Boosting

Richard Alligier

Ecole Nationale de l'Aviation Civile
Office 005, Building Z
richard.alligier@enac.fr

Outline of this session

1. Introduction to Boosting
2. AdaBoost [Freund and Schapire, 1997]
3. Gradient Boosting [Friedman, 2001]
4. Gradient Boosted Trees [Friedman, 2001]

Outline of this session

Boosting: meta-algorithm using "weak" learning algorithm

Goal
Build an accurate model using a "weak" learning algorithm $\mathcal{A}$

"Weak" learning algorithm
Learn a model that can do slightly better than a random guess

Learning algorithm with high bias (and low variance)

Idea
Sequentially apply $\mathcal{A}$ on a repeatedly modified data to incrementally refine the model we build

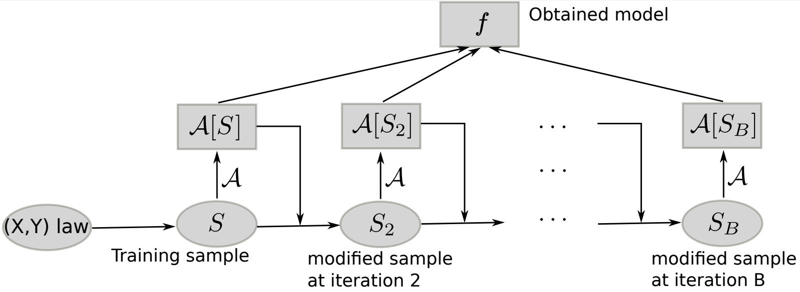


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} \left(y - \phi(x;\gamma)\right)^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)}\mathbb{1}_{x\leqslant\gamma^{(0)}}(x) + \gamma^{(2)}\mathbb{1}_{x>\gamma^{(0)}}(x)$

$$\begin{tikzcd} & \boxed{x \leqslant \gamma^{(0)}} \arrow[dl, "\text{true}"'] \arrow[dr, "\text{false}"] & \\ \gamma^{(1)} & & \gamma^{(2)} \end{tikzcd}$$

Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\displaystyle\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

$\boxed{x \leqslant \gamma^{(0)}}$

true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

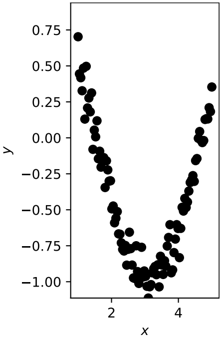


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$$\boxed{x \leqslant \gamma^{(0)}}$$
true — $\gamma^{(1)}$; false — $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

Model: $\phi(x;\gamma_1)$

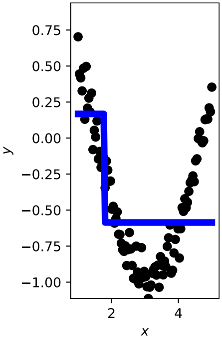


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$$\boxed{x \leqslant \gamma^{(0)}}$$
true $\gamma^{(1)}$ false $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)}\mathbb{1}_{x\leqslant\gamma^{(0)}}(x) + \gamma^{(2)}\mathbb{1}_{x>\gamma^{(0)}}(x)$

Model: $\color{blue}{\phi(x;\gamma_1)}+\color{red}{\phi(x;\gamma_2)}$

Idea

Add a corrective term $\color{red}{\phi(x,\gamma_2)}$

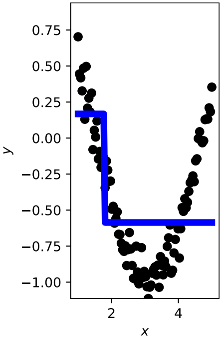


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$x \leqslant \gamma^{(0)}$

true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x\leqslant\gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x>\gamma^{(0)}}(x)$

Model: $\phi(x;\gamma_1)+\phi(x;\gamma_2)$

Idea

Add a corrective term $\phi(x,\gamma_2)$

Choose $\phi(x,\gamma_2)$ minimizing:

$$\sum_{(x,y)\in S} (y - \phi(x;\gamma_1) - \phi(x;\gamma_2))^2$$

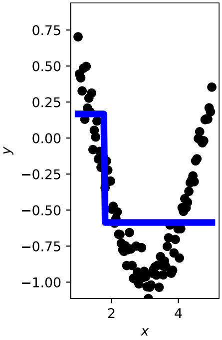


Illustration of Boosting Approach on Univariate Regression

“Weak” Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

$\boxed{x \leqslant \gamma^{(0)}}$

true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Model: $\color{blue}{\phi(x;\gamma_1)}+\color{red}{\phi(x;\gamma_2)}$

Idea

Add a corrective term $\color{red}{\phi(x,\gamma_2)}$

Choose $\color{red}{\phi(x,\gamma_2)}$ minimizing:

$$\sum_{(x,y)\in S} (\color{orange}{y - \phi(x;\gamma_1)} - \color{red}{\phi(x;\gamma_2)})^2$$

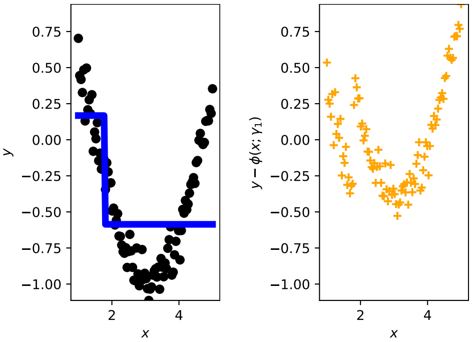


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$$\boxed{x \leqslant \gamma^{(0)}}$$
true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

Model: $\phi(x;\gamma_1) + \phi(x;\gamma_2)$

Idea

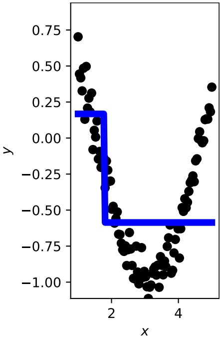
Add a corrective term $\phi(x,\gamma_2)$

Choose $\phi(x,\gamma_2)$ minimizing:

$$\sum_{(x,y)\in S} (y - \phi(x;\gamma_1) - \phi(x;\gamma_2))^2$$

Use $\mathcal{A}$ on S_2 to get $\phi(x;\gamma_2)$:

$S_2 = \{(x, y - \phi(x;\gamma_1)) \,|\, (x,y) \in S\}$



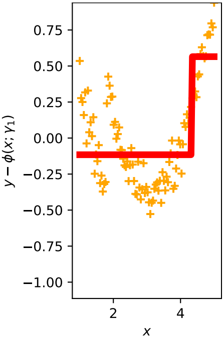


Illustration of Boosting Approach on Univariate Regression

“Weak” Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$\boxed{x \leqslant \gamma^{(0)}}$

true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x\leqslant\gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x>\gamma^{(0)}}(x)$

Model: $\phi(x;\gamma_1)+\phi(x;\gamma_2)$

Idea

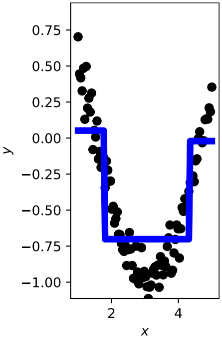
Add a corrective term $\phi(x,\gamma_2)$

Choose $\phi(x,\gamma_2)$ minimizing:

$$\sum_{(x,y)\in S} (y - \phi(x;\gamma_1) - \phi(x;\gamma_2))^2$$

Use $\mathcal{A}$ on S_2 to get $\phi(x;\gamma_2)$:

$S_2 = \{(x, y - \phi(x;\gamma_1)) \,|(x,y) \in S\}$



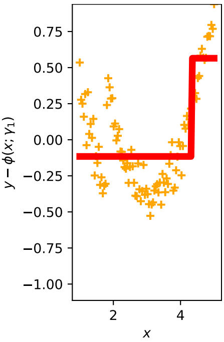


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$$\boxed{x \leqslant \gamma^{(0)}}$$
true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

Model: $\color{blue}{\phi(x;\gamma_1)}+\color{blue}{\phi(x;\gamma_2)}+\color{red}{\phi(x;\gamma_3)}$

Idea

Add a corrective term $\color{red}{\phi(x,\gamma_3)}$

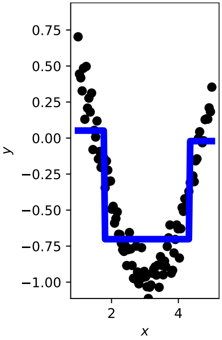


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

$\boxed{x \leqslant \gamma^{(0)}}$

true / \ false

$\gamma^{(1)}$ $\gamma^{(2)}$

Model: $\color{blue}{\phi(x;\gamma_1)} + \color{blue}{\phi(x;\gamma_2)} + \color{red}{\phi(x;\gamma_3)}$

$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

Add a corrective term $\color{red}{\phi(x,\gamma_3)}$

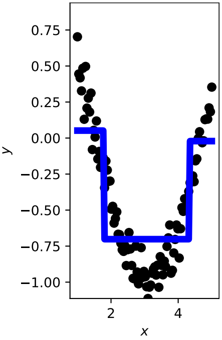
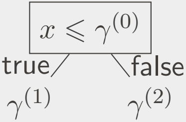


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$



Stump: $\phi(x;\gamma) = \gamma^{(1)}\mathbb{1}_{x\leqslant\gamma^{(0)}}(x) + \gamma^{(2)}\mathbb{1}_{x>\gamma^{(0)}}(x)$

Model: ${\color{blue}f_2(x)}+{\color{red}\phi(x;\gamma_3)}$

$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

Add a corrective term ${\color{red}\phi(x,\gamma_3)}$

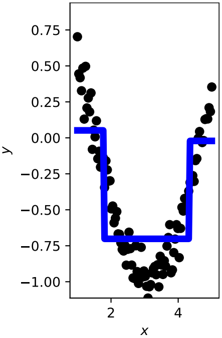


Illustration of Boosting Approach on Univariate Regression

“Weak” Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

$\boxed{x \leqslant \gamma^{(0)}}$

true / \ false

$\gamma^{(1)}$ $\gamma^{(2)}$

Model: $f_2(x) + \phi(x;\gamma_3)$

$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

Add a corrective term $\phi(x,\gamma_3)$

Choose $\phi(x,\gamma_3)$ minimizing:

$$\sum_{(x,y)\in S} (y - f_2(x) - \phi(x;\gamma_3))^2$$

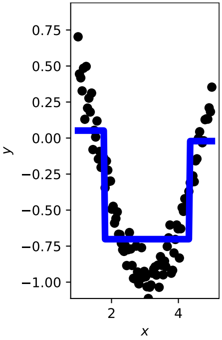


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$\boxed{x \leqslant \gamma^{(0)}}$

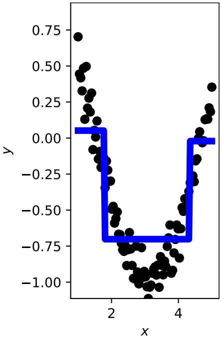
true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x\leqslant\gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x>\gamma^{(0)}}(x)$

Model: $f_2(x) + \phi(x;\gamma_3)$
$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

Add a corrective term $\phi(x,\gamma_3)$
Choose $\phi(x,\gamma_3)$ minimizing:
$$\sum_{(x,y)\in S} (y - f_2(x) - \phi(x;\gamma_3))^2$$



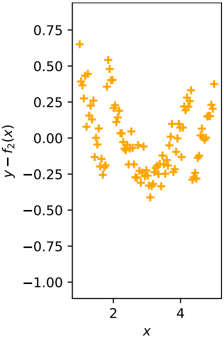


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\displaystyle\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

$$\begin{array}{ccc} & \boxed{x \leqslant \gamma^{(0)}} & \\ \text{true} \swarrow & & \searrow \text{false} \\ \gamma^{(1)} & & \gamma^{(2)} \end{array}$$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

Model: $f_2(x) + \phi(x;\gamma_3)$
$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

Add a corrective term $\phi(x, \gamma_3)$
Choose $\phi(x, \gamma_3)$ minimizing:
$$\sum_{(x,y)\in S} (y - f_2(x) - \phi(x;\gamma_3))^2$$
Use $\mathcal{A}$ on S_3 to get $\phi(x;\gamma_2)$:
$S_3 = \{(x, y - f_2(x)) \,|\, (x,y) \in S\}$

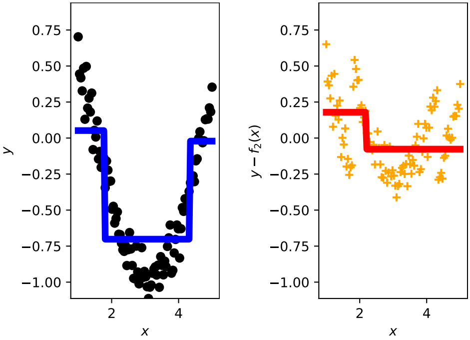
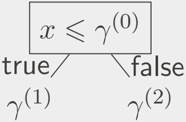


Illustration of Boosting Approach on Univariate Regression

“Weak” Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:
$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$



Model: $f_2(x) + \phi(x;\gamma_3)$
$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

Add a corrective term $\phi(x,\gamma_3)$
Choose $\phi(x,\gamma_3)$ minimizing:

$$\sum_{(x,y)\in S} (y - f_2(x) - \phi(x;\gamma_3))^2$$

Use $\mathcal{A}$ on S_3 to get $\phi(x;\gamma_2)$:
$S_3 = \{(x, y - f_2(x)) \,|(x,y) \in S\}$

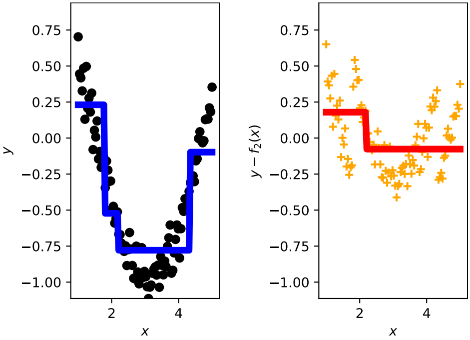


Illustration of Boosting Approach on Univariate Regression

"Weak" Learning Algorithm Used in This Illustration

We have a learning algorithm $\mathcal{A}$ such that:

$\forall S,\ \mathcal{A}[S]$ returns γ minimizing $\sum_{(x,y)\in S} (y - \phi(x;\gamma))^2$

Stump: $\phi(x;\gamma) = \gamma^{(1)} \mathbb{1}_{x \leqslant \gamma^{(0)}}(x) + \gamma^{(2)} \mathbb{1}_{x > \gamma^{(0)}}(x)$

$\boxed{x \leqslant \gamma^{(0)}}$

true: $\gamma^{(1)}$ false: $\gamma^{(2)}$

Model: $f_3(x)$

$f_B(x) = \sum_{b=1}^{B} \phi(x;\gamma_b)$

Idea

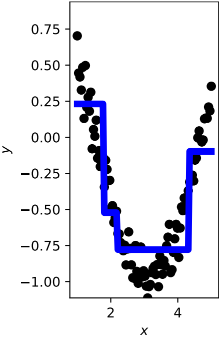
Add a corrective term $\phi(x,\gamma_3)$

Choose $\phi(x,\gamma_3)$ minimizing:

$$\sum_{(x,y)\in S} (y - f_2(x) - \phi(x;\gamma_3))^2$$

Use $\mathcal{A}$ on S_3 to get $\phi(x;\gamma_2)$:

$S_3 = \{(x, y - f_2(x)) \,|\, (x,y) \in S\}$



Forward Stagewise Additive Model (FSAM)

Idea

At each step, sum a model correcting the error made

$$f_B(x) = \sum_{b=1}^{B} \phi(x; \gamma_b)$$

where $x \mapsto \phi(x; \gamma)$ is a "simple" model parametrized by γ

Algorithm 1 Forward Stagewise Additive Model: Incremental Tuning

1: Initialize $f_0 : \mathrm{x} \to 0$
2: **for** $b = 1$ to B **do**
3: Compute parameters γ_b minimizing loss ℓ :
$\gamma_b = \arg\min_\gamma \sum_{i=1}^{N} \ell\left(\mathrm{y}_i, f_{b-1}(\mathrm{x}_i) + \phi(\mathrm{x}_i; \gamma)\right)$ ▷ use of $\mathcal{A}$
4: $f_b : \mathrm{x} \to f_{b-1}(\mathrm{x}) + \phi(\mathrm{x}; \gamma_b)$
5: **end for**

Greedy approach:
The parameters obtained $(\gamma_1, \ldots, \gamma_B)$ gives a low value for ℓ, not the lowest one.

Differences and Similarity Between Bagging and Boosting

Similarity

Both are sums of models fitted using a learning algorithm $\mathcal{A}$

Differences

Bagging:

- Reduces the variance
- Models can be trained in parallel
- Each model solves same ML problem
- Large B reduces overfit but $\lim_{B \to +\infty} \frac{1}{B} \sum_{b=1}^{B} \mathcal{A}[S_b]$ can overfit

Boosting:

- Reduces the bias
- Models trained sequentially
- Each model trained on $\neq$ problem
- Large B leads to overfit

Choice of $\mathcal{A}/\phi(.;\gamma)$ for Boosting ?

Use Boosting with Low Bias Algorithm is Not a Good Idea?

- High bias learning algorithm are usually faster
- Choosing the number of models B (fitted by a high bias algorithm) can be used to control overfit

Very Common $\rightarrow$ **Tree**

- Stump or more generally trees with a small number nodes
- In combination with Gradient Boosting, very efficient method

Not Very Common

- Splines: piecewise polynomial with continuity constraints
- Shallow neural network ([Badirli et al., 2020])
- Linear models
  - Boosted linear models are linear models
  - Full linear model **not** useful for quadratic loss
  - Univariate linear model with most correlated: behavior close to LASSO

Additive Model ANOVA Expansion

Let us consider an input $x \in \mathbb{R}^P$ with P features $x_{.,1}, \dots, x_{.,P}$
The additive model is:
$f_B(x) = \sum_{b=1}^{B} \phi(x; \gamma_b)$

Boosted Model Using Stumps

$f_B(x) = \sum_{p=1}^{P} \eta_p(x_{.,p})$
No non-linear interaction between two different features!

Additive Model ANOVA Expansion

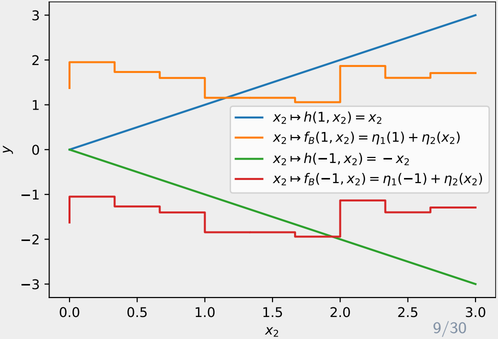
Let us consider an input $x \in \mathbb{R}^P$ with P features $x_{.,1}, \ldots, x_{.,P}$
The additive model is:

$$f_B(x) = \sum_{b=1}^{B} \phi(x; \gamma_b)$$

Boosted Model Using Stumps

$f_B(x) = \sum_{p=1}^{P} \eta_p(x_{.,p})$
No non-linear interaction between two different features!

True model:
$h(x_1, x_2) = x_1 x_2$
Boosted model:
$f_B(x_1, x_2) = \eta_1(x_1) + \eta_2(x_2)$



Additive Model ANOVA Expansion

Let us consider an input $x \in \mathbb{R}^P$ with P features $x_{.,1}, \ldots, x_{.,P}$

The additive model is:

$$f_B(x) = \sum_{b=1}^{B} \phi(x; \gamma_b)$$

Boosted Model Using Stumps

$f_B(x) = \sum_{p=1}^{P} \eta_p(x_{.,p})$

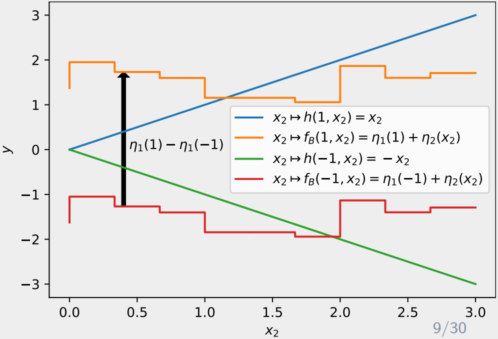
No non-linear interaction between two different features!

True model:

$h(x_1, x_2) = x_1 x_2$

Boosted model:

$f_B(x_1, x_2) = \eta_1(x_1) + \eta_2(x_2)$



Additive Model ANOVA Expansion

Let us consider an input $x \in \mathbb{R}^P$ with P features $x_{.,1}, \ldots, x_{.,P}$
The additive model is:
$f_B(x) = \sum_{b=1}^{B} \phi(x; \gamma_b)$

Boosted Model Using Stumps

$f_B(x) = \sum_{p=1}^{P} \eta_p(x_{.,p})$
No non-linear interaction between two different features!

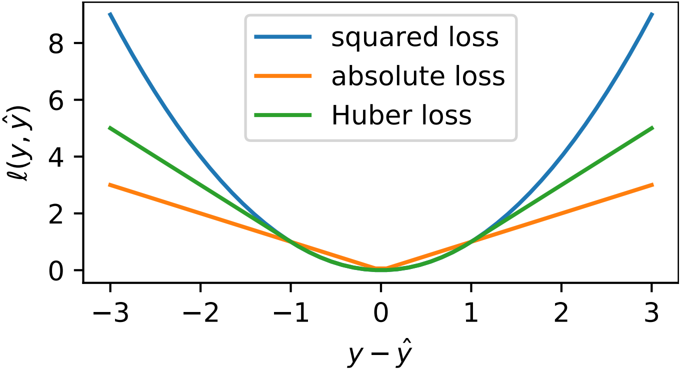
Boosted Model Using Small Trees with J Nodes

For each tree $\phi(x; \gamma_b)$ at most J features actually used
$f_B(x) = \sum_{p_1,\ldots,p_J} \eta_{(p_1,\ldots,p_J)}(x^{(p_1)}, \ldots, x^{(p_J)})$
At most J features can interact!

Choice of Loss ℓ ?

Losses for Regression

- Squared loss: $\ell(y, \hat{y}) = (y - \hat{y})^2 \to \arg\min_c \sum_{i=1}^{N} \ell(y_i, c) = \mathrm{mean}(y_{1\ldots N})$
- Absolute loss: $\ell(y, \hat{y}) = |y - \hat{y}| \to \arg\min_c \sum_{i=1}^{N} \ell(y_i, c) = \mathrm{median}(y_{1\ldots N})$
- Huber loss [1964]: $\ell(y, \hat{y}) = (y - \hat{y})^2$ if $|y - \hat{y}| \leqslant \delta$ else $2\delta|y - \hat{y}| - \delta^2$



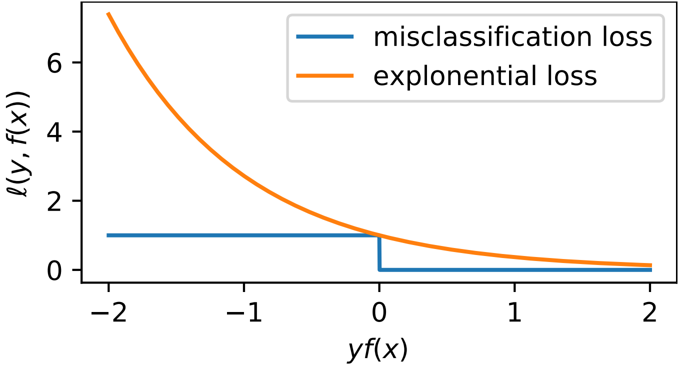
Choice of Loss ℓ ?

Losses for Binary Classification: $y \in \{-1, 1\}$

$\hat{y} = \mathrm{sign}(f(x))$, with $f(x) \in \mathbb{R}$

- Misclassification loss: $\ell(y, f(x)) = \mathbb{1}_{]-\infty;0]}(yf(x))$
- Exponential loss: $\ell(y, f(x)) = \exp(-yf(x))$

  $\rightarrow \arg\min_c \sum_{i=1}^{N} \ell(y_i, c) = \frac{1}{2} \log \frac{\text{Proportion}(y_i=1; i \in [\![1;N]\!])}{\text{Proportion}(y_i=-1; i \in [\![1;N]\!])}$



Outline of this session

AdaBoost = FSAM with Exponential Loss and Particular ϕ

Hypothesis

- Binary classification: $y \in \{-1; 1\}$; $\hat{y} = \operatorname{sign}\left(f\left(x\right)\right)$
- Exponential loss: $\ell(y, f(x)) = \exp\left(-y f(x)\right)$
- $\phi(x; \alpha, \beta) = \alpha\ h(x; \beta)$ with h classifier predicting -1 or 1

We incrementally build $f_B(x) = \sum_{b=1}^{B} \phi(x; \alpha_b, \beta_b) = \sum_{b=1}^{B} \alpha_b h(x; \beta_b)$

FSAM: $(\alpha_b, \beta_b) = \arg\min_{(\alpha,\beta)} \sum_{i=1}^{N} \ell\left(\mathrm{y}_i, f_{b-1}(\mathrm{x}_i) + \phi(\mathrm{x}_i; \alpha, \beta)\right)$

$$\sum_{i=1}^{N} e^{-\mathrm{y}_i (f_{b-1}(\mathrm{x}_i) + \phi(\mathrm{x}_i; \alpha, \beta))} = \sum_{i=1}^{N} \underbrace{e^{-\mathrm{y}_i f_{b-1}(\mathrm{x}_i)}}_{w_i} e^{-\mathrm{y}_i h(\mathrm{x}_i; \beta)\alpha}$$

$$= e^{-\alpha} \sum_{i=1}^{N} w_i + \left(e^{\alpha} - e^{-\alpha}\right) \underbrace{\sum_{i=1}^{N} w_i \mathbb{1}\left(y_i \neq h(x_i; \beta)\right)}_{\text{Minimized using } \mathcal{A}}$$

Nota Bene: $\mathcal{A}$ minimizes misclassification loss, not exponential loss

AdaBoost Algorithm [Freund and Schapire, 1997]
One of the First Boosting Algorithm (Gödel prize 2003)

"weak" learner $\mathcal{A}$ for classification

- input: $T = \{(x_i, y_i, w_i)\}_{i=1,\dots,N}$ with $y_i \in \{-1, 1\}$
- output: a classifier $\mathcal{A}[T]$ trying minimizing $\sum_{i=1}^{n} w_i \mathbb{1}\left(y_i \neq \mathcal{A}[T](x_i)\right)$

AdaBoost for binary classification (-1 or 1)

---

function $\textsc{AdaBoost}$(training set:$\{(x_i, y_i)\}_{i=1,\dots,N}$, weak learner: $\mathcal{A}$)
    $w_{1,\dots,N} \leftarrow 1$
    for $b = 1$ to B **do**
        $h_b \leftarrow \mathcal{A}[\{(x_i, y_i, w_i)\}_{i=1,\dots,N}]$
        $\mathrm{err}_b \leftarrow \sum_{i=1}^{N} \tilde{w}_i \mathbb{1}\left(h_b(x_i) \neq y_i\right)$ where $\tilde{w}_i = \frac{w_i}{\sum_{i=1}^{N} w_i}$
        $\alpha_b \leftarrow 0.5 \log \frac{1-\mathrm{err}_b}{\mathrm{err}_b}$
        For $i = 1, \dots, N$: $w_i \leftarrow w_i \exp\left(-\alpha_b y_i h_b(x_i)\right)$
    end for
    return $f : x \mapsto \mathrm{sign}\left(\sum_{b=1}^{B} \alpha_b h_b(x)\right)$
end function

Adaboost Illustration

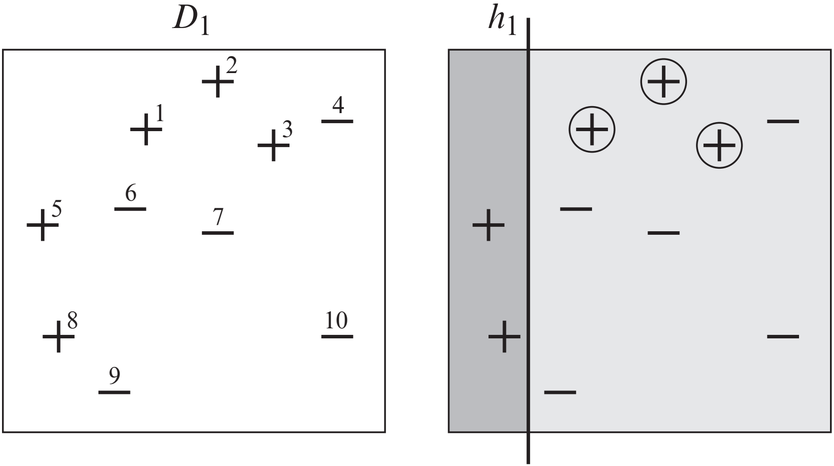


Figure from [Schapire and Freund, 2012]

Adaboost Illustration

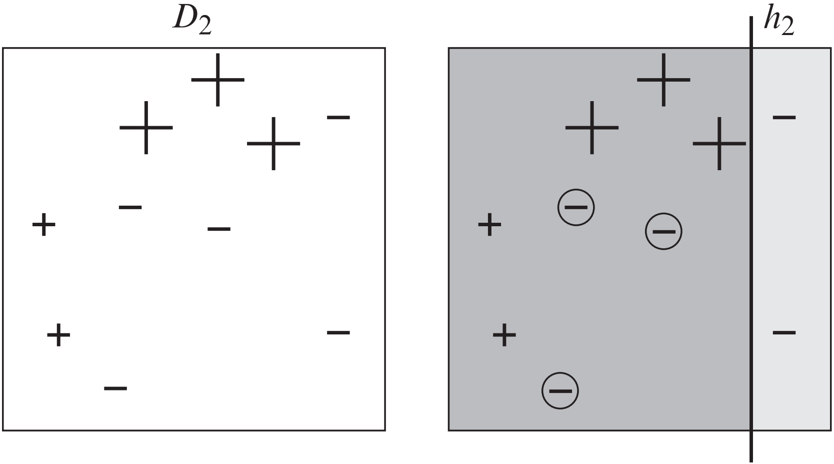


Figure from [Schapire and Freund, 2012]

Adaboost Illustration

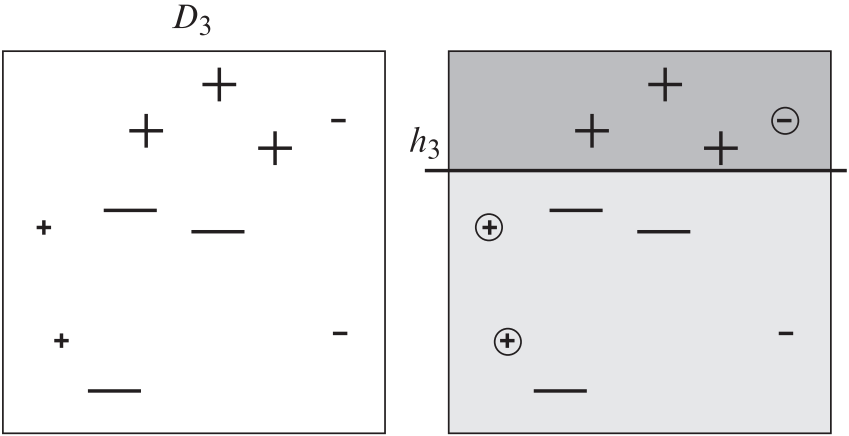


Figure from [Schapire and Freund, 2012]

Adaboost Illustration

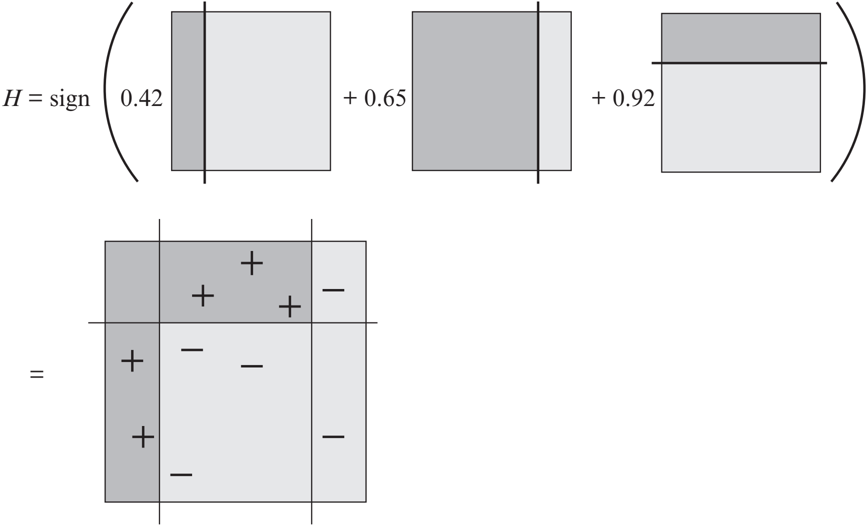


Figure from [Schapire and Freund, 2012]

Performance of Adaboost

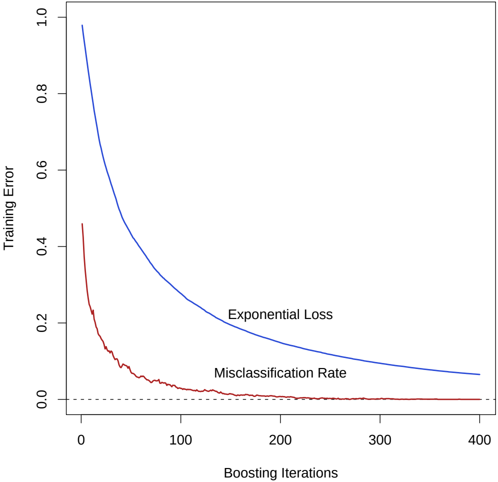


Figure from [Friedman et al., 2001]

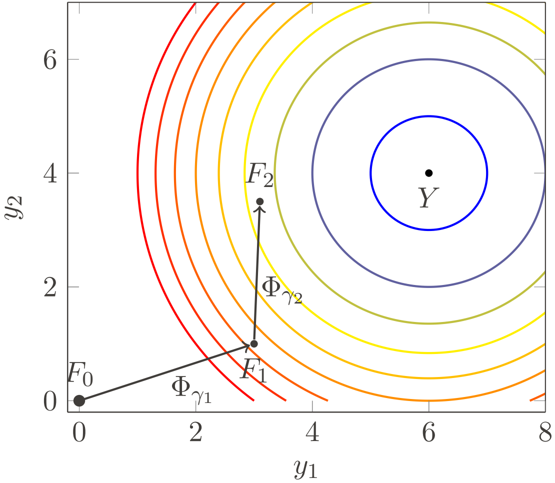
Conclusion on AdaBoost

- One of the first efficient boosting algorithm
- Easy to implement
- Only one hyperparameter B
- Sensitive to noise and outliers

Outline of this session

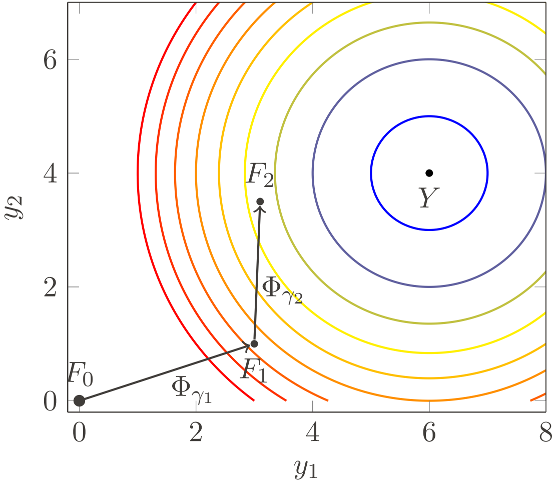
Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$



Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$

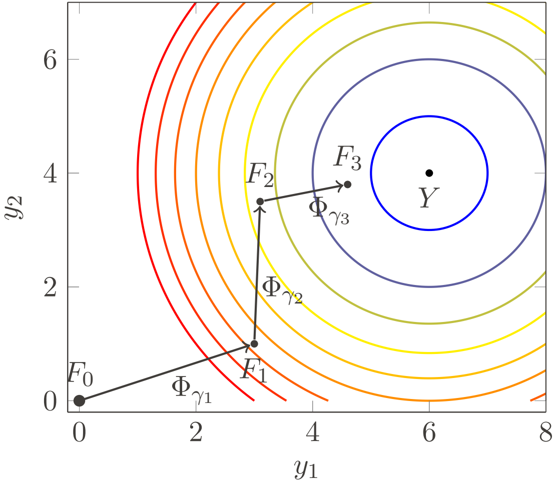


FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$

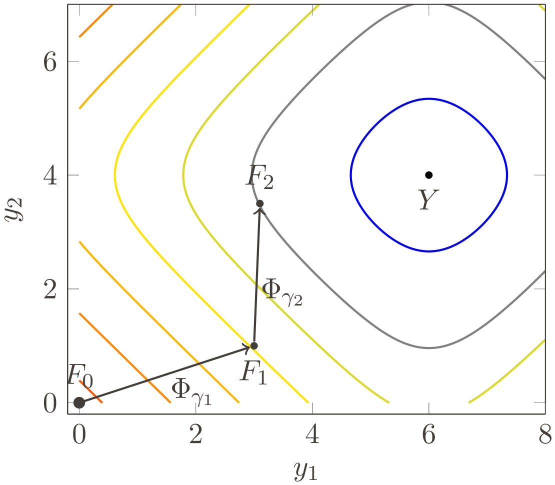


FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$

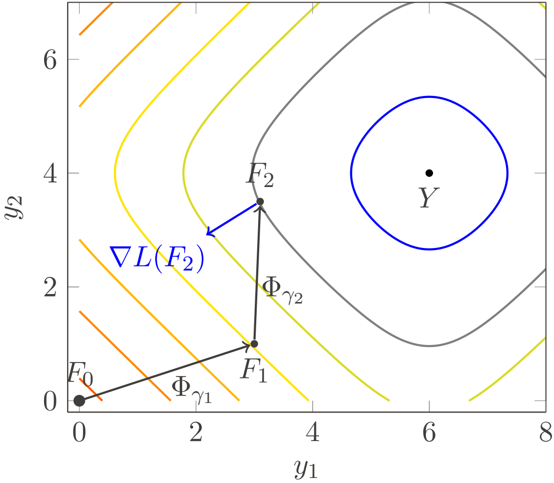


FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$

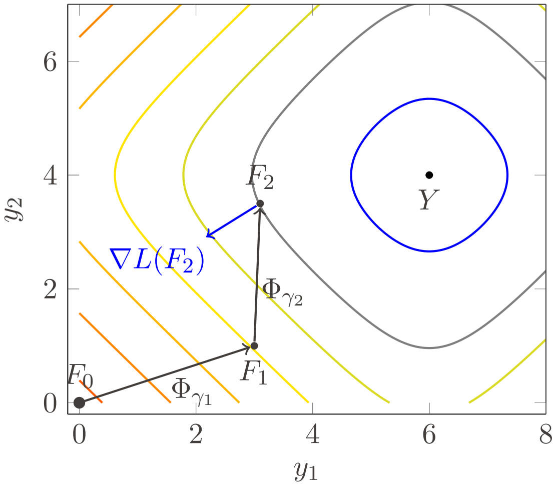


FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$



FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

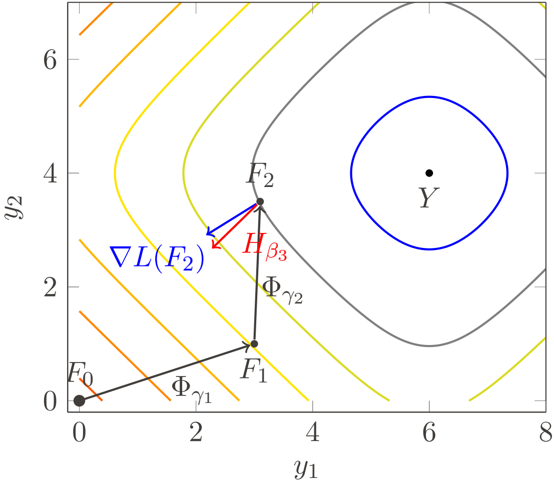
Gradient Boosting

Let us note $H_\beta = \begin{pmatrix} h(x_1;\beta) \\ \vdots \\ h(x_n;\beta) \end{pmatrix}$

$$\beta_b = \underset{\beta}{\operatorname{argmin}} \|\nabla L(F_{b-1}) - H_\beta\|^2$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \; \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix}, \; F_B = \sum_{b=1}^{B} \Phi_{\gamma_b}, \; L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$



FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}} \; L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

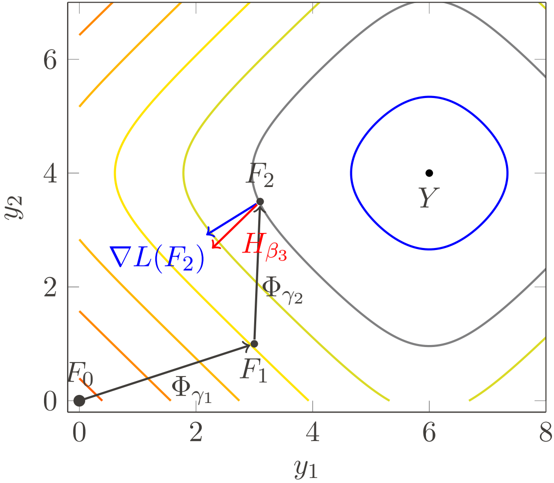
Gradient Boosting

Let us note $H_\beta = \begin{pmatrix} h(x_1;\beta) \\ \vdots \\ h(x_n;\beta) \end{pmatrix}$

$$\beta_b = \underset{\beta}{\operatorname{argmin}} \|\nabla L(F_{b-1}) - H_\beta\|^2$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$



FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

Gradient Boosting

Let us note $H_\beta = \begin{pmatrix} h(x_1;\beta) \\ \vdots \\ h(x_n;\beta) \end{pmatrix}$

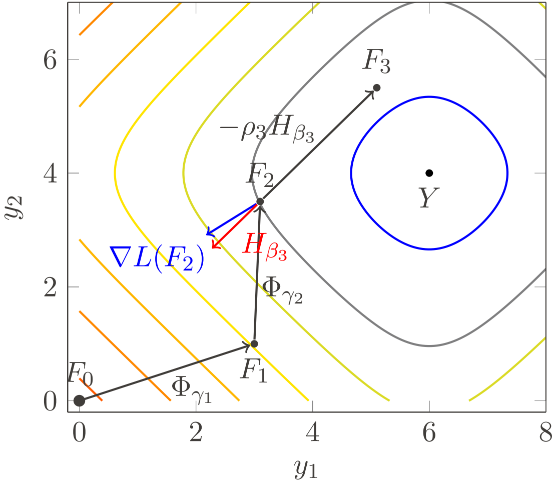
$$\beta_b = \underset{\beta}{\operatorname{argmin}} \|\nabla L(F_{b-1}) - H_\beta\|^2$$

$$\rho_b = \underset{\rho}{\operatorname{argmin}}\ L(F_{b-1} - \rho H_{\beta_b})$$

$$\Phi_{\gamma_b} = -\rho_b H_{\beta_b}$$

Visualization of Boosting

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix},\ \Phi_\gamma = \begin{pmatrix} \phi(x_1;\gamma) \\ \vdots \\ \phi(x_N;\gamma) \end{pmatrix},\ F_B = \sum_{b=1}^{B} \Phi_{\gamma_b},\ L\left(\hat{Y}\right) = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right)$$



FSAM

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}}\ L(\underbrace{F_{b-1} + \Phi_\gamma}_{F_b})$$

Gradient Boosting

Let us note $H_\beta = \begin{pmatrix} h(x_1;\beta) \\ \vdots \\ h(x_n;\beta) \end{pmatrix}$

$$\beta_b = \underset{\beta}{\operatorname{argmin}} \|\nabla L(F_{b-1}) - H_\beta\|^2$$

$$\rho_b = \underset{\rho}{\operatorname{argmin}}\ L(F_{b-1} - \rho H_{\beta_b})$$

$$\Phi_{\gamma_b} = -\rho_b H_{\beta_b}$$

Gradient Boosting [Friedman, 2001]

Computing the Model to Add in FSAM

$\gamma_b = \arg\min_{\gamma} \sum_{i=1}^{n} \ell\left(\mathrm{y}_i, f_{m-1}(\mathrm{x}_i) + \phi(\mathrm{x}_i; \gamma)\right)$

For some choice of loss ℓ and ϕ, can be a difficult problem

Computing the Model to Add in Gradient Boosting

Gradient Boosting $\simeq$ gradient descent in a functional space

compute direction: $\beta_b = \arg\min_{\beta} \sum_{i=1}^{n} \left(\left[\frac{\partial \ell(\mathrm{y}_i, \hat{y})}{\partial \hat{y}} \right]_{\hat{y} = f_{b-1}(\mathrm{x}_i)} - h(\mathrm{x}_i; \beta) \right)^2$

compute step:

$\rho_b = \arg\min_{\rho} \sum_{i=1}^{n} \ell\left(\mathrm{y}_i, f_{b-1}(\mathrm{x}_i) - \rho h(\mathrm{x}_i; \beta_b)\right)$

Update additive model:

$f_b : x \mapsto f_{b-1}(x) - \rho_b h(x; \beta_b)$

FSAM=GB for $\ell(y, \hat{y}) = 0.5(y - \hat{y})^2$

Outline of this session

Gradient Boosted Trees [Friedman, 2001]

FSAM with Trees

$\phi(x; R, c) = \sum_{m=1}^{M} c^{(m)} \mathbb{1}_{R^{(m)}}(x)$
$(R_b, c_b) = \underset{(R,c)}{\arg\min} \sum_{i=1}^{n} \ell\left(y_i, f_{b-1}(x_i) + \phi(x_i; R, c)\right)$

Can be Seen as a Two-Levels Optimization Problem:

1. Find regions R_b partitioning input space:
   Difficult combinatorial optimization problem, especially for robust loss
2. Find values c_b to predict for each region R_b:
   Easy problem even for robust loss

Idea: Use a mix of GB and FSAM in Two Steps

1. Find regions R_b using the direction as computed in Gradient Boosting:
   $$(R_b, \_) = \underset{(R,c)}{\arg\min} \sum_{i=1}^{n} \left(\left[\frac{\partial \ell(\mathrm{y}_i, \hat{y})}{\partial \hat{y}} \right]_{\hat{y} = f_{b-1}(\mathrm{x}_i)} - \phi(\mathrm{x}_i; R, c) \right)^2$$
2. Find values c_b using FSAM:
   $c_b = \underset{c}{\arg\min} \sum_{i=1}^{n} \ell\left(y_i, f_{b-1}(x_i) + \phi(x_i; R_b, c)\right)$

Gradient Boosted Trees [Friedman, 2001]

Gradient Boosted Trees is actually a mix of GB and FSAM

Gradient Boosted Trees

1: Initialize $f_0 : \mathrm{x} \to \arg\min_v \sum_{i=1}^n \ell(y_i, v)$ ▷ $\mathrm{mean}(y_{1,\dots,n})$ if quadratic loss
2: **for** $b = 1$ to B **do**
3: Compute regions R_b by fitting the gradient of ℓ on the training set:
$$(R_b, \_) = \arg\min_{(R,c)} \sum_{i=1}^n \left(\left[\frac{\partial \ell(\mathrm{y}_i, \hat{y})}{\partial \hat{y}} \right]_{\hat{y} = f_{b-1}(\mathrm{x}_i)} - \phi(\mathrm{x}_i; R, c) \right)^2$$
4: Compute values c_b by minimizing ℓ: ▷ more powerful than step size search
$$c_b = \arg\min_c \sum_{i=1}^n \ell(y_i, f_{b-1}(x_i) + \phi(x_i; R_b, c))$$
5: $f_b : \mathrm{x} \to f_{b-1}(\mathrm{x}) + \phi(\mathrm{x}; R_b, c_b)$
6: **end for**

Shrinkage Coefficient (Learning Rate) ν

$f_b : \mathrm{x} \to f_{b-1}(\mathrm{x}) + \nu\phi(\mathrm{x}; R_b, c_b)$, $(0 < \nu \leqslant 1)$

$\to$ Smaller ν favor better test error but requires larger B

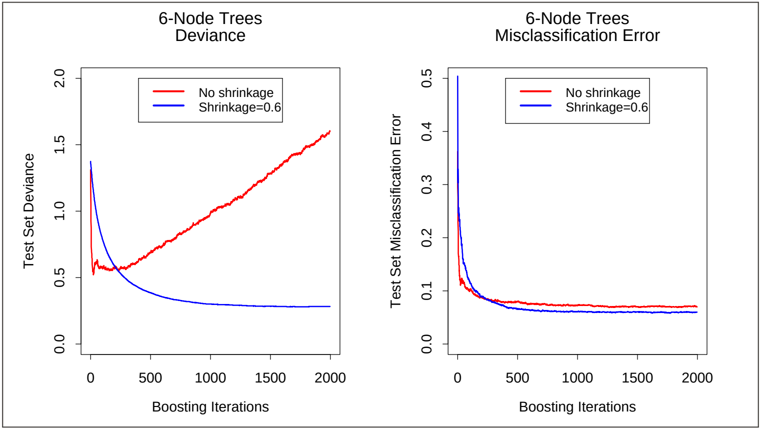


Figure from [Friedman et al., 2001] plotting the deviance loss $\log(1 + \exp(-2yf(x)))$ used to train the model and the misclassification error

Subsampling fraction η: Stochastic Gradient Descent

At each boosting step, only a fraction η of the data is used (randomly selected at each iteration without replacement) to obtain the tree
$\rightarrow$ Faster and ultimately more accurate

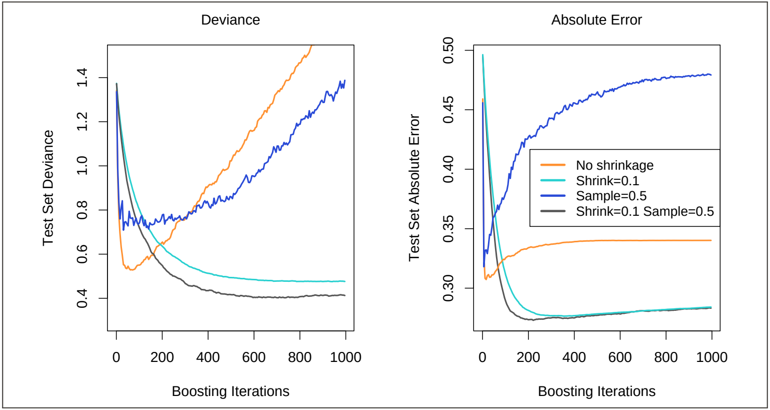


Figure from [Friedman et al., 2001] plotting the deviance loss $\log(1 + \exp(-2yf(x)))$ and the absolute error for two different problems

Hyper-parameters Tuning and Performance

Most Important Hyper-parameters

- B: number of trees (thousands, fixed value)
- J: number of nodes ($4 \leqslant J \leqslant 8$)
- ν: shrinkage factor or learning rate (between 10^{-3} and 10^{-2})
- η: subsampling fraction (0.5, lower if lots of data, higher if few data)

Advantage/Disadvantage

No preprocessing required	Lots of hyper-parameters
Can optimize robust loss	Requires extensive tuning
Very flexible	Computationally intensive

Performance

Efficient libraries: **LightGBM**, XGBoost and CatBoost

One of the most powerful method on "tabular data"/"unstructured data"

My two cents' worth ranking:

gradient boosted trees $>$ random forest $>$ bagging of trees $>$ single tree

[Badirli et al., 2020] Badirli, S., Liu, X., Xing, Z., Bhowmik, A., and Keerthi, S. S. (2020).
Gradient boosting neural networks: Grownet.
arXiv preprint arXiv:2002.07971.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997).
A decision-theoretic generalization of on-line learning and an application to boosting.
Journal of computer and system sciences, 55(1):119–139.

[Friedman et al., 2001] Friedman, J., Hastie, T., and Tibshirani, R. (2001).
The elements of statistical learning, volume 1.
Springer series in statistics New York.

[Friedman, 2001] Friedman, J. H. (2001).
Greedy function approximation: a gradient boosting machine.
Annals of statistics, pages 1189–1232.

[Schapire and Freund, 2012] Schapire, R. E. and Freund, Y. (2012).
Boosting: Foundations and algorithms.