# Gradient Boosting の基礎理論

Jake Underland

October 14, 2021

# Setup

シンプルなセットアップから出発する.

$\{x_i, y_i\}$ はデータセットを表す.

$N$ はトレーニングサンプルの数を指し, $M$ は特徴量の数を指す.

# Setup

回帰という文脈においては，

$$y_i \approx f(x_i) \text{ for all } i$$

つまり

$$\arg\min_{f} \underbrace{\frac{1}{N}\sum_{i}^{N}(y_i - f(x))^2}_{\textit{Mean Squared Error (MSE)}}$$

となる $f$ を探すのが目的である．

# Setup

最終的に，$f$ は弱い学習器の和として表現される

$$\underbrace{f(x)}_{\textit{final model}} = \underbrace{f_0(x) + f_1(x) + f_2(x) + \cdots + f_{\max}(x)}_{\textit{weak learners}}$$

モデルを作る段階は以下のように表せる：

$$\begin{aligned} S_0(x) &= f_0(x) \\ S_1(x) &= f_0(x) + f_1(x) \\ S_2(x) &= f_0(x) + f_1(x) + f_2(x) \\ &\vdots \\ S_{k+1}(x) &= S_k(x) + f_{k+1}(x) \end{aligned}$$

$S$ は各段階でのモデルを表しており，最終段階では $S_{max}$ でモデルが完成する．

# Optimizing

全ての段階で目指すのは $S$ による $y$（観察されたトレーニングデータ）の近似である．

$$S_{k+1}(x) = S_k(x) + f_{k+1}(x) \approx y$$

この時の弱学習器 $f_{k+1}$ に着目して並べ替えると，

$$f_{k+1}(x) \approx \underbrace{y - S_k(x)}_{Residual}$$

つまり各段階で前段階のモデルの残差項に $x$ を回帰していき，最適な $f$ を得る．

Gradient Boosting では，段階を追って調整しながらモデルの $f$ を決める．

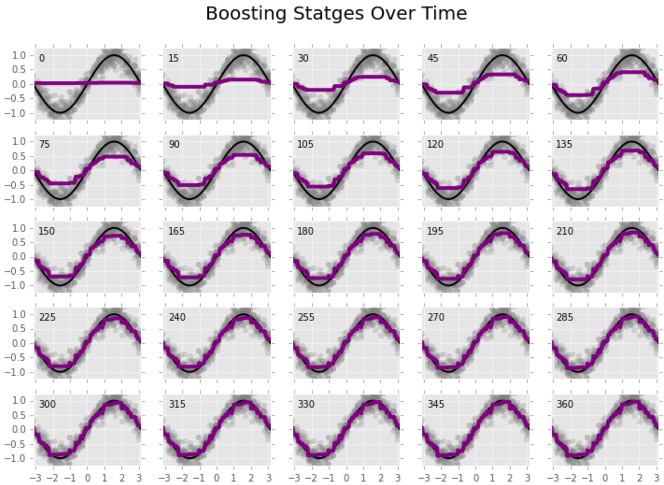
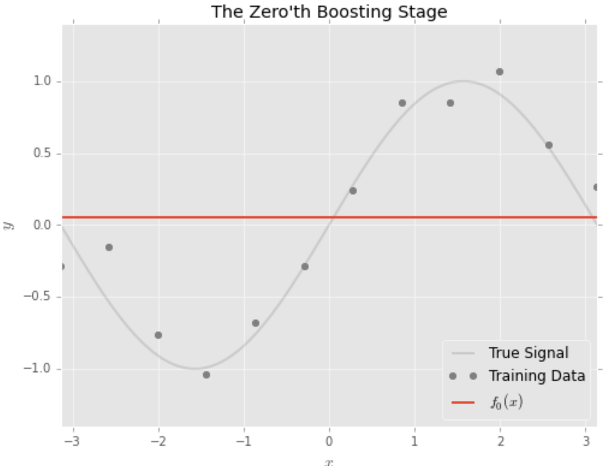



**Figure:** Boosting による段階的近似, borrowed from Matthew Drury

# 図で段階を追う

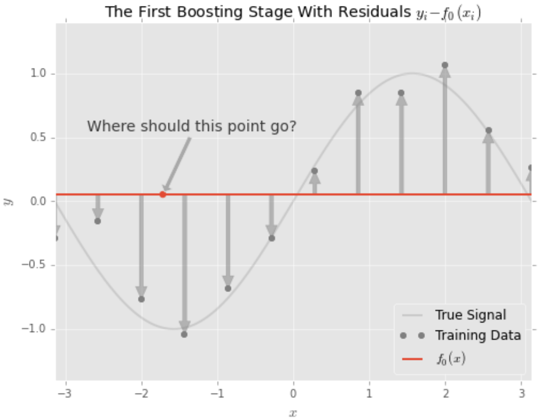
まず $S_0(x) = f_0(x)$ だが，この時 MSE を最小化する値は言わずとも

$$f_0(x) = \frac{1}{N}\sum_i y_i$$



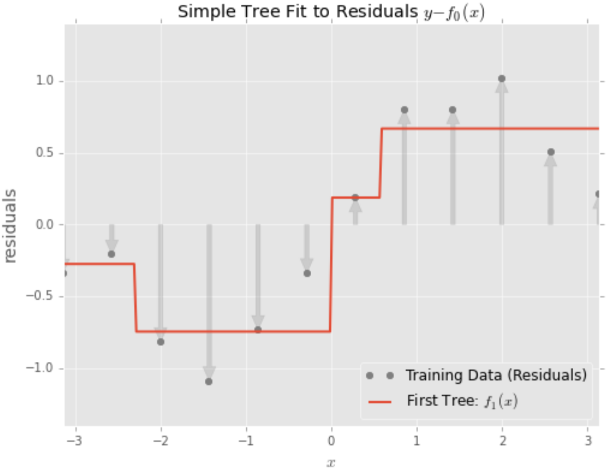
残差である $y_i - f_0(x_i)$ の方向へモデルを調整していく！
この際，トレーニングデータが存在する点でしか残差は定義されない

$\longrightarrow$ 回帰を通じて残差にモデルをフィットする！



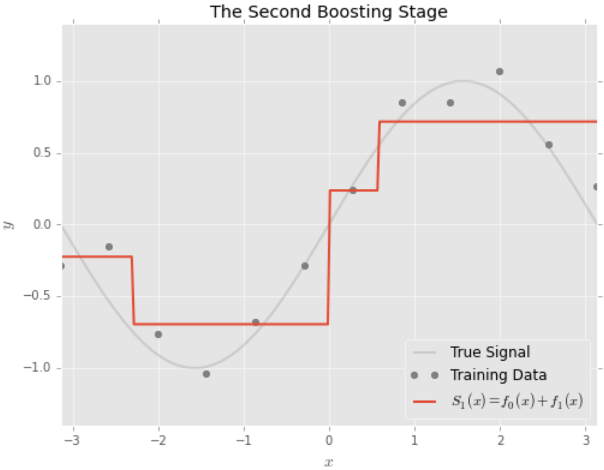
# $f_1(x)$ を計算する

残差を新たなデータセットとして，決定木回帰により最初の Tree$f_1(x)$ を推定する．



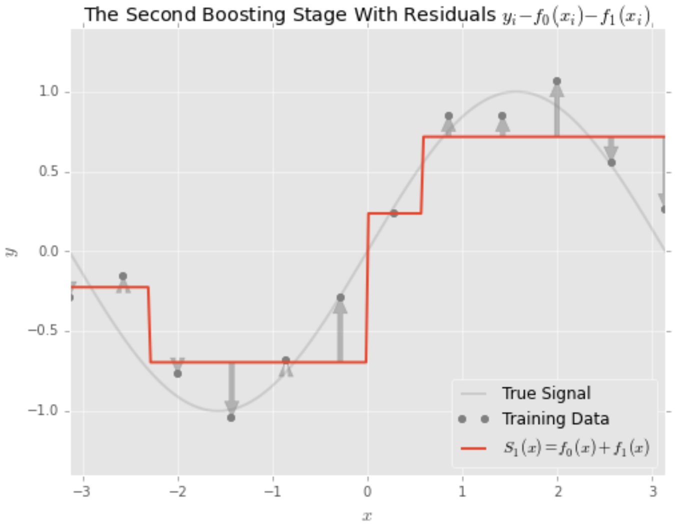
# モデルの更新

$$S_1(x) = f_0(x) + f_1(x) \leftarrow \text{Model fit to residuals!}$$



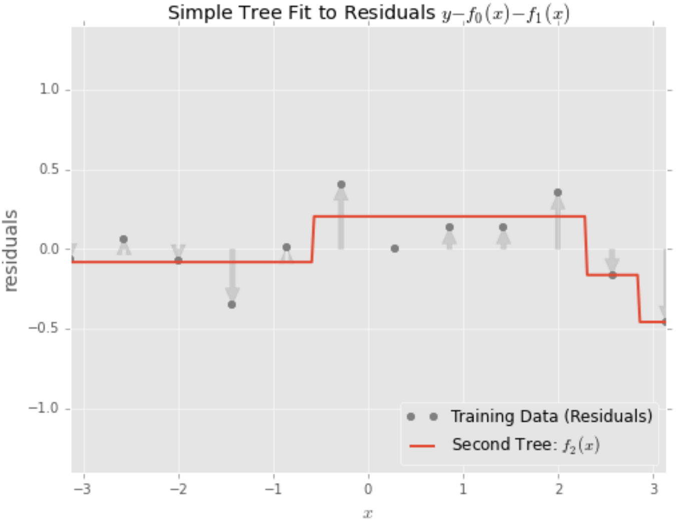
# Another Step

現時点のモデルの残差を計算...

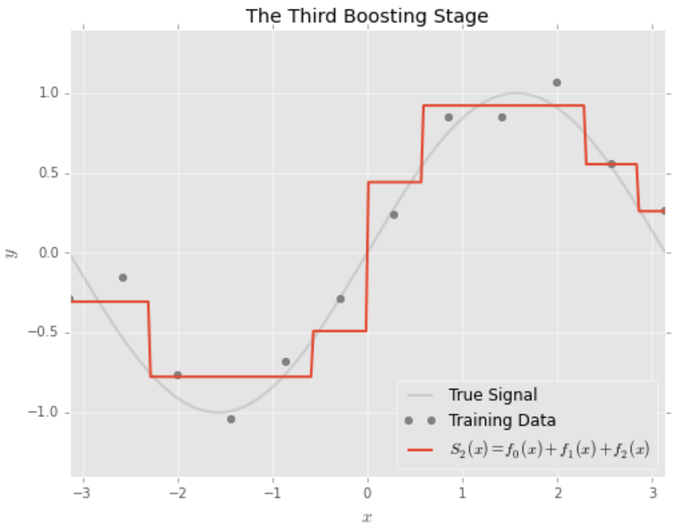


残差が被説明変数となるようなトレーニングデータセットを作る...

# モデルに追加する弱学習器を推定する

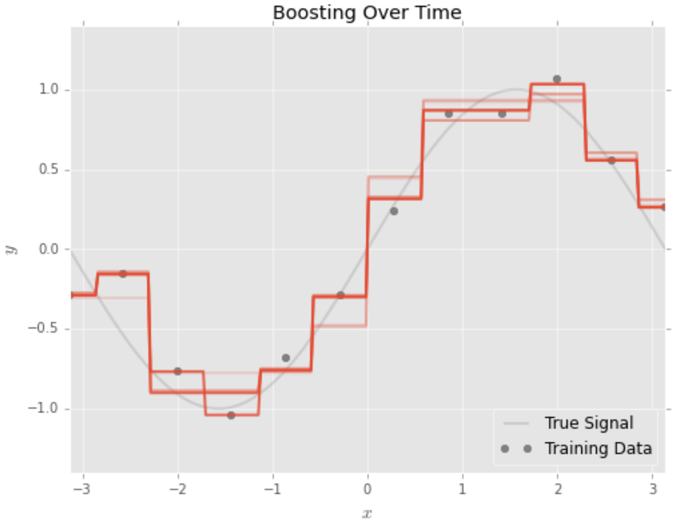


# モデルの更新！



The Third Boosting Stage
1.0
0.5
0.0
−0.5
−1.0
y
−3
−2
−1
0
1
2
3
x
True Signal
Training Data
$S_2(x) = f_0(x) + f_1(x) + f_2(x)$

# 漸近的に近づける！


Boosting Over Time
1.0
0.5
0.0
−0.5
−1.0
y
−3
−2
−1
0
1
2
3
x
True Signal
Training Data

# 損失関数

先ほどまでの例では MSE を損失関数として使用

それに限らず，損失関数 $L\left(y, S(x)\right)$ を定義できる $\downarrow$

$$S_{k+1}(x) = S_k(x) + \arg\min_f \sum_{i=1}^{N} L(y_i, S_k(x_i) + f_{k+1}(x_i))$$

しかし，毎回グローバルなミニマムを探すのは難しい...
$\rightarrow$ 勾配法を導入, ローカルな極小を発見

$$S_{k+1}(x) = S_k(x) - \left[\sum_{i=1}^{N} \nabla_{S_k} L(y_i, S_k(x_i))\right]$$

**Gradient Boosting!**

# 勾配降下法

最急勾配法は任意の微分可能な関数 $L(x)$ の最適化手法

**Inputs:** 関数 $L$.

**Outputs:** $L$ を最小化する点 $x^*$.

**Algorithm:** $x$ が $x^*$ へ収束するまで $x_{i+1} = x_i - \lambda \nabla L(x_i)$ を繰り返す.

# Gradient Boosting

$$S_{k+1}(x) = S_k(x) - \left[\sum_{i=1}^{N} \nabla_{S_k} L(y_i, S_k(x_i))\right]$$

**Gradient Boosting!**

# 弱学習器のウェイト

モデル更新の際，全ての弱学習器をただモデルに足していくのではなく，ウェイトを付加することで予測性能をさらに上げる．

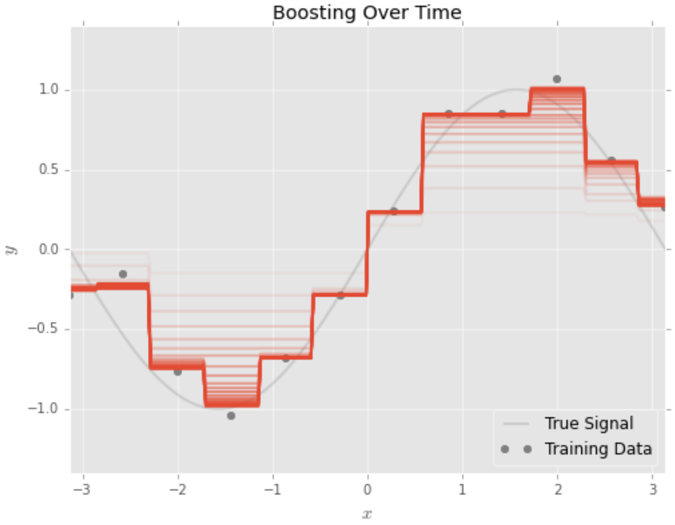
$$S_{k+1}(x) = S_k(x) + \lambda_{k+1} f_{k+1}(x)$$

$$\left( = S_k(x) - \lambda_{k+1} \sum_{i=1}^{N} \nabla_{S_k} L(y_i, S_k(x_i)) \right)$$

この時，

$$\begin{aligned}
\lambda_{k+1} &= \arg\min_{\lambda} \sum_{i=1}^{N} L\left(y_i, S_{k+1}(x_i)\right) \\
&= \arg\min_{\lambda} \sum_{i=1}^{N} L\left(y_i, S_k(x_i) + \lambda_{k+1} f_{k+1}(x_i)\right)
\end{aligned}$$

# ウェイトによる滑らかな近似



# Gradient Tree Boosting

GTB は使われている弱学習器が決定木の場合をいう.

- 決定木のリーフの数を $J_k$ とする
- 木はフィーチャースペースを $R_{1,k}, \ldots, R_{J_k,k}$ に分画する.

$$f_k(x) = \sum_{j=1}^{J_m} \underbrace{b_j m}_{R_{jm}\text{の予測値}} \mathbb{1}_{R_{jm}}(x)$$

# ちなみに...

最小二乗法に勾配降下法を当てはめると，

$$\begin{aligned} x_{k+1} &= x_k - \lambda \nabla_x L(x_k, y) \\ &= x_k - \lambda \nabla_x \frac{\partial}{\partial x}\left(\frac{1}{2}(y - x_k)^2\right) \\ &= x_k + \lambda(y - x_k) \end{aligned}$$

つまり，最小二乗法のもとでは段階的に残差の方向に向かって動いていく．

# 復習

**Inputs:** A training data set $\{x_i, y_i\}$, and, optionally, a learning rate $\lambda$ to replace weights.
**Returns:** A function $f$ such that $f(x_i) \approx y_i$.

- Initialize $S_0(x) = f_0(x) = \frac{1}{N}\sum_i y_i$.
- Iterate (parameter $k$) until satisfied:
  - Create the working data set $W_k = \{x_i, y_i - S_k(x_i)\}$.
  - Fit a regression tree to $W_k$, minimizing least squares. Call this tree $f_k$.
  - Set $S_{k+1}(x) = S_k(x) + \lambda f_k(x)$.
- Return $f_{\max}(x) = f_0(x) + f_1(x) + f_2(X) + \cdots + f_{\max}(x)$.