

Extreme gradient boosting

Journal club May 18th, 2022

Gradient boosting

Input: training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x))$, number of iterations M .

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma).$$

2. For $m = 1$ to M :

1. Compute so-called *pseudo-residuals*:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \dots, n.$$

2. Fit a base learner (or weak learner, e.g. tree) closed under scaling $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^n$.

3. Compute multiplier γ_m by solving the following [one-dimensional optimization](#) problem:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)).$$

4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

Gradient boosting : Input

Input: Data $\{(x_i, y_i)\}_{i=1}^n$, and a differentiable **Loss Function** $L(y, F(x))$

...and **$F(x)$** is a function that gives us the **Predicted** values.

$$\frac{1}{2} (\text{Observed} - \text{Predicted})^2$$

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

Gradient boosting : Step 1 Constant

Step 1: Initialize model with a constant value: $F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, \gamma)$

$$\frac{1}{2} (88 - \text{Predicted})^2 + \\ \frac{1}{2} (76 - \text{Predicted})^2 + \\ \frac{1}{2} (56 - \text{Predicted})^2$$

...and the “**argmin over gamma**” means we need to find a **Predicted** value that minimizes this sum.

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

Gradient boosting : Step 1 Constant

Step 1: Initialize model with a constant value: $F_0(x) = \operatorname{argmin}_{\gamma} \sum_{i=1}^n L(y_i, \gamma)$

$$\frac{1}{2} (88 - \text{Predicted})^2 + \frac{1}{2} (76 - \text{Predicted})^2 + \frac{1}{2} (56 - \text{Predicted})^2 \longrightarrow -(88 - \text{Predicted}) + -(76 - \text{Predicted}) + -(56 - \text{Predicted}) = 0$$

$\frac{d}{d \text{ Predicted}}$

Then we set sum of the derivatives equal to zero...

Gradient boosting : Step 1 Constant

Step 1: Initialize model with a constant value: $F_0(x) = \operatorname{argmin}_{\gamma} \sum_{i=1}^n L(y_i, \gamma)$

That means that the
initial predicted value,
 $F_0(x)$, is just a leaf.

$$F_0(x) = \frac{88 + 76 + 56}{3} \\ = 73.3$$



73.3

Gradient boosting : Step 2

Step 2 is huge, but we'll take it one step at a time. :)

Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm} , for $j = 1 \dots J_m$

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

Gradient boosting : Step 2(A) calculate residuals

Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

$$\frac{d}{d \text{ Predicted}} \frac{1}{2} (\text{Observed} - \text{Predicted})^2$$

...and we've already
calculated this.

$$= -(\text{Observed} - \text{Predicted})$$

Gradient boosting : Step 2(A) calculate residuals

Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	-17.3

Hooray! We've finished **Part A** of **Step 2** by calculating a **Residual** for each sample.

Gradient boosting : Step 2(A) calculate residuals

Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	-17.3

Hooray! We've finished **Part A** of **Step 2** by calculating a **Residual** for each sample.

Gradient boosting : Step 2(B) Regression tree to residuals

Gradient Boost Part 2 (of 4): Regression Details



(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm} , for $j = 1 \dots J_m$

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	50	-17.3

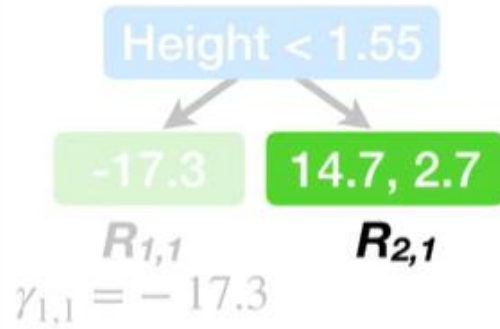
Gradient boosting : Step 2(C) Optimize leaf output values



Specifically, since two residuals ended up in this leaf, it's unclear what its **Output Value** should be.

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \operatorname{argmin}_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

Gradient boosting : Step 2(C) Optimize leaf output values

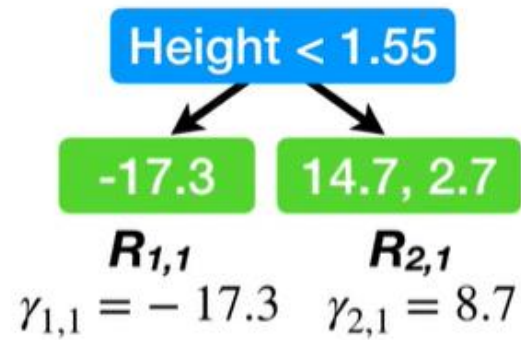


...and plug in **73.3**
for $F_{m-1}(x_1)$ and
 $F_{m-1}(x_2)$.

$F_0(x)$
73.3

$$\gamma_{2,1} = \operatorname{argmin}_{\gamma} \left[\frac{1}{2} (88 - (F_{m-1}(x_1) + \gamma))^2 + \frac{1}{2} (76 - (F_{m-1}(x_2) + \gamma))^2 \right]$$

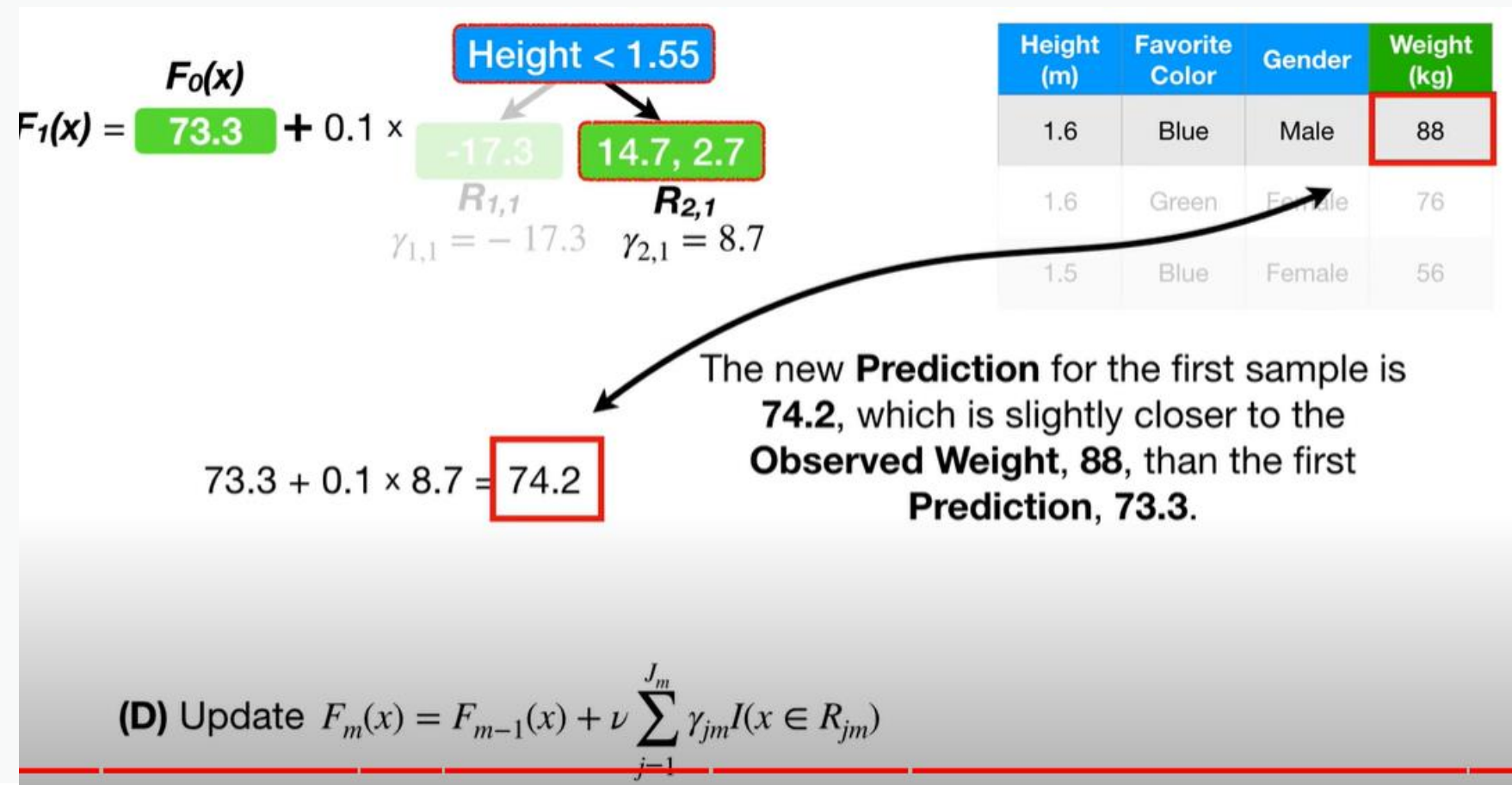
Gradient boosting : Step 2(C) Optimize leaf output values



Hooray!!!! We finished **Part C** of **Step 2** by computing **gamma** values, or **Output Values**, for each leaf.

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \operatorname{argmin}_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

Gradient boosting : Step 2(D) Update prediction with new values



Gradient boosting : Step 2(D) Update prediction with new values

$$F_1(x) = F_0(x) + 0.1 \times \begin{cases} -17.3 & \text{Height} < 1.55 \\ 14.7, 2.7 & \text{Height} \geq 1.55 \end{cases}$$

$R_{1,1}$ $R_{2,1}$
 $\gamma_{1,1} = -17.3$ $\gamma_{2,1} = 8.7$

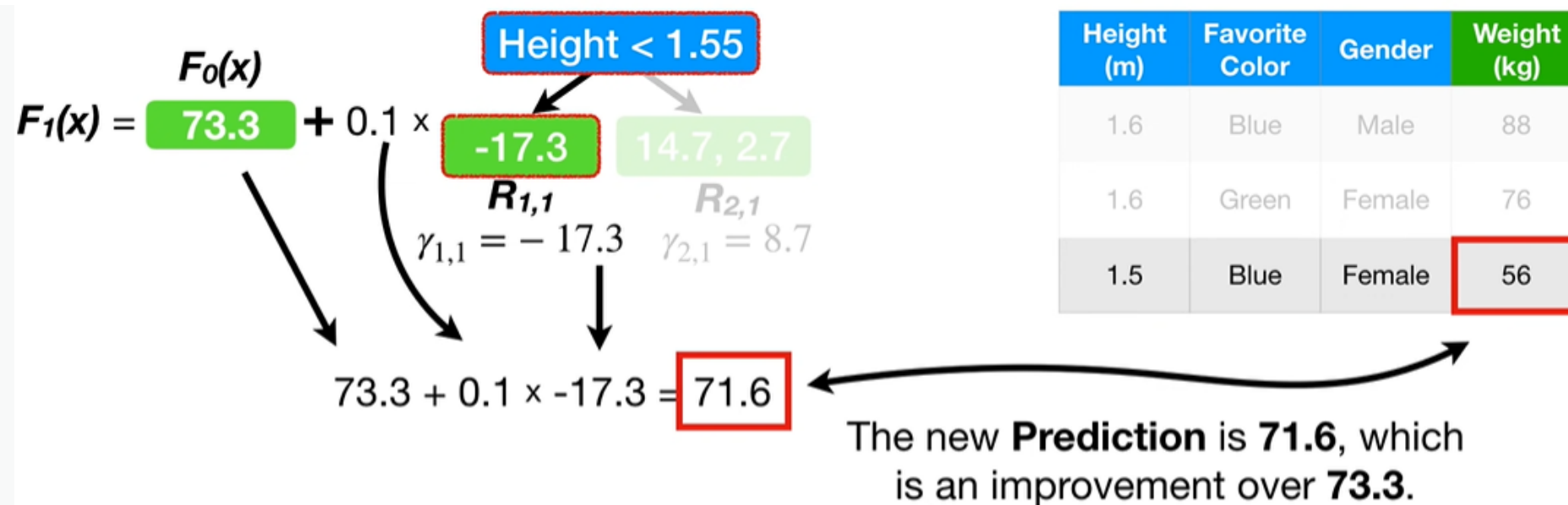
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

$$73.3 + 0.1 \times 8.7 = 74.2$$

The new **Prediction** for x_2 is also **74.2**, which is an improvement over the first **Prediction, 73.3**.

(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

Gradient boosting : Step 2(D) Update prediction with new values



(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

Hooray!!!!

We made it through one iteration of of **Step 2!!!**

Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm} , for $j = 1 \dots J_m$

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

Summary of step 2 >

Gradient boosting : Recap

Then we solved for the negative **Gradient**...

...and the latest **Predictions**...

$F_0(x)$
73.3

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

...and that gave us **Residuals**.

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

...plugged in the **Observed** values...

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	-17.3

Gradient boosting : Recap

Then we fit a **Regression Tree**
to the **Residuals**...

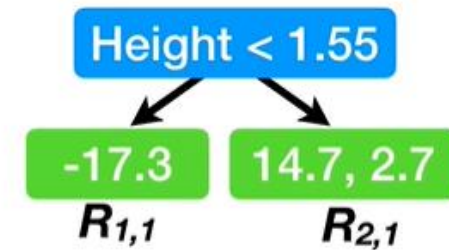
Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm} , for $j = 1 \dots J_m$

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \operatorname{argmin}_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$

(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$



Gradient boosting : Recap

...and computed the **Output Values**, $\gamma_{j,m}$, for each leaf.

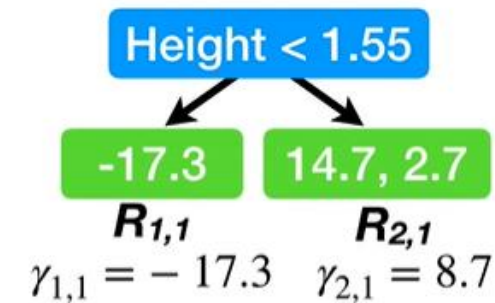
Step 2: for $m = 1$ to M :

(A) Compute $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \dots, n$

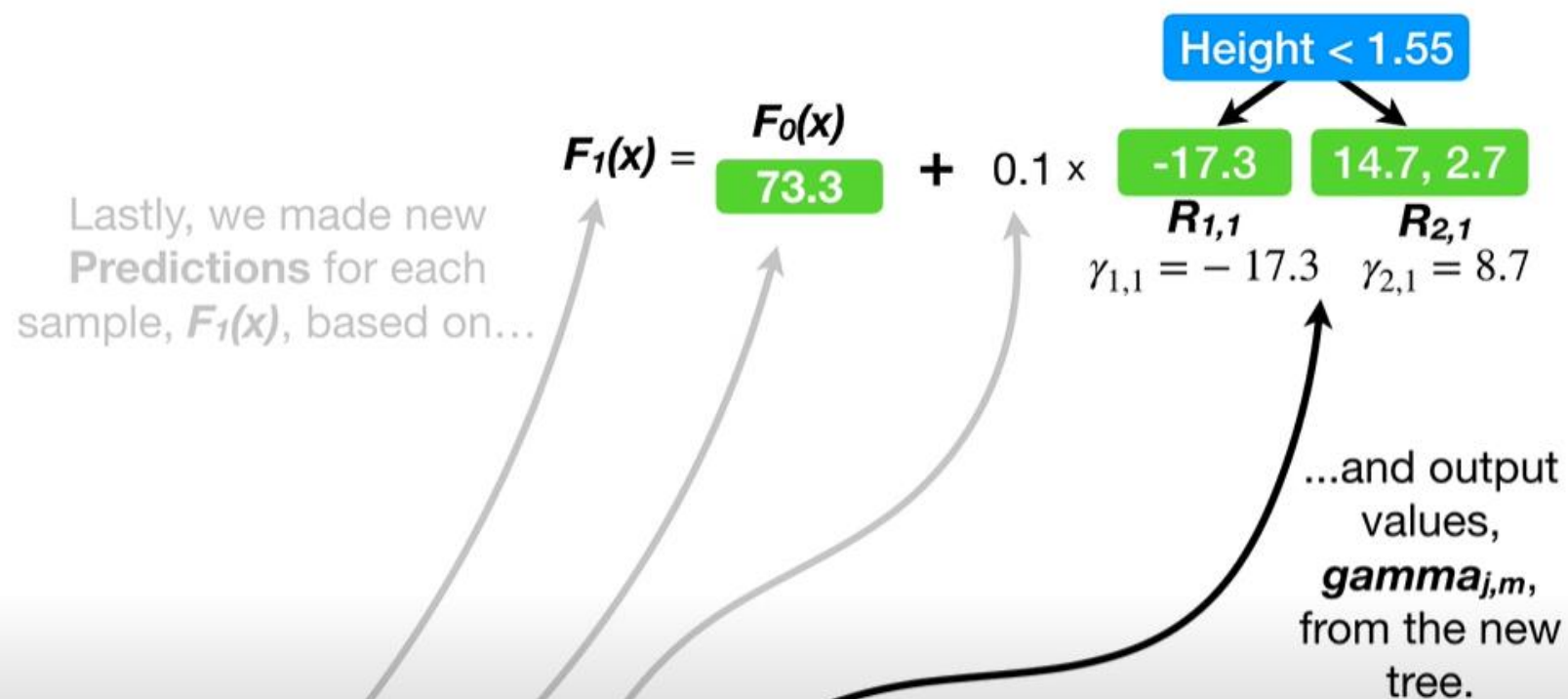
(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm} , for $j = 1 \dots J_m$

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \operatorname{argmin}_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$



Gradient boosting : Recap



$$\text{(D) Update } F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

Gradient boosting : Recap

If $M = 2$, then $F_2(x)$ is the output from the **Gradient Boost** algorithm.

$$F_2(x) = F_0(x) + 0.1 \times \begin{array}{c} \text{Height} < 1.55 \\ \swarrow \quad \searrow \\ \begin{array}{cc} -17.3 & 14.7, 2.7 \\ R_{1,1} & R_{2,1} \\ \gamma_{1,1} = -17.3 & \gamma_{2,1} = 8.7 \end{array} \end{array} + 0.1 \times \begin{array}{c} \text{Height} < 1.55 \\ \swarrow \quad \searrow \\ \begin{array}{cc} -15.6 & 13.8, 1.8 \\ R_{1,2} & R_{2,2} \\ \gamma_{1,2} = -15.6 & \gamma_{2,2} = 7.8 \end{array} \end{array}$$

Input: training set $\{(x_i, y_i)\}_{i=1}^N$, a differentiable loss function $L(y, F(x))$, a number of weak learners M and a learning rate α .

Algorithm:

1. Initialize model with a constant value:

$$\hat{f}_{(0)}(x) = \arg \min_{\theta} \sum_{i=1}^N L(y_i, \theta).$$

2. For $m = 1$ to M :

1. Compute the 'gradients' and 'hessians':

$$\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=\hat{f}_{(m-1)}(x)}.$$

$$\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2} \right]_{f(x)=\hat{f}_{(m-1)}(x)}.$$

2. Fit a base learner (or weak learner, e.g. tree) using the training set $\left\{ x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} \right\}_{i=1}^N$ by solving the optimization problem below:

$$\hat{\phi}_m = \arg \min_{\phi \in \Phi} \sum_{i=1}^N \frac{1}{2} \hat{h}_m(x_i) \left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \phi(x_i) \right]^2.$$

$$\hat{f}_m(x) = \alpha \hat{\phi}_m(x).$$

3. Update the model:

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_m(x).$$

3. Output $\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^M \hat{f}_m(x).$

Extreme !!! Gradient boosting

Gradient Boost-(ish)

Regularization

A Unique Regression Tree

Approximate Greedy Algorithm

Weighted Quantile Sketch

Sparsity-Aware Split Finding

Parallel Learning

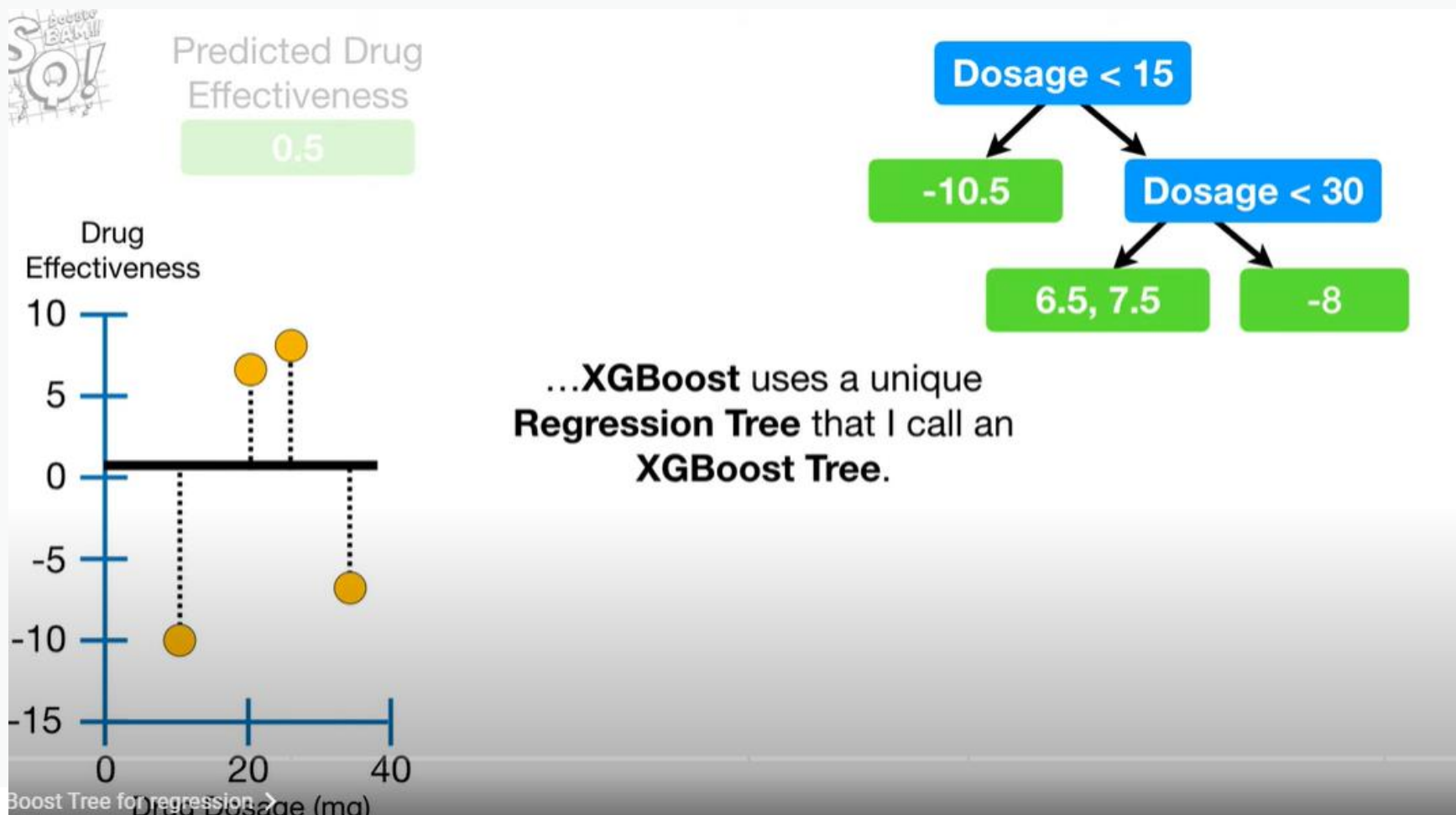
Cache-Aware Access

Blocks for Out-of-Core Computation

...so we'll start by learning about
XGBoost's unique **Regression Trees**.



Extreme Gradient boosting : unique regression tree



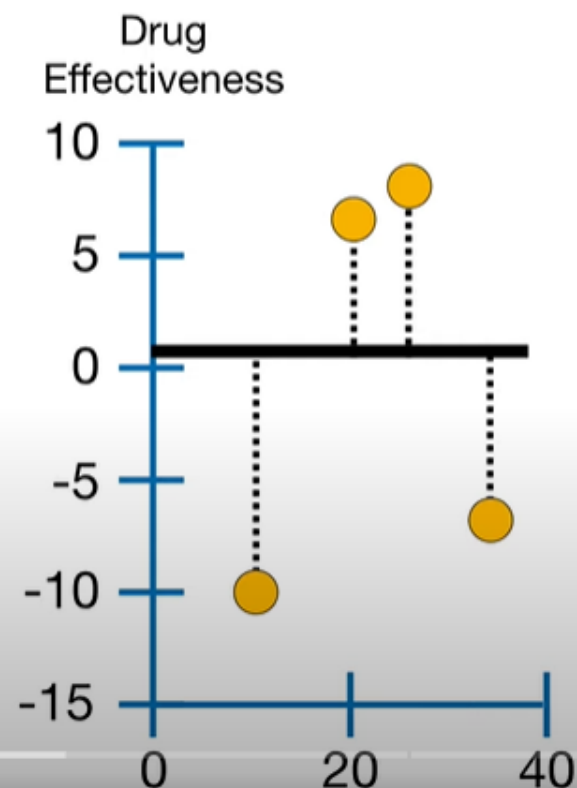
Extreme Gradient Boosting: unique regression tree



Predicted Drug Effectiveness

0.5

-10.5, 6.5, 7.5, -7.5



$$\text{Similarity Score} = \frac{(-10.5 + 6.5 + 7.5 + -7.5)}{4 + 0}$$

NOTE: Because we do not square the **Residuals** before we add them together in the numerator, **7.5** and **-7.5** cancel each other out.

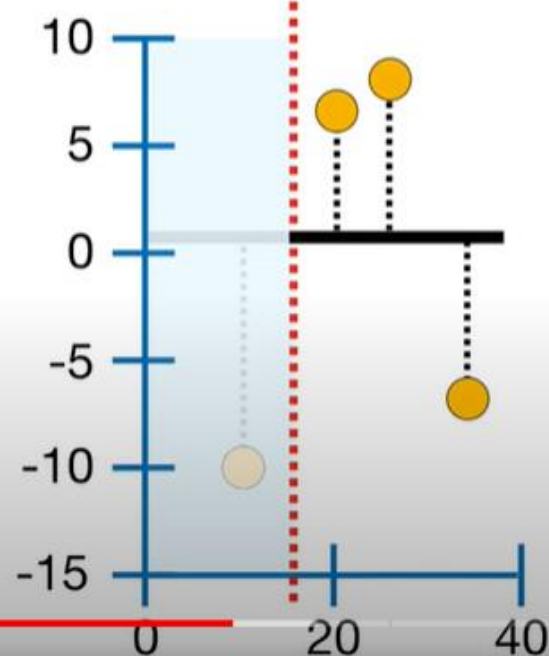
Extreme Gradient boosting : unique regression tree



Predicted Drug Effectiveness

0.5

Drug Effectiveness



Dosage < 15 Similarity = 4

-10.5

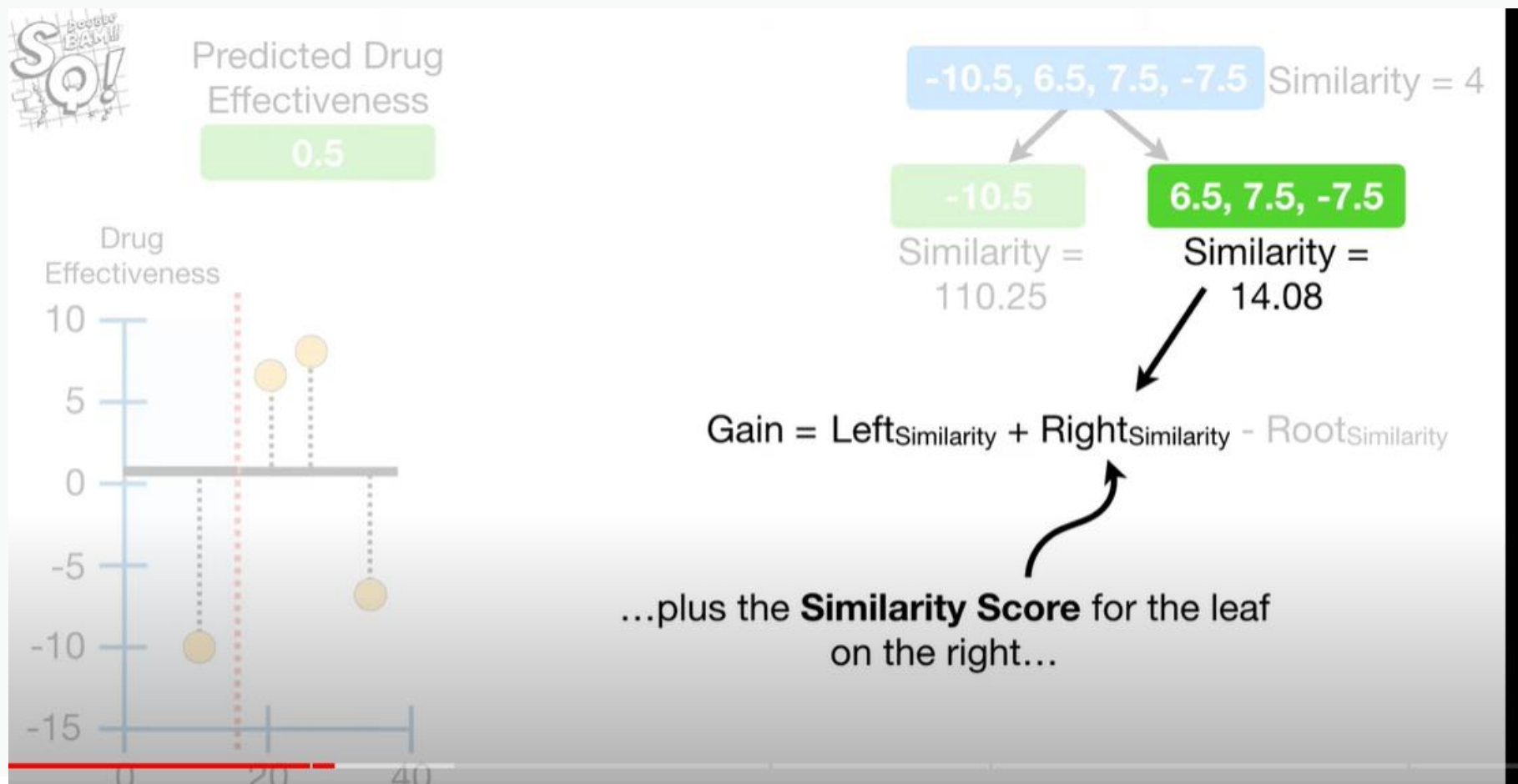
Similarity =
110.25

6.5, 7.5, -7.5

$$\text{Similarity Score} = \frac{(6.5 + 7.5 + -7.5)^2}{3 + \mathbf{0}}$$

...and just like before,
let's let $\lambda = \mathbf{0}$.

Extreme Gradient boosting : unique regression tree

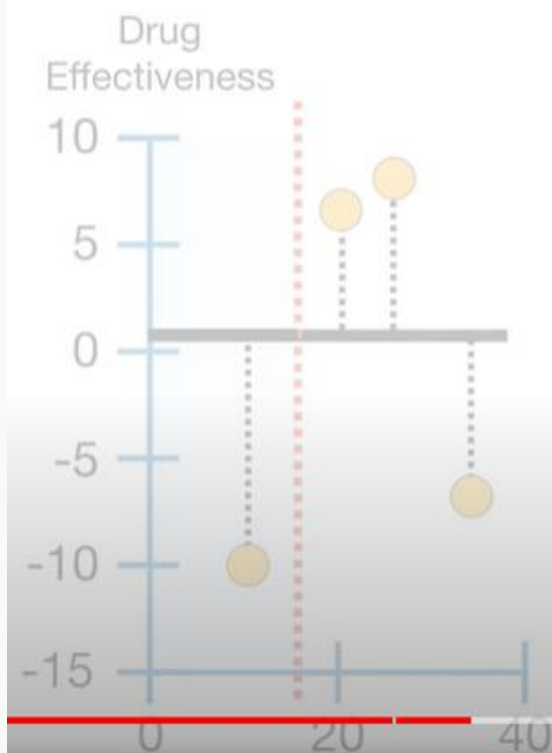


Extreme Gradient boosting : unique regression tree



Predicted Drug Effectiveness

0.5



gain to evaluate different thresholds ->

Dosage < 15

Similarity = 4

-10.5

Similarity =
110.25

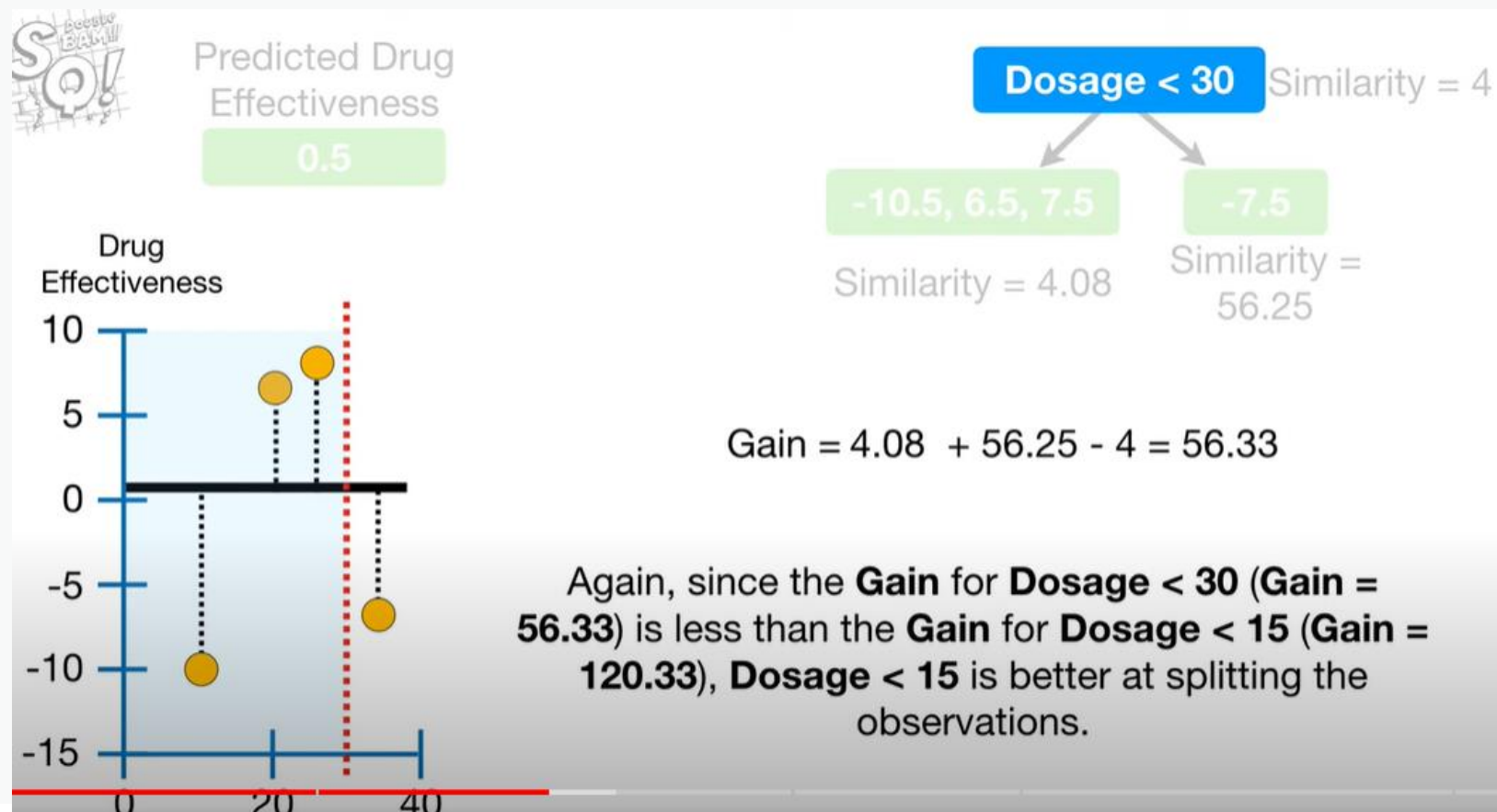
6.5, 7.5, -7.5

Similarity =
14.08

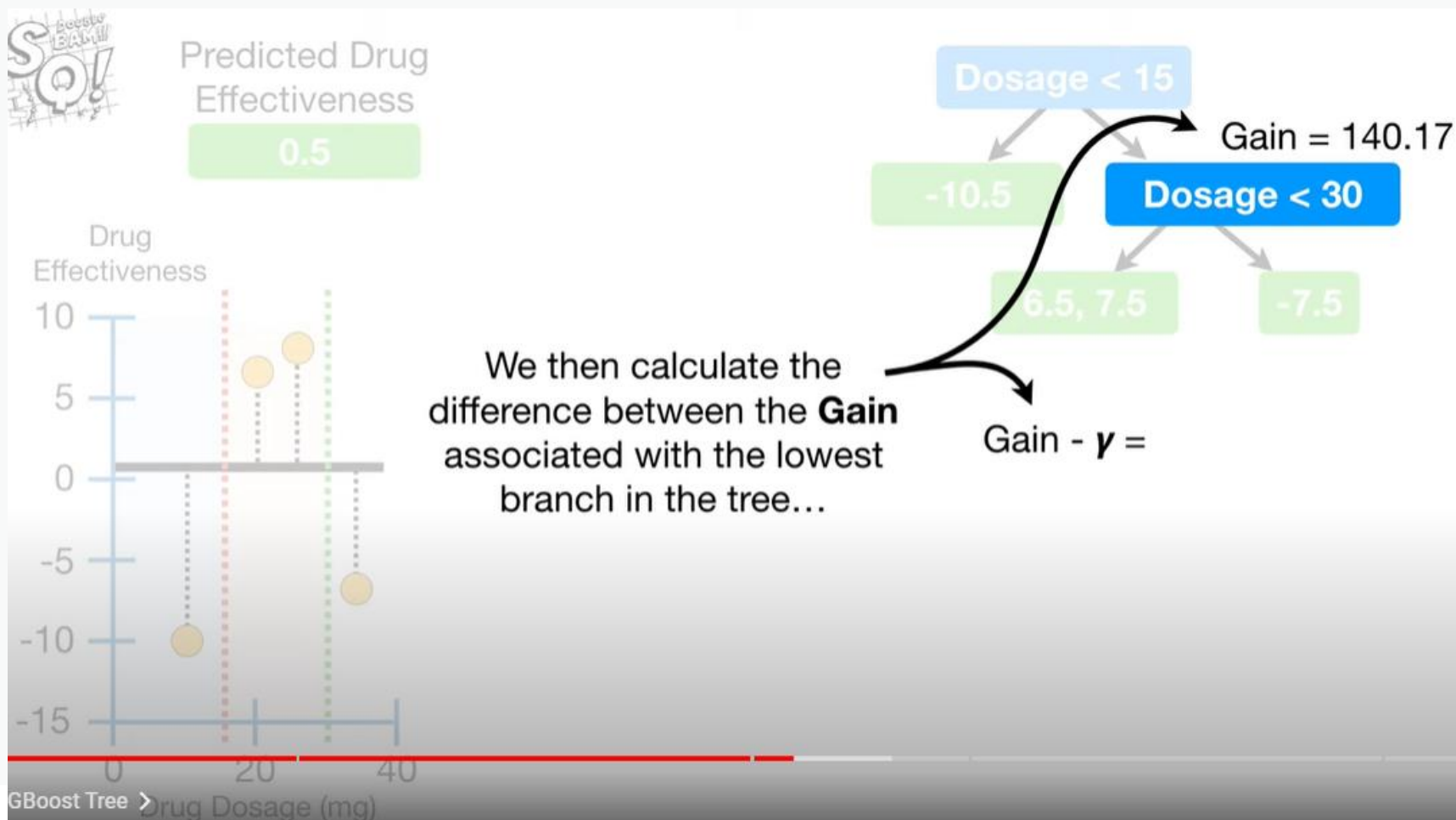
$$\text{Gain} = 110.25 + 14.08 - 4 = 120.33$$

Now that we have calculated the **Gain** for the threshold **Dosage < 15**, we can compare it to the **Gain** calculated for other thresholds.

Extreme Gradient boosting : unique regression tree



Extreme Gradient boosting : Pruning



Extreme Gradient boosting : Pruning

