# Generalized Additive Models via Direct Optimization of Regularized Decision Stump Forests

**Magzhan Gabidolla** and **Miguel Á. Carreira-Perpiñán**, EECS, UC Merced

ICML International Conference On Machine Learning

## 1 Introduction

- Generalized Additive Models (GAMs) are an important model class in interpretable machine learning.
- One effective approach in learning these models is through an ensemble of decision stumps.
- Since each stump depends on a single feature, stumps using the same feature $x_d$ can be grouped to define a *shape function* $f_d(x_d)$ for this feature.
- Let $s(\mathbf{x};\boldsymbol{\theta})$: $\mathbb{R}^D \to \mathbb{R}$ be a decision stump with 4 learnable parameters: $\boldsymbol{\theta} = \{\phi, \tau, \mu^l, \mu^r\}$. $\phi \in \{1,\dots,D\}$ is a feature index to split, $\tau \in \mathbb{R}$ is a threshold value, $\mu^l, \mu^r \in \mathbb{R}$ are the left and right leaf prediction values:

$$s(\mathbf{x};\boldsymbol{\theta}) = \begin{cases} \mu^l, & \text{if } x_\phi < \tau \\ \mu^r, & \text{if } x_\phi \geq \tau. \end{cases}$$

- A stump forest $F(\mathbf{x};\boldsymbol{\Theta})$ is defined as a sum of $T$ stumps plus a bias term $\mu$:

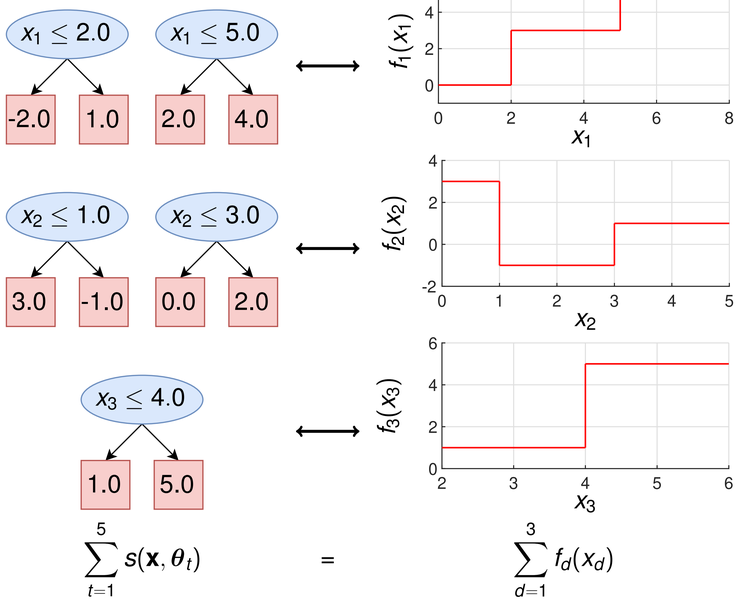
$$F(\mathbf{x};\boldsymbol{\Theta}) = \mu + \sum_{t=1}^{T} s(\mathbf{x};\boldsymbol{\theta}_t).$$

- Regrouping the stumps by feature indices, we obtain an additive model:

$$F(\mathbf{x};\boldsymbol{\Theta}) = \mu + \sum_{t=1}^{T} s(\mathbf{x};\boldsymbol{\theta}_t) = \mu + \sum_{d=1}^{D} \sum_{t:\ \phi_t=d} s(\mathbf{x};\boldsymbol{\theta}_t) = \mu + \sum_{d=1}^{D} f_d(x_d),$$

where $f_d(x_d) = \sum_{t:\ \phi_t=d} s(\mathbf{x};\boldsymbol{\theta}_t)$.

- Illustration on a toy example:




## 2 Alternating Optimization

- Consider a regression problem with a squared error loss:

$$\min_{\boldsymbol{\Theta}} \frac{1}{2}\sum_{n=1}^{N}[y_n - F(\mathbf{x}_n;\boldsymbol{\Theta})]^2.$$

- We apply alternating optimization with the following steps:
- **Individual stumps.** The optimization problem over a given stump $s(\cdot;\boldsymbol{\theta}_t)$ when others are fixed is:

$$\min_{\boldsymbol{\theta}_t} \frac{1}{2}\sum_{n=1}^{N}[y_n - \sum_{u\neq t} s(\mathbf{x}_n;\boldsymbol{\theta}_u) - s(\mathbf{x}_n;\boldsymbol{\theta}_t)]^2$$

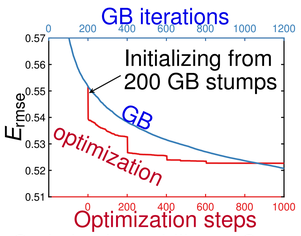
This is a standard regression problem over a stump but with targets corresponding to the residuals. It can be solved exactly through enumeration over each (feature, threshold) pair as in traditional decision tree algorithms.

- **All leaf parameters.** Once the splits $\{\phi_t, \tau_t\}_{t=1}^T$ are fixed, the problem simplifies to a linear regression on features corresponding to stump partitions. To see this, we can rewrite the predictive function of a stump using an indicator function: $s(\mathbf{x};\boldsymbol{\theta}) = \mu^l\, I(x_\phi < \tau) + \mu^r\, I(x_\phi \geq \tau)$. With this notation, the objective function over all leaves is:

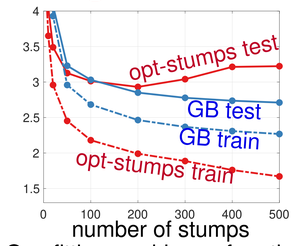
$$\min_{\mu,\{\mu_t^l,\mu_t^r\}_{t=1}^T} \frac{1}{2}\sum_{n=1}^{N}\left[y_n - \sum_{t=1}^{T} \mu_t^l\, I(x_{n\phi_t} < \tau_t) + \mu_t^r\, I(x_{n\phi_t} \geq \tau_t)\right]^2.$$

Since in this step all the splits $\{\phi_t, \tau_t\}_{t=1}^T$ are fixed, the indicator functions $I(\cdot)$ are just constants multiplying the unknowns $\{\mu_t^l, \mu_t^r\}_{t=1}^T$ that appear linearly inside the squared error. And so, eq. 1 is a simple linear regression problem on features induced by the stump splits, and can also be solved exactly and efficiently.

## 3 Overfitting and Regularization




Optimization ability. Comparison of FAO against GB for stump forests in train error for the California Housing dataset.




Overfitting problem of optimized stump forests on the cpuact dataset. Here GB avoids overfitting by using a learning rate 0.3.

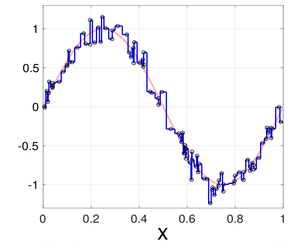
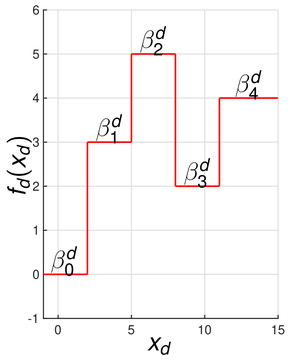



Illustration on a simple 1D regression problem with 100 stumps. Ground truth targets are from a sine function plotted in light red.

- We apply an $\ell_1$ discontinuity penalty to shape functions:

$$r\left(\{\tau_t, \mu_t^l, \mu_t^r\}_{t:\ \phi_t=d}\right) = \sum_{t=0}^{T_d-1} \left|\beta_{t+1}^d - \beta_t^d\right|$$

where $\beta_t$ is the value of the $t$-th constant piece from the left, and $T_d$ is the number of constant pieces for feature $d$. This penalizes the difference between adjacent constant piece values.




- We also propose is to penalize the deviation from the bias for each leaf value:

$$\sum_{t=1}^{T}(\mu_t^l - \mu)^2 + (\mu_t^r - \mu)^2.$$

- With these two types of regularization, our final objective function is:

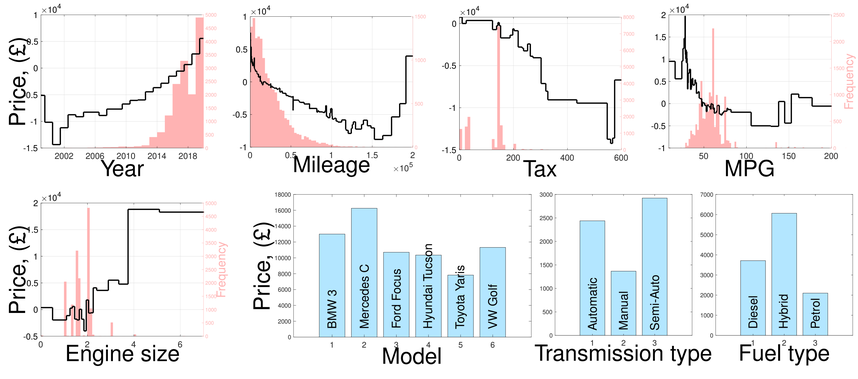
$$\min_{\boldsymbol{\Theta}} \frac{1}{2}\sum_{n=1}^{N}[y_n - F(\mathbf{x}_n;\boldsymbol{\Theta})]^2 + \lambda \sum_{d=1}^{D} r(\{\tau_t, \mu_t^l, \mu_t^r\}_{t:\ \phi_t=d}) + \alpha \sum_{t=1}^{T}\left((\mu_t^l - \mu)^2 + (\mu_t^r - \mu)^2\right)$$

## 4 Experiments, comparison

Table: RMSE error. Green color is the best test error, and blue is the second best.

| Dataset | | Ours | GB | EBM | Splines | FastSparse |
|---|---|---|---|---|---|---|
| **Cpuact** | train | 2.12±0.01 | 2.20±0.04 | 2.19±0.02 | 2.53±0.02 | 2.76±0.03 |
| $N$=8.2$k$ | test | 2.37±0.03 | 2.43±0.06 | 2.50±0.05 | 2.69±0.06 | 2.91±0.17 |
| $D$=21 | size | 642±0 | 3.4k±133 | 16.6k±36 | 271±3 | 119±4 |
| | time (s) | 9.4±0.3 | 46±17 | 39±2 | 37±0.03 | 3.8±0.5 |
| **Wine** | train $\times 10^{-2}$ | 65.70±0.15 | 68.13±0.27 | 66.73±0.27 | 67.99±0.29 | 68.01±0.25 |
| $N$=6.5$k$ | test $\times 10^{-2}$ | 70.02±0.66 | 70.92±0.51 | 70.12±0.39 | 71.79±1.40 | 71.77±0.63 |
| $D$=11 | size | 724±12 | 770±32 | 3.9k±11 | 197±7 | 182±4 |
| | time (s) | 6.0±0.3 | 2.87±0.58 | 4.44±1.33 | 56±16 | 0.57±0.07 |
| **Housing** | train $\times 10^{-2}$ | 51.84±0.16 | 54.24±0.27 | 52.70±0.04 | 53.37±0.21 | 54.62±0.20 |
| $N$=21$k$ | test $\times 10^{-2}$ | 54.80±0.65 | 56.15±0.58 | 55.23±0.68 | 55.49±0.61 | 56.29±0.65 |
| $D$=8 | size | 1.4k±20 | 2.4k±31 | 7.2k±8 | 528±2 | 579±9 |
| | time (s) | 13.6±0.4 | 42±8 | 36±2 | 37±2 | 3.94±0.73 |
| **Diamond** | train $\times 10^{2}$ | 9.95±0.02 | 10.07±0.05 | 10.11±0.03 | 10.02±0.02 | 10.01±0.02 |
| $N$=54$k$ | test $\times 10^{2}$ | 10.15±0.08 | 10.19±0.08 | 10.23±0.06 | 10.96±1.45 | 10.17±0.09 |
| $D$=26 | size | 934±16 | 1182±81 | 3.4k±7 | 273±24 | 516±11 |
| | time (s) | 25.1±0.9 | 140±58 | 20±2 | 42±0.4 | 45±10 |

## 5 Experiments, interpretability




Visualization of the resulting GAM on the UK used car price prediction dataset. For the numerical features, the light red bars show the histogram of the training points with the frequency values given on the right $y$-axis.