\}} + \theta_R \times 1_{\{i \in R(j^\star, s)\}},$$

where $L(j, s)$ and $R(j, s)$ are the children subset:

- numerical: $i \in L(j, s) \Leftrightarrow z_{i,j} \in]-\infty, s]$, $i \in R(j, s) \Leftrightarrow z_{i,j} \in]s, +\infty[$;
- categorical: $i \in L(j, s) \Leftrightarrow i \in s$, $i \in R(j, s) \Leftrightarrow i \notin s$.

Based on [BDR20], [DG22] show that the objective function $s \mapsto O(\overline{\boldsymbol{y}_{j^\star}}(s), \boldsymbol{m}_{j^\star}(s))$ is explicit

$$\begin{aligned} O(\overline{\boldsymbol{y}_{j^\star}}(s), \boldsymbol{m}_{j^\star}(s)) = \tilde{b}\left(\overline{y}^L_{j^\star}(s)\right) m^L_{j^\star}(s)\overline{y}^L_{j^\star}(s) - b\left(\tilde{b}(\overline{y}^L_{j^\star}(s))\right) m^L_{j^\star}(s) \\ + \tilde{b}\left(\overline{y}^R_{j^\star}(s)\right) m^R_{j^\star}(s)\overline{y}^R_{j^\star}(s) - b\left(\tilde{b}(\overline{y}^R_{j^\star}(s))\right) m^R_{j^\star}(s), \end{aligned} \tag{4}$$

where $\tilde{b} = (b')^{-1}$ is the inverse of b'.

	Left node	Right node
Frequency	$m^L_j(s) = \sum_{i=1}^n 1_{\{i \in L(j,s)\}}$	$m^R_j(s) = \sum_{i=1}^n 1_{\{i \in R(j,s)\}}$
Average	$\overline{y}^L_j(s) = \frac{1}{m^L_j(s)} \sum_{i=1}^n y_i 1_{\{i \in L(j,s)\}}$	$\overline{y}^R_j(s) = \frac{1}{m^R_j(s)} \sum_{i=1}^n y_i 1_{\{i \in R(j,s)\}}$

TABLE 2: Notations for conditional frequencies and averages

Examples of distributions for GLM trees

- For a Bernoulli response, Equation (4) becomes of type $p\log(p) + (1-p)\log(1-p)$ as the entropy function used in classification trees, [VR02].
- For a Gaussian response, Equation (4) becomes

$$O(\overline{\boldsymbol{y}}_j, \boldsymbol{m}_j) = \frac{1}{2}\left(\overline{y}_j^L\right)^2 m_j^L + \frac{1}{2}\left(\overline{y}_j^R\right)^2 m_j^R,$$

  and is proportional to the loss deviance used in regression tree, [CH93]
- A gamma distribution with mean μ and shape parameter ν for which $\mathbb{Y} = (0, +\infty)$, $\Lambda = \mathbb{R}_-$

$$O(\overline{\boldsymbol{y}}_j, \boldsymbol{m}_j) = -m_j^L\left(1 + \log(\overline{y}_j^L)\right) - m_j^R\left(1 + \log(\overline{y}_j^R)\right).$$

- A Poisson distribution with a mean μ for which $\mathbb{Y} = \mathbb{N}$ and $\Lambda = \mathbb{R}$

$$O(\overline{\boldsymbol{y}}_j, \boldsymbol{m}_j) = \left(\log(\overline{y}_j^L) - 1\right)\overline{y}_j^L m_j^L + \left(\log(\overline{y}_j^R) - 1\right)\overline{y}_j^R m_j^R.$$

- An Inverse Gaussian distribution with mean μ and shape parameter σ^2 for which $\mathbb{Y} = (0, +\infty)$ and $\Lambda = (-\infty, 0)$

$$O(\overline{\boldsymbol{y}}_j, \boldsymbol{m}_j) = \frac{m_j^L}{2\overline{y}_j^L} + \frac{m_j^R}{2\overline{y}_j^R}.$$

These examples illustrate non-quadratic objective functions for continuous distributions and non-logit objective functions for discrete distributions.

Taking weights into account

Similarly to [BDR20], the weighted MLE is $\widehat{\theta}(s) = (\hat{\theta}_L(s), \hat{\theta}_R(s))$ obtained by changing arithmetical means $\overline{y}_j^{L/R}(s)$ to weighted means $\overline{y}_{j,w}^{L/R}(s)$.

Equation (4) is used with means given in Table 3.

	Left node	Right node
Frequency	$m_{j,w}^L(s) = \sum_{i=1}^n w_i 1_{\{i \in L(j,s)\}}$	$m_{j,w}^R(s) = \sum_{i=1}^n w_i 1_{\{i \in R(j,s)\}}$
Average	$\overline{y}_{j,w}^L(s) = \frac{1}{m_{j,w}^L(s)} \sum_{i=1}^n w_i y_i 1_{\{i \in L(j,s)\}}$	$\overline{y}_{j,w}^R(s) = \frac{1}{m_{j,w}^R(s)} \sum_{i=1}^n w_i y_i 1_{\{i \in R(j,s)\}}$

Table 3: Notations for conditional weighted frequency and average

Example – the binomial distribution:

- we model Y_i/m_i when $Y_i \sim \mathcal{B}(m_i, p_i)$.
- A weighted MLE is used without changing a, b function of the Bernoulli distribution.
- c becomes $c(x, \phi) = 1/m_i \log \binom{m_i}{m_i x}$.

Outlines

SIMULATED DATASETS CONFIGURATION

We consider various test datasets based on [Woo11] for benchmarking GAM.

- we generate independent and uniformly random variables $(x_{i,j})_{i,j}$.
- we simulate continuous independent variables Y_i, $i = 1, \dots, n$ with mean $\mu_i = g^{-1}(\eta_i)$ where

$$\eta_i = 1 + \sum_{j=1}^{m} f_{j-1(\text{mod } 15)}(x_{i,j}),$$

where f_j are nonlinear test functions.

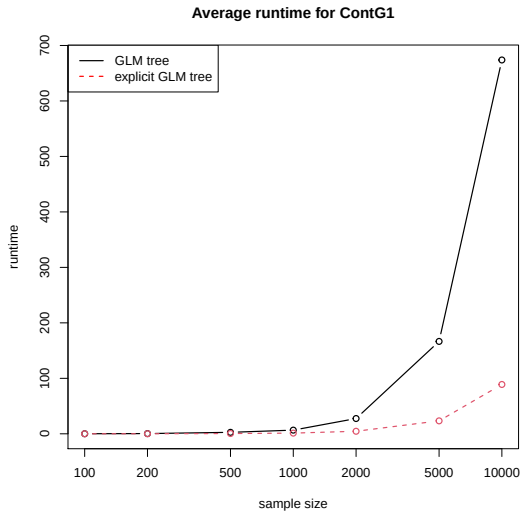
Distribution	μ_i	ϕ
Gaussian $\mathcal{N}(\mu_i, \sigma^2)$	$\mu_i = \eta_i$	0.25
gamma $\mathcal{G}(\nu, \mu_i)$	$\mu_i = e^{\eta_i/5}$	0.25
inverse Gaussian $\mathcal{IG}(\mu_i, \sigma^2)$	$\mu_i = e^{\eta_i/5}$	0.1

TABLE 4: Mean and dispersion parameters μ, ϕ used in simulations

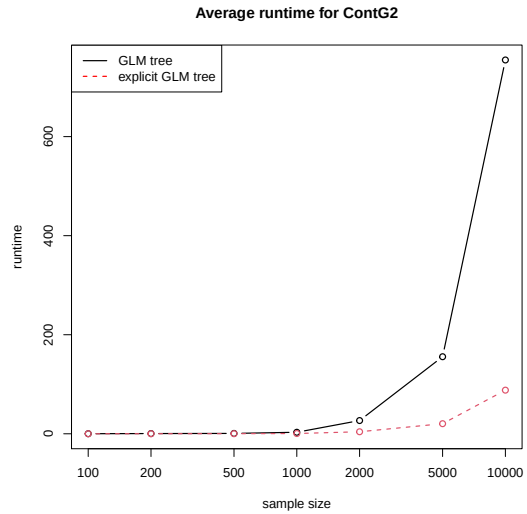
`variable`	`distribution`	`number`
`cont` for continuous expl. variables	`IG` for inverse Gaussian	1 for $m = 10$ covariates
`categ` for categorical expl. variables	`G` for gamma	2 for $m = 20$ covariates

TABLE 5: Naming convention for datasets `<variable><distribution><number>`

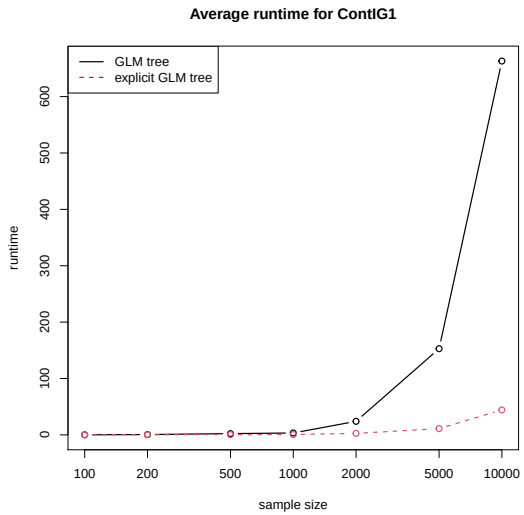
RUNTIME COMPARISON BETWEEN *GLM tree* AND *explicit GLM tree*



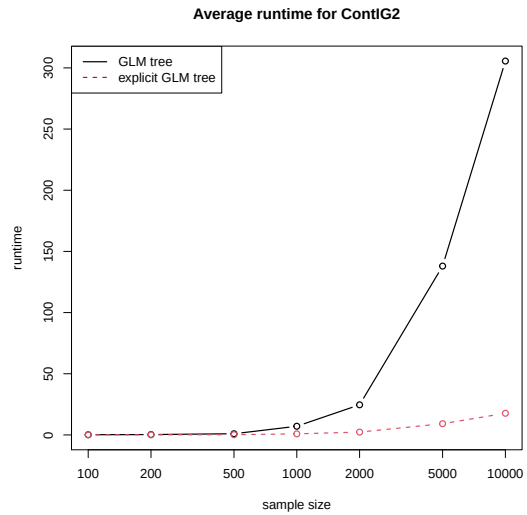
(A) Gamma $m = 10$



(B) Gamma $m = 20$



(C) Inverse Gaussian $m = 10$



(D) Inverse Gaussian $m = 20$

Comparison with benchmark models

We compare the performance in accuracy, complexity and computation time of

- a GLM tree (*Explicit GLM Tree*) with a intercept-node only;
- a GLM tree with a one explanatory variable (*GLM Tree reg*),
- conditional inference trees *CTREE* with different test specifications,
- a linear model tree based on `lmtree`, which is equivalent to GLM trees with a Gaussian distribution,
- *CART* trees based on **rpart**.

We perform a bootstrap cross-validation approach for each dataset with a sample size $n = 1000$.

COMPARISON WITH BENCHMARK MODELS – RMSE

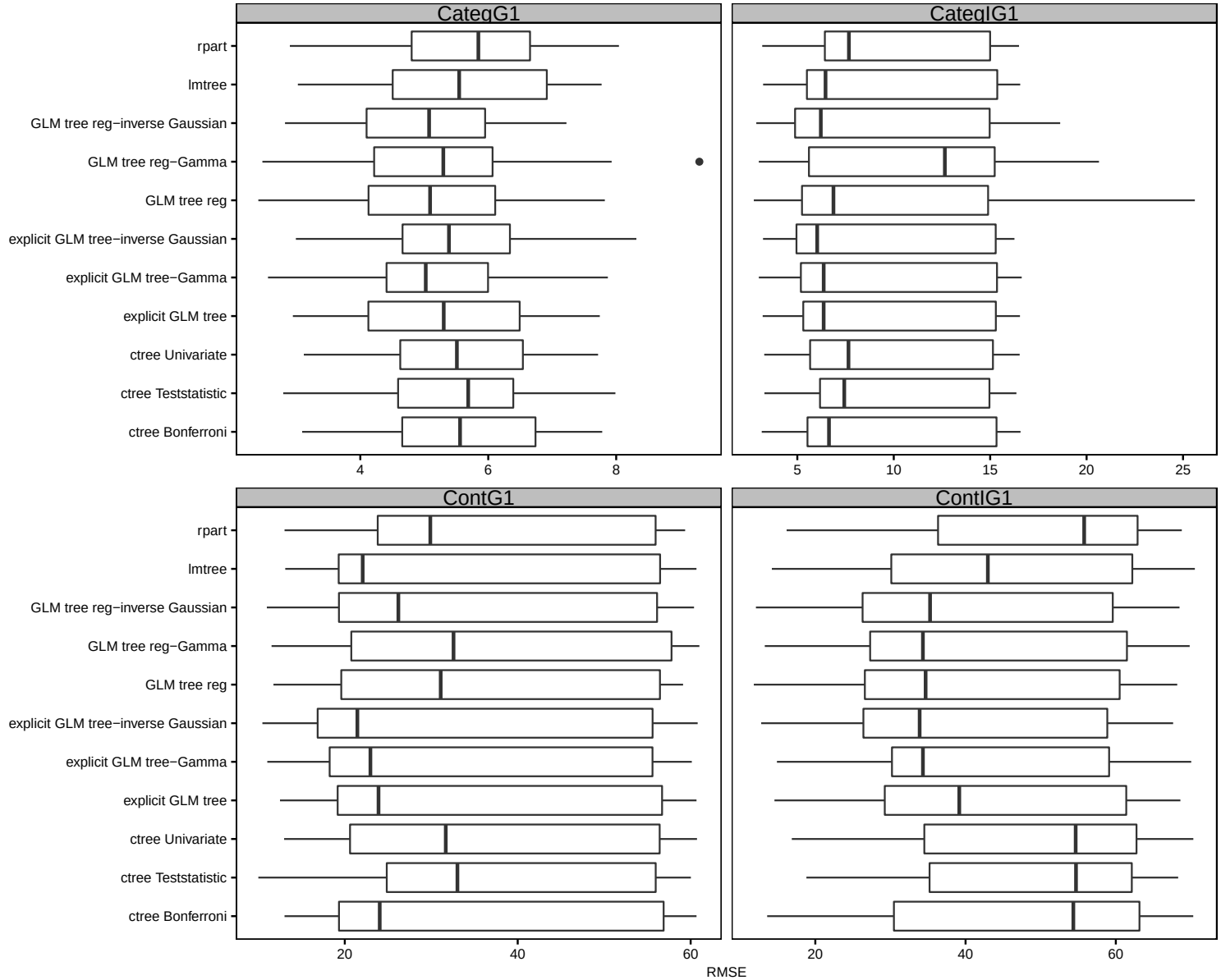


FIGURE 4: Predictive RMSE with 100 bootstrap replications

Comparison with benchmark models – Complexity

rpart produces less complex trees, hence it retains an important interest compared to its competitors by providing more interpretable and simpler to explain results.

Method	CategG1	CategIG1	ContG1	ContIG1
ctree Bonferroni	51.770 (4.156)	64.570 (4.841)	7.880 (3.006)	18.690 (3.826)
ctree Teststatistic	77.430 (2.417)	77.520 (2.834)	53.000 (8.299)	68.560 (5.907)
ctree Univariate	64.880 (3.264)	71.910 (3.232)	17.830 (5.520)	36.950 (6.559)
GLM tree reg	30.990 (1.982)	32.220 (1.643)	5.520 (3.301)	13.660 (5.769)
GLM tree reg-Gamma	30.022 (2.022)	32.225 (1.814)	5.340 (2.886)	12.420 (6.054)
GLM tree reg-inverse Gaussian	30.391 (1.827)	32.261 (1.725)	5.140 (3.291)	13.210 (5.186)
explicit GLM tree	31.640 (1.738)	31.090 (2.327)	10.950 (2.858)	25.800 (3.162)
explicit GLM tree-Gamma	31.450 (1.714)	31.170 (2.070)	10.220 (2.939)	25.030 (3.395)
explicit GLM tree-inverse Gaussian	31.420 (1.742)	30.800 (2.429)	12.560 (3.016)	22.990 (3.611)
lmtree	32.010 (1.888)	32.130 (2.377)	11.370 (3.139)	24.130 (3.620)
rpart	9.530 (1.795)	6.880 (3.291)	5.040 (2.558)	5.360 (1.494)

Table 6: Mean predictive complexity over 100 bootstrap replications with standard deviations in parentheses.

rpart is fast as it relies on C code, whereas the **partykit** package is entirely developed in the R language.

Method	CategG1	CategIG1	ContG1	ContIG1
ctree Bonferroni	0.541 (0.247)	0.418 (0.073)	0.089 (0.043)	0.162 (0.041)
ctree Teststatistic	0.682 (0.326)	0.471 (0.079)	0.365 (0.138)	0.415 (0.084)
ctree Univariate	0.622 (0.312)	0.453 (0.091)	0.166 (0.081)	0.262 (0.068)
GLM tree reg	0.372 (0.043)	0.484 (0.068)	2.310 (1.215)	3.095 (1.071)
GLM tree reg-Gamma	0.484 (0.067)	0.517 (0.070)	2.214 (1.076)	2.980 (1.350)
GLM tree reg-inverse Gaussian	0.436 (0.069)	0.471 (0.064)	2.141 (1.455)	3.590 (1.766)
explicit GLM tree	0.922 (0.452)	0.687 (0.170)	2.943 (1.711)	3.047 (0.461)
explicit GLM tree-Gamma	0.658 (0.099)	0.674 (0.104)	3.826 (0.651)	4.494 (0.585)
explicit GLM tree-inverse Gaussian	0.612 (0.102)	0.632 (0.134)	2.887 (0.659)	2.761 (0.338)
GLM tree	1.121 (0.525)	0.827 (0.163)	5.243 (1.857)	5.708 (0.828)
GLM tree-Gamma	1.325 (0.204)	1.326 (0.197)	17.887 (2.517)	18.601 (1.887)
GLM tree-inverse Gaussian	1.433 (0.227)	1.432 (0.274)	16.165 (4.342)	21.721 (1.693)
lmtree	0.713 (0.357)	0.483 (0.117)	1.651 (1.017)	1.493 (0.308)
rpart	0.022 (0.009)	0.015 (0.007)	0.022 (0.015)	0.018 (0.004)

TABLE 7: Mean predictive runtime of different methods over 100 bootstrap replications.

PERFORMANCE ON BOSTONHOUSING AND HITTERS DATASETS

We assess the performance on two public benchmark datasets: `BostonHousing` and `Hitters` from R packages **mlbench** and **ISLR**.

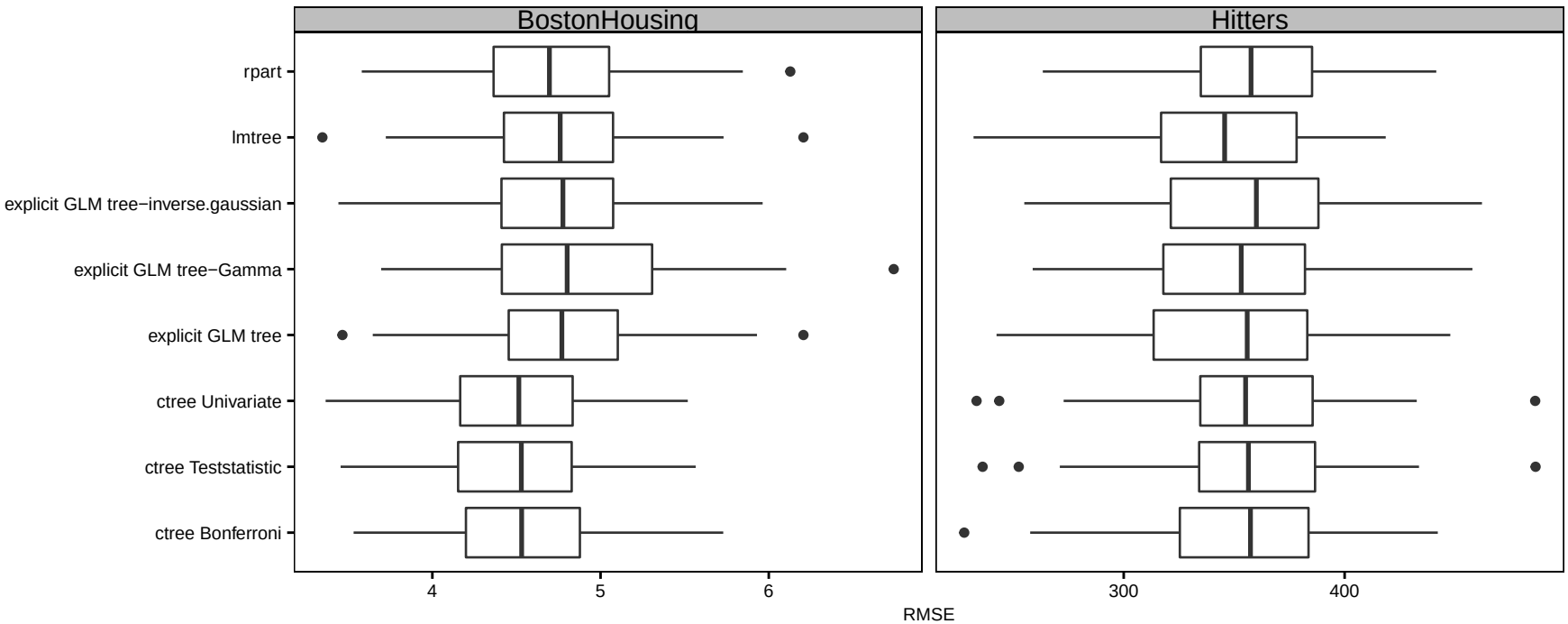


FIGURE 5: Predictive RMSE with 100 bootstrap replications for `BostonHousing` and `Hitters`

Outlines

Random forest based on GLM Trees

We assess the benefits of our approach based on a closed-form formula by implementing a random forest type approach for GLM tree model called *GLM forest*.

We compare the performance of GLM forest against two classical random forest competitors:

- the function `cforest` from package **partykit** to fit random forests based on *CTREE*,
- the function `randomForest` from the R package **randomForest**.

Datasets and implementation details

We use `ContG2` and `ContIG2` with $m = 20$ continuous explanatory variables for gamma or inverse Gaussian responses and $n = 1000$ observations, see Table 5.

We consider

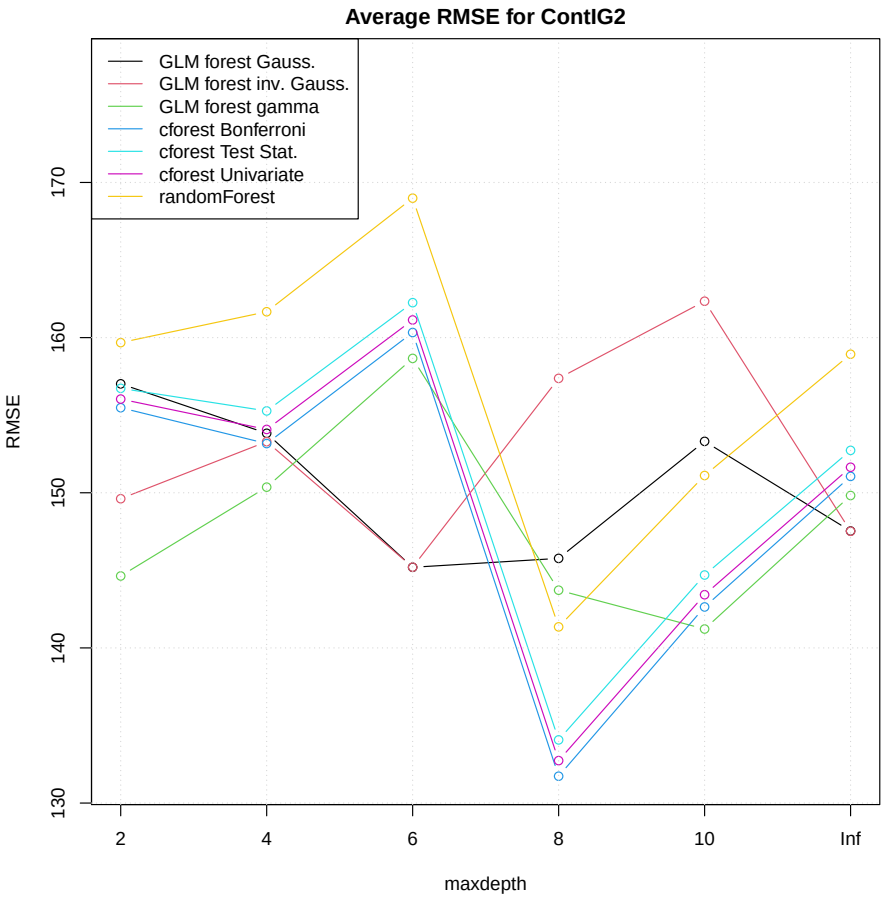
- three versions of *GLM forest* by choosing Gaussian, gamma and inverse Gaussian distributions (with canonical link).
- Regarding *cforest*, we also consider three versions depending on the way the distribution of the test statistic is computed: *Teststatistic* refers to the raw statistic, *Bonferroni* and *Univariate* correspond respectively to adjusted and unadjusted p-values
- `randomForest` from **randomForest**

We control for

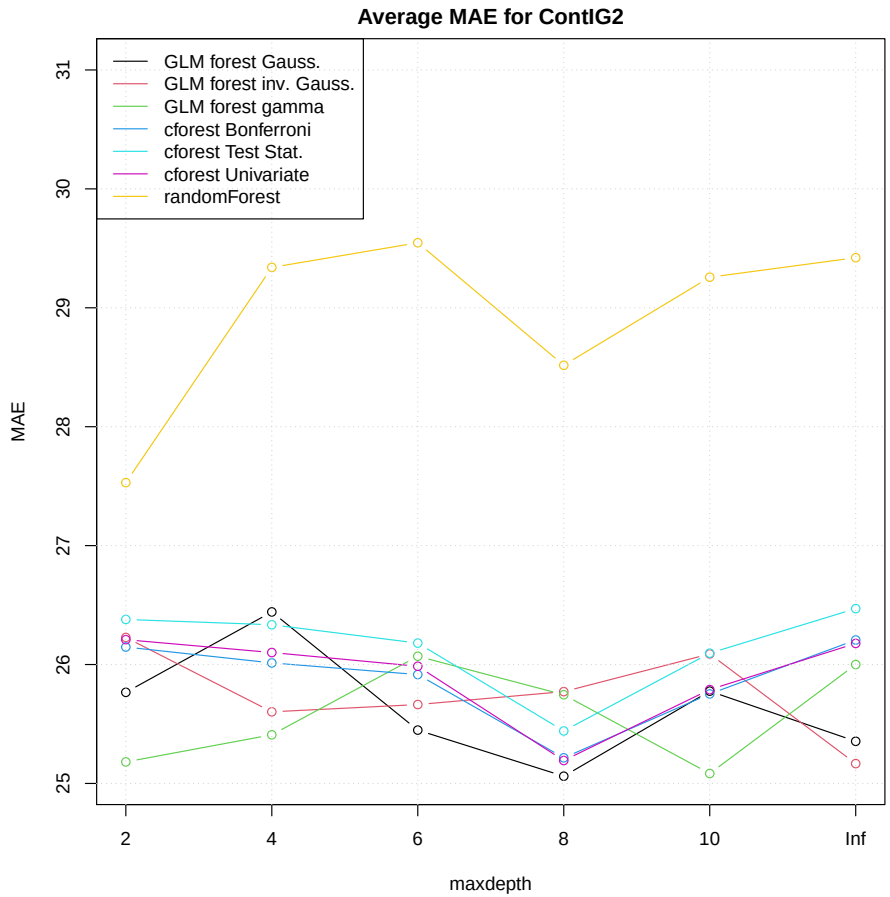
- the number of trees, called `ntree`,
- the number of input variables randomly sampled as splitting candidates at each node, called `mtry`,
- the maximum depth of the tree, called `maxdepth`.

For `randomForest`, the terminal node number is capped by 2^{maxdepth} since there is no argument for the maximum depth.

BENCHMARK ACCURACY RESULTS – INVERSE GAUSSIAN



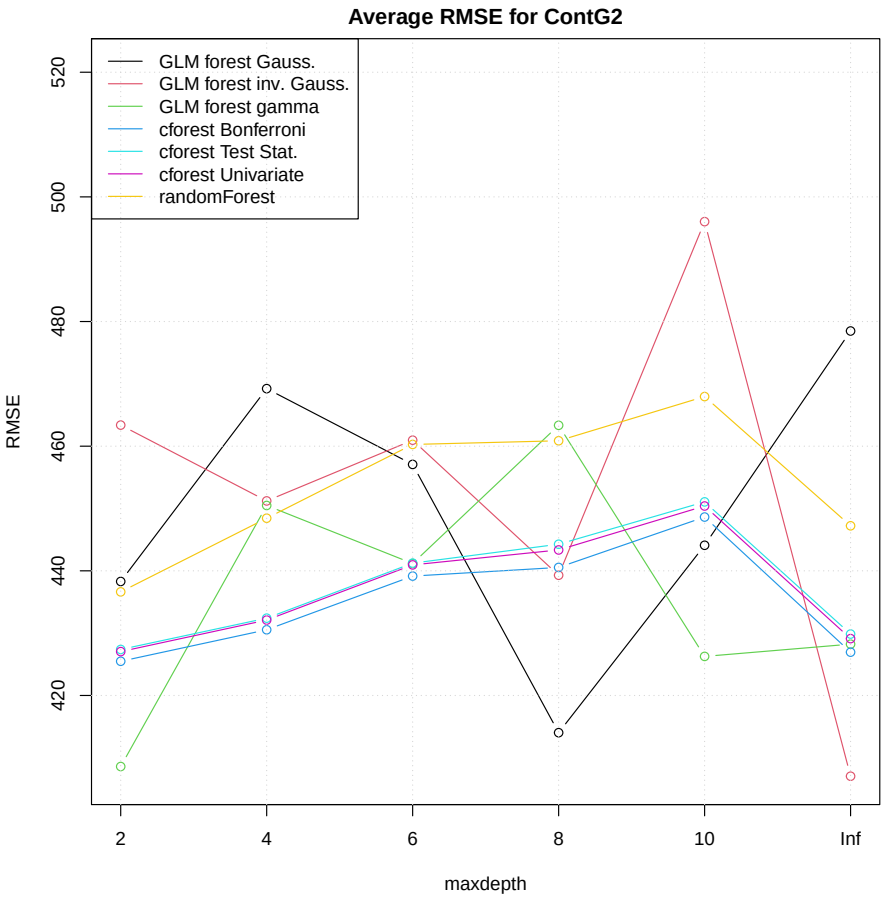
(A) Inverse Gaussian $m = 20$



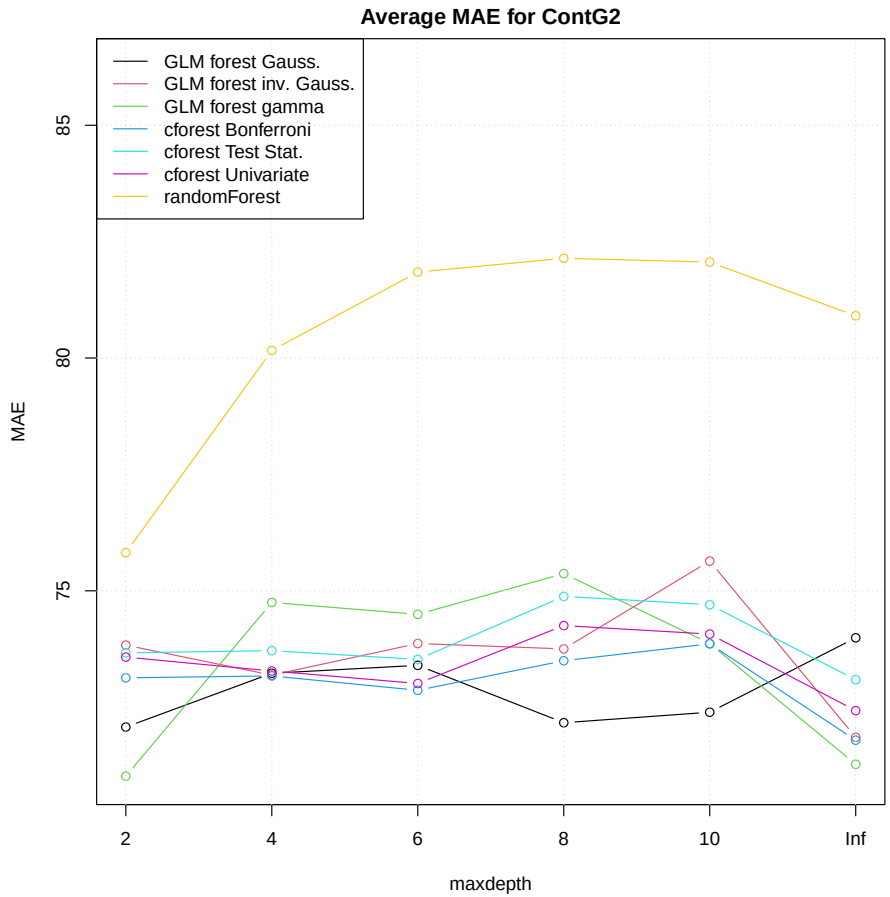
(B) Inverse Gaussian $m = 20$

FIGURE 6: Error metrics as a function of `maxdepth`

BENCHMARK ACCURACY RESULTS – GAMMA



(A) Gamma $m = 20$



(B) Gamma $m = 20$

FIGURE 7: Error metrics as a function of `maxdepth`

RUNTIME AND COMPLEXITY ANALYSIS

			Complexity		Computation time	
data	method	family	mean	median	mean	median
ContG2	cforest_Bonferroni	Gaussian	728.00	640.5	2.03	1.72
	cforest_Teststatistic	Gaussian	13244.31	13226.0	34.90	34.86
	cforest_Univariate	Gaussian	1775.54	1732.5	5.61	5.59
	glmforest	Gamma	1263.94	1224.5	219.71	219.28
	glmforest	Gaussian	1432.09	1388.0	177.48	181.01
	glmforest	Inverse Gaussian	1516.63	1504.0	176.86	178.98
	randomForest	gaussian	127905.45	127912.0	3.55	3.55
ContIG2	cforest_Bonferroni	Gaussian	845.10	794.0	5.39	4.20
	cforest_Teststatistic	Gaussian	15303.13	15202.0	82.62	61.72
	cforest_Univariate	Gaussian	2670.41	2567.0	18.35	14.38
	glmforest	Gamma	1451.66	1387.5	424.80	406.55
	glmforest	Gaussian	1587.87	1530.0	253.51	247.73
	glmforest	Inverse Gaussian	1480.79	1381.5	224.90	213.51
	randomForest	gaussian	127306.26	127334.0	7.18	5.75

TABLE 8: Complexity and runtime mean and median for ContG2 and ContIG2 over 100 runs, maxdepth=8

Conclusion and perspectives

In this presentation, we

- propose a new fast algorithm for growing GLM trees,
- demonstrate that this approach greatly increases the computation speed of the GLM-based tree model,
- derive a GLM forest algorithm.

This approach opens up some pathways for future research.

- Other types of distributions could be studied in the framework model-based trees,
  - for instance inflated distributions such as zero-inflated Poisson,
  - two-parameter exponential families such as beta, negative binomial distributions or heavy-tailed distributions.
- In addition, we believe this method can be applied to other ensemble decision tree algorithm,
  - such as boosted trees,
  - or for prediction rule ensembles where the features of MOBs is of interest for interpretable rule generation.

References I

[BDR20] Brouste, A., Dutang, C., and Rohmer, T. Closed form Maximum Likelihood Estimator for Generalized Linear Models in the case of categorical explanatory variables: Application to insurance loss modelling. In: *Computational Statistics* 35 (2020), pp. 689–724 (cit. on pp. 17, 19, 40–45).

[Bre01] Breiman, L. Random Forests. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32 (cit. on p. 5).

[Bre+84] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. Classification and Regression Trees. New Ed. Boca Raton: Chapman and Hall/CRC, Jan. 1984 (cit. on p. 4).

[Bre96] Breiman, L. Bagging Predictors. In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140 (cit. on p. 5).

[CH93] Chambers, J. and Hastie, T. Statistical Models in S. Chapman and Hall, 1993 (cit. on p. 18).

[CV95] Cortes, C. and Vapnik, V. Support-vector networks. In: *Machine Learning* 20.3 (1995), pp. 273–297 (cit. on p. 5).

[DG22] Dutang, C. and Guibert, Q. An explicit split point procedure in model-based trees allowing for a quick fitting of GLM trees and GLM forests. In: *Statistics and Computing* 32.6 (2022) (cit. on p. 17).

[Fri02] Friedman, J. H. Stochastic gradient boosting. In: *Computational Statistics & Data Analysis*. Nonlinear Methods and Data Mining 38.4 (Feb. 2002), pp. 367–378 (cit. on p. 5).

[HHZ06] Hothorn, T., Hornik, K., and Zeileis, A. Unbiased Recursive Partitioning: A Conditional Inference Framework. In: *Journal of Computational and Graphical Statistics* 15.3 (Sept. 2006), pp. 651–674 (cit. on p. 4).

[KL01] Kim, H. and Loh, W.-Y. Classification Trees With Unbiased Multiway Splits. In: *Journal of the American Statistical Association* 96.454 (June 2001). Publisher: Taylor & Francis _eprint: https://doi.org/10.1198/016214501753168271, pp. 589–604 (cit. on p. 4).

[Law94] Lawrence, J. Introduction To Neural Networks: Design, Theory and Applications. 6th. California Scientific Software, 1994 (cit. on p. 5).

[Loh14] Loh, W.-Y. Fifty Years of Classification and Regression Trees. In: *International Statistical Review* 82.3 (2014), pp. 329–348 (cit. on p. 4).

[LS97] Loh, W.-Y. and Shih, Y.-S. Split Selection Methods for Classification Trees. In: *Statistica Sinica* 7.4 (1997). Publisher: Institute of Statistical Science, Academia Sinica, pp. 815–840 (cit. on p. 4).

[LV88] Loh, W.-Y. and Vanichsetakul, N. Tree-Structured Classification via Generalized Discriminant Analysis. In: *Journal of the American Statistical Association* 83.403 (Sept. 1988). Publisher: Taylor & Francis, pp. 715–725 (cit. on p. 4).

[NW72] Nelder, J. A. and Wedderburn, R. W. M. Generalized Linear Models. In: *Journal of the Royal Statistical Society. Series A (General)* 135.3 (1972), pp. 370–384 (cit. on p. 8).

References II

[RZ13] Rusch, T. and Zeileis, A. Gaining insight with recursive partitioning of generalized linear models. In: *Journal of Statistical Computation and Simulation* 83.7 (July 2013), pp. 1301–1315 (cit. on p. 12).

[SHZ18] Seibold, H., Hothorn, T., and Zeileis, A. Generalised linear model trees with global additive effects. In: *Advances in Data Analysis and Classification* (Oct. 2018) (cit. on p. 12).

[VR02] Venables, W. and Ripley, B. Modern Applied Statistics with S. Springer, 2002 (cit. on p. 18).

[Woo11] Wood, S. N. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.1 (2011), pp. 3–36 (cit. on pp. 21, 46).

[ZH07] Zeileis, A. and Hornik, K. Generalized M-fluctuation tests for parameter instability. In: *Statistica Neerlandica* 61.4 (2007), pp. 488–508 (cit. on p. 13).

[ZHH08] Zeileis, A., Hothorn, T., and Hornik, K. Model-Based Recursive Partitioning. In: *Journal of Computational and Graphical Statistics* 17.2 (June 2008), pp. 492–514 (cit. on p. 11).

Appendix

For the exponential family, we have

- Expectation
$$E(X) = \mu = b'(\theta),$$
- Variance
$$Var(X) = a(\phi)V(\mu) = a(\phi)b''(\theta),$$
where V is the unit variance function.
- Skewness
$$\gamma_3(X) = \frac{dV}{d\mu}(\mu)\sqrt{\frac{a(\phi)}{V(\mu)}} = \frac{b^{(3)}(\theta)a(\phi)^2}{Var(Y)^{3/2}},$$
- Kurtosis
$$\gamma_4(X) = 3 + \left[\frac{d^2V}{d\mu^2}(\mu)V(\mu) + \left(\frac{dV}{d\mu}(\mu)\right)^2\right]\frac{a(\phi)}{V(\mu)} = 3 + \frac{b^{(4)}(\theta)a(\phi)^3}{Var(Y)^2}.$$

IWLS ALGORITHM FOR GLM

The IWLS algorithm is

1. Init:
   1. Shift $\mu_i^{(0)} = y_i + 0.1$ pour calculer $\eta_i^{(0)} = g(\mu_i^{(0)})$.
   2. Compute working response $Z^{(0)} = (\eta_i^{(0)} + (y_i - \mu_i^{(0)})g'(\mu_i^{(0)}))_i$.
   3. Compute working weights $W^{(0)} = \text{diag}(w_1, \ldots, w_n)$ and $w_i = \frac{1}{a(\phi_i)(g'(\mu_i^{(0)}))^2 V(\mu_i^{(0)})}$.
   4. Solve
   $$X^T W^{(0)} X \beta^{(0)} = X^T W^{(0)} Z^{(0)}.$$
2. Iterate: for $k = 1, \ldots, m$ do
   1. Compute working response $Z^{(k)} = (z_i)_i$ and $z_i = \eta_i(\beta^{(k)}) + (y_i - \mu_i(\beta^{(k)}))g'(\mu_i(\beta^{(k)}))$.
   2. Compute working weights $W^{(k)} = \text{diag}(w_1, \ldots, w_n)$ and $w_i = \frac{1}{a(\phi_i)(g'(\mu_i(\beta^{(k)}))^2 V(\mu_i(\beta^{(k)}))}$.
   3. Solve
   $$X^T W^{(k)} X \beta^{(k+1)} = X^T W^{(k)} Z^{(k)}.$$
   4. Check convergence: $||Dev(\beta^{(k+1)}) - Dev(\beta^{(k)})|| \leq \epsilon$.

In practice, $X^T W^{(k)} X \beta^{(k+1)} = X^T W^{(k)} Z^{(k)}$ is solved by QR decompostion.

Situations where a closed-form MLE exists: A single categorical explanatory variable

Consider $x_i^{(1)} = 1$ is the intercept and $x_i^{(2)}$ takes values in $\{v_1, \ldots, v_{d_2}\}$ with $d_2 > 2$.

Theorem ([BDR20])

Suppose that for all $i \in I$, Y_i takes values in $b'(\Lambda)$. If the vector $\boldsymbol{R}$ is such that $\boldsymbol{R}\boldsymbol{\theta} = 0$ and $\sum_{j=1}^{d_2} r_j - r_0 \neq 0$, then there exists a unique, consistent and explicit MLE $\widehat{\boldsymbol{\theta}}_n$ of $\boldsymbol{\theta}$ given by

$$\widehat{\theta}_{n,(1)} = \frac{\sum_{k=1}^{d_2} r_k g(\overline{Y}_n^{(k)})}{\sum_{k=1}^{d_2} r_k - r_0}, \widehat{\theta}_{n,(2),j} = g(\overline{Y}_n^{(j)}) - \frac{\sum_{k=1}^{d_2} r_k g(\overline{Y}_n^{(k)})}{\sum_{k=1}^{d_2} r_k - r_0}, j \in J,$$

where $m_j = \sum_{i=1}^{n} x_i^{(2),j}, \quad j \in J \quad$ *and* $\quad \overline{y}_n^{(j)} = \frac{1}{m_j} \sum_{i=1}^{n} y_i x_i^{(2),j}, \quad j \in J.$

Corollary ([BDR20])

The fitted log-likelihood does not depend on the link function g.

$$\log L(\widehat{\boldsymbol{\theta}}_n \,|\, \boldsymbol{y}) = \frac{1}{a(\phi)} \sum_{j=1}^{d} \sum_{i, x_i^{(2),j}=1} \left(y_i \tilde{b}\left(\overline{y}_n^{(j)}\right) - b\left(\tilde{b}\left(\overline{y}_n^{(j)}\right)\right)\right) + \sum_{i=1}^{n} c(y_i, \phi), \tilde{b} = \left(b'\right)^{-1}.$$

Case 1: A single categorical explanatory variable

Consider $x_i^{(1)} = 1$ is the intercept and $x_i^{(2)}$ takes values in a set of d_2 modalities $\{v_1, \ldots, v_{d_2}\}$ with $d_2 > 2$.

To perform the estimation, $\boldsymbol{R}\boldsymbol{\theta} = 0$ and an incidence matrix is derived

$$\left(x_i^{(2),j}\right)_{i,j} = \left(\mathbf{1}_{x_i^{(2)}=v_j}\right)_{i,j} \Rightarrow g(E(Y_i)) = \theta_{(1)} + \sum_{k=1}^{d_2} x_i^{(2),k}\theta_{(2),k},$$

where $x_i^{(2),j}$ is the binary dummy of the jth category for $i \in I$ and $j \in J = \{1, \ldots, d_2\}$.

Theorem ([BDR20])

Suppose that for all $i \in I$, Y_i takes values in $b'(\Lambda)$. If the vector $\boldsymbol{R}$ is such that $\sum_{j=1}^{d_2} r_j - r_0 \neq 0$, then there exists a unique, consistent and explicit MLE $\widehat{\boldsymbol{\theta}}_n$ of $\boldsymbol{\theta}$ given by

$$\widehat{\theta}_{n,(1)} = \frac{\sum_{k=1}^{d_2} r_k g(\overline{Y}_n^{(k)})}{\sum_{k=1}^{d_2} r_k - r_0}, \widehat{\theta}_{n,(2),j} = g(\overline{Y}_n^{(j)}) - \frac{\sum_{k=1}^{d_2} r_k g(\overline{Y}_n^{(k)})}{\sum_{k=1}^{d_2} r_k - r_0}, j \in J,$$

where $m_j = \sum_{i=1}^{n} x_i^{(2),j}, \quad j \in J \quad \text{and} \quad \overline{y}_n^{(j)} = \frac{1}{m_j}\sum_{i=1}^{n} y_i x_i^{(2),j}, \quad j \in J.$

Note that if $\overline{Y}_n^{(j)}$ does not belong to $b'(\Lambda)$, $g(\overline{Y}_n^{(j)})$ and hence $\widehat{\theta}_{n,(l),j}$ are not defined.

Examples

- no-intercept: $\widehat{\theta}_{n,(1)} = 0$, $\widehat{\theta}_{n,(2),j} = g(\overline{Y}_n^{(j)})$;
- no first-level: $\widehat{\theta}_{n,(1)} = g\left(\overline{Y}_n^{(1)}\right), \widehat{\theta}_{n,(2),1} = 0, \widehat{\theta}_{n,(2),j} = g\left(\overline{Y}_n^{(j)}\right) - \widehat{\theta}_{n,1}, j \in J \setminus \{1\}$.

COROLLARY ([BDR20])

The fitted log-likelihood ($\Rightarrow$ AIC, BIC) does not depend on the link function g.
$\forall i \in I, \quad \ell(\hat{\eta}_i) = (b')^{-1}(\overline{y}_n^{(j)})$ *for $j \in J$ such that $x_i^{(2),j} = 1$ and*

$$\log L(\widehat{\boldsymbol{\theta}}_n \mid \boldsymbol{y}) = \frac{1}{a(\phi)} \sum_{j=1}^{d} \sum_{i, x_i^{(2),j}=1} \left(y_i \tilde{b}\left(\overline{y}_n^{(j)}\right) - b\left(\tilde{b}\left(\overline{y}_n^{(j)}\right)\right)\right) + \sum_{i=1}^{n} c(y_i, \phi),$$

with $\tilde{b} = (b')^{-1}$.

The estimator of ϕ is obtained by maximizing $\log L(\widehat{\boldsymbol{\theta}}_n \mid \boldsymbol{y})$ with respect to ϕ given a, b, c functions.

CASE 2: TWO CATEGORICAL EXPLANATORY VARIABLES

Dummy	Frequency	Mean	Index
$x_i^{(2),k} = 1_{x_i^{(2)}=v_{2k}}$	$m_k^{(2)} = \sum_{i=1}^n x_i^{(2),k}$	$\overline{y}_n^{(2),k} = \frac{1}{m_k^{(2)}} \sum_{i=1}^n y_i x_i^{(2),k}$	$k \in K = \{1, \ldots, d_2\}$
$x_i^{(3),l} = 1_{x_i^{(3)}=v_{3l}}$	$m_l^{(3)} = \sum_{i=1}^n x_i^{(3),l}$	$\overline{y}_n^{(3),l} = \frac{1}{m_l^{(3)}} \sum_{i=1}^n y_i x_i^{(3),l}$	$l \in L = \{1, \ldots, d_3\}$
$x_i^{(k,l)} = x_i^{(2),k} x_i^{(3),l}$	$m_{k,l} = \sum_{i=1}^n x_i^{(k,l)}$	$\overline{y}_n^{(k,l)} = \frac{1}{m_{k,l}} \sum_{i=1}^n y_i x_i^{(k,l)}$	$(k,l) \in K \times L$

TABLE 9: Dummies, frequencies and averages w.r.t explanatory variables

We define $Q = (\mathbf{1}_{d_2 d_3}, \mathbf{1}_{d_3} \otimes I_{d_2}, I_{d_3} \otimes \mathbf{1}_{d_2}, I_{d_2 d_3})$, where $\otimes$ is the Kronecker product, and a contrast matrix R.

Consider GLM

$$g(E(Y_i)) = \theta_1 + \sum_{k=1}^{d_2} x_i^{(2),k} \theta_{(2),k} + \sum_{l=1}^{d_3} x_i^{(3),l} \theta_{(3),l} + \sum_{k=1}^{d_2} \sum_{l=1}^{d_3} x_i^{(k,l)} \theta_{k,l}.$$

THEOREM ([BDR20])

Suppose that for all $i \in \{1, \ldots, n\}$, Y_i takes values in $b'(\Lambda)$. Under $\boldsymbol{R}\boldsymbol{\theta} = \mathbf{0}$, and if $\boldsymbol{R}$ such that $(\boldsymbol{Q}', \boldsymbol{R}')$ is of rank $d_2 d_3$, there exists a unique, consistent and explicit MLE $\widehat{\theta}_n$ of $\boldsymbol{\theta}$ given by

$$\widehat{\theta}_n = (\boldsymbol{Q}'\boldsymbol{Q} + \boldsymbol{R}'\boldsymbol{R})^{-1} \boldsymbol{Q}' \boldsymbol{g}(\overline{\boldsymbol{Y}}), \tag{5}$$

where the vector $\boldsymbol{g}(\overline{\boldsymbol{Y}})$ is $((g(\overline{Y}_n^{(k,l)}))_l)_k$.

[BDR20] study:

- exact distribution of the MLE $\widehat{\theta}_n$ for Pareto 1 and lognormal,
- compute the bias of the MLE,
- a closed-form MLE estimator of the dispersion ϕ,
- model diagnostic via residuals with a known distribution.

[BDR20] provide an illustration an actuarial dataset of 211,739 claims of corporate business lines : they fit GLM for claim amount above $\mu = 340,000$ (in euros).

TABLE 10: Coefficients for the guarantee variable

Model	Pareto 1			Shifted log normal	
Variable	canonical	loginv	shifted.loginv	canonical	symlog
Intercept	1.89	0.64	-0.11	11.75	2.46
Guarantee 2	0.04	0.02	0.04	0.10	0.01
Guarantee 3	-0.67	-0.43	-1.36	0.75	0.06
Guarantee 4	-0.86	-0.60	-3.13	1.04	0.08
Guarantee 5	-0.71	-0.47	-1.55	0.72	0.06
Guarantee 6	-0.42	-0.25	-0.63	0.42	0.04
Guarantee 7	-0.48	-0.29	-0.78	0.59	0.05
log likelihood	-14507.53	-14507.53	-14507.53	-14517.37	-14517.37

[BDR20] analyze two distributions: Pareto 1 and shifted log-normal.

Using a transformation T such that $Z_i = T(Y_i)$ belongs to the exponential family (1), see Table 11.

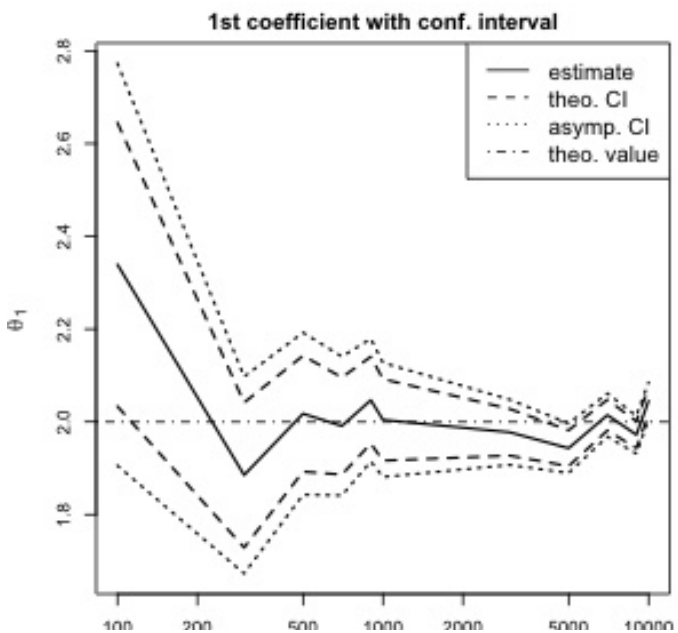
Name	$T(x)$	$a(x)$	$b(x)$	$c(x, \phi)$
Pareto 1	$-\log(x/\mu)$	1	$-\log(\lambda)$	0
shifted lognormal	$\log(x-\mu)$	x	$x^2/2$	$-\frac{1}{2}(x^2/\phi + \log(2\pi\phi))$

TABLE 11: log-transformed distributions

[BDR20] study

- exact distribution of the MLE $\widehat{\theta}_n$ for Pareto 1 and lognormal,
- compute the bias of the MLE,
- a closed-form MLE estimator of the dispersion ϕ,
- model diagnostic via residuals with a known distribution.

FIGURE 8: Pareto 1 – coef $\tilde{\theta}_1^{(2)}$



Simon Wood's test functions

We define smooth functions to Simon Wood's test datasets [Woo11] as

$$f_0 = 5\sin(2\pi x),\ f_1 = \exp(3x) - 7,\ f_2 = 0.5 \times x^{11}(10(1-x))^6 - 10(10x)^3(1-x)^{10},$$
$$f_3 = 15\exp(-5|x - 1/2|) - 6,$$
$$f_4 = 2 - 1_{(x<=1/3)}(6x)^3 - 1_{(x>=2/3)}(6-6x)^3 - 1_{(2/3>x>1/3)}(8 + 2\sin(9(x-1/3)\pi)),$$
$$f_5(x) = \lfloor 20x \rfloor - 10,\ f_6(x) = 10 - \lceil 20x \rceil,\ f_7(x) = \sin(50x) + 10x - 10,$$
$$f_8(x) = 8 + 2\cos(50x) - 50x(1-x),\ f_9(x) = \lceil 50x(1-x) \rceil - 5,$$
$$f_{10}(x) = 5\log(x + 10^{-6}) + 5,\ f_{11}(x) = -10 - 5\log(x + 10^{-6}) + \sin(50x),$$
$$f_{12}(x) = 2\log(x + 10^{-6}) - 2\log(1 - x + 10^{-6}),\ f_{13}(x) = 10|\sin(20x)|,$$
$$f_{14}(x) = 1_{(x<=1/2)} \times 5\sin(20x) + 1_{(x>1/2)} \times (5\sin(10) + (\exp(5(x-0.5)) - 1)).$$

SPLIT INTO MORE THAN TWO SEGMENTS

Our approach can easily deal with multiway splits at tree nodes.
The linear predictor

$$\eta_i = \theta_{L_1} \times 1_{\{i \in L_1(j,s)\}} + \cdots + \theta_{L_m} \times 1_{\{i \in L_m(j,s)\}}, \tag{6}$$

where $L_k(j, s)$ is the k-th leaf subset resulting from the split.

For numeric partitioning variables, $L_1 \cup \cdots \cup L_m$ is a partition of the interval $[\min_i z_{i,j}, \max_i z_{i,j}]$, while for categorical partitioning variables, it is a partition of the modalities set $\{v_{j,1}, \ldots, v_{j,l_j}\}$.

The MLE $\widehat{\boldsymbol{\theta}}(s)$ depends only on the link function g and is given by

$$\hat{\theta}_{L_k}(s) = g\left(\overline{y}_{j^\star}^{L_k}(s)\right), \; m_{j^\star}^{L_k}(s) = \sum_{i=1}^{n} 1_{\{i \in L_k(j^\star, s)\}},$$

$$\overline{y}_{j^\star}^{L_k}(s) = \frac{1}{m_{j^\star}^{L_k}(s)} \sum_{i=1}^{n} y_i 1_{\{i \in L_k(j^\star, s)\}}.$$

The objective function (4) is generalized to

$$O(\overline{\boldsymbol{y}_{j^\star}}(s), \boldsymbol{m}_{j^\star}(s)) = \sum_{k=1}^{m} \tilde{b}\left(\overline{y}_{j^\star}^{L_k}(s)\right) m_{j^\star}^{L_k}(s) \overline{y}_{j^\star}^{L_k}(s) - \sum_{k=1}^{m} b\left(\tilde{b}(\overline{y}_{j^\star}^{L_k}(s))\right) m_{j^\star}^{L_k}(s). \tag{7}$$

Other special cases for log transformed variable

We consider the transformation $t(x) = \log(d_1 x + d_2)$ and denote by $T_i = t(Y_i)$ the transformed random variables, where d_1, d_2 are known parameters.

We assume that $T_1, \ldots, T_n$ are independent random variables with a distribution in the exponential family.

The log-likelihood only differs by a new $\tilde{c}$ function

$$\tilde{c}(y, \phi) = c(y, \phi) + \log\left(\frac{d_1}{d_1 y + d_2}\right),$$

whereas a and b remain identical to the original distribution.

As for non-transformed responses, the fitted log-like-lihood is explicit so that

$$\begin{aligned} O(\overline{\boldsymbol{t}_{j\star}}(s), \boldsymbol{m}_{j\star}(s)) = \tilde{b}\left(\bar{t}^L_{j\star}(s)\right) m^L_{j\star}(s)\bar{t}^L_{j\star}(s) - b\left(\tilde{b}(\bar{t}^L_{j\star}(s))\right) m^L_{j\star}(s) \\ + \tilde{b}\left(\bar{t}^R_{j\star}(s)\right) m^R_{j\star}(s)\bar{t}^R_{j\star}(s) - b\left(\tilde{b}(\bar{t}^R_{j\star}(s))\right) m^R_{j\star}(s). \end{aligned} \tag{8}$$

	Left node	Right node
Average	$\bar{t}^L_j(s) = \frac{1}{m^L_j(s)} \sum_{i=1}^n t(y_i) 1_{\{i \in L(j,s)\}}$	$\bar{t}^R_j(s) = \frac{1}{m^R_j(s)} \sum_{i=1}^n t(y_i) 1_{\{i \in R(j,s)\}}$

Table 12: Notations for conditional average for transform $t(x)$ with frequencies given in Table 2

COMPLEXITY FOR BOSTONHOUSING AND HITTERS DATASETS

			Complexity		Computation time	
data	method	family	mean	median	mean	median
BostonHousing	ctree Bonferroni	Gaussian	21.90	22.0	0.359	0.363
	ctree Teststatistic	Gaussian	40.91	41.0	0.523	0.525
	ctree Univariate	Gaussian	36.21	36.5	0.493	0.494
	explicit GLM tree	Gaussian	14.15	14.0	1.330	1.343
	GLM tree	Gaussian	14.15	14.0	2.171	2.216
	explicit GLM tree	Gamma	13.82	14.0	1.752	1.742
	GLM tree	Gamma	13.82	14.0	4.678	4.684
	explicit GLM tree	inverse Gaussian	14.09	14.0	1.462	1.464
	GLM tree	inverse Gaussian	14.09	14.0	4.868	4.875
	lmtree	Gaussian	13.70	14.0	0.662	0.676
	rpart	Gaussian	9.01	9.0	0.036	0.038
Hitters	ctree Bonferroni	Gaussian	9.35	9.0	0.209	0.211
	ctree Teststatistic	Gaussian	21.77	22.0	0.333	0.333
	ctree Univariate	Gaussian	18.43	19.0	0.301	0.311
	explicit GLM tree	Gaussian	7.08	7.0	0.475	0.484
	GLM tree	Gaussian	7.08	7.0	0.701	0.720
	explicit GLM tree	Gamma	6.50	6.5	0.604	0.597
	GLM tree	Gamma	6.50	6.5	1.430	1.435
	explicit GLM tree	inverse Gaussian	6.56	6.0	0.550	0.545
	GLM tree	inverse Gaussian	6.56	6.0	1.514	1.521
	lmtree	Gaussian	6.06	6.0	0.265	0.254
	rpart	Gaussian	9.47	9.0	0.030	0.030

PERFORMANCE ON BOSTONHOUSING AND HITTERS DATASETS

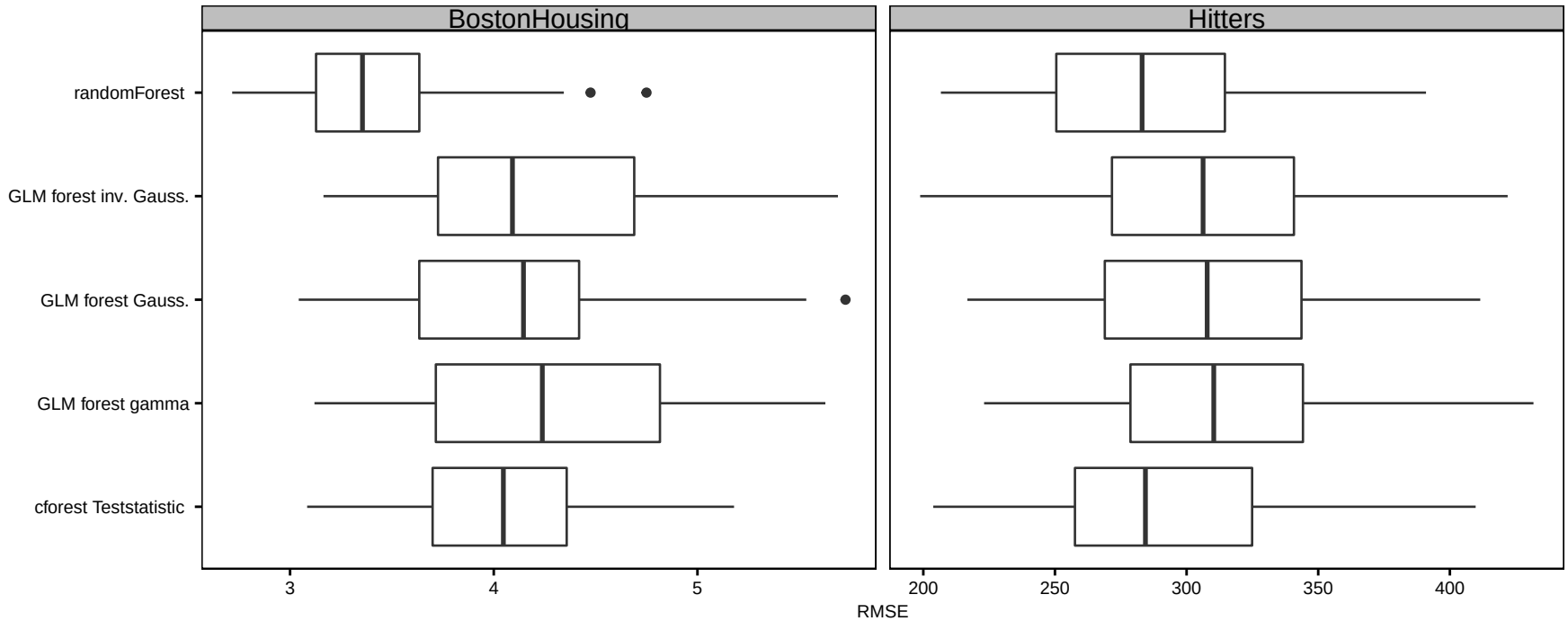


FIGURE 9: Predictive RMSE with 100 bootstrap replications for `BostonHousing` and `Hitters`