Extreme gradient boosting

Journal club May 18th , 2022

patientforward

confidential

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:

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

2. For m = 1 to M:

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

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

    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:

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

    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** $L(y_i, \boxed{F(x)})$

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

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

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: $F_0(x) = \boxed{\underset{\gamma}{\operatorname{argmin}}} \sum_{i=1}^{n} L(y_i, \gamma)$

$$\boxed{\begin{aligned}&\frac{1}{2}(\mathbf{88} - \textbf{Predicted})^2 + \\ &\qquad \frac{1}{2}(\mathbf{76} - \textbf{Predicted})^2 + \\ &\qquad\qquad \frac{1}{2}(\mathbf{56} - \textbf{Predicted})^2\end{aligned}}$$

…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: $F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$

$$\frac{d}{d\,Predicted}$$

$$\frac{1}{2}(88 - \text{Predicted})^2 + \longrightarrow -(88 - \textbf{Predicted}) +$$

$$\frac{1}{2}(76 - \text{Predicted})^2 + \longrightarrow -(76 - \textbf{Predicted}) +$$

$$\frac{1}{2}(56 - \text{Predicted})^2 \longrightarrow -(56 - \textbf{Predicted})$$

$$\boxed{= 0}$$

Then we set sum of the derivatives equal to zero…

Gradient boosting : Step 1 Constant

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

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

↓

73.3

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

$$= 73.3$$

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 $i = 1,\ldots,n$

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

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

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

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 $i = 1,...,n$

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

…and we've already calculated this.

$$\boxed{= -(\textbf{Observed - Predicted})}$$

Gradient boosting : Step 2(A) calculate residuals

Step 2: for m = 1 to M:

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

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

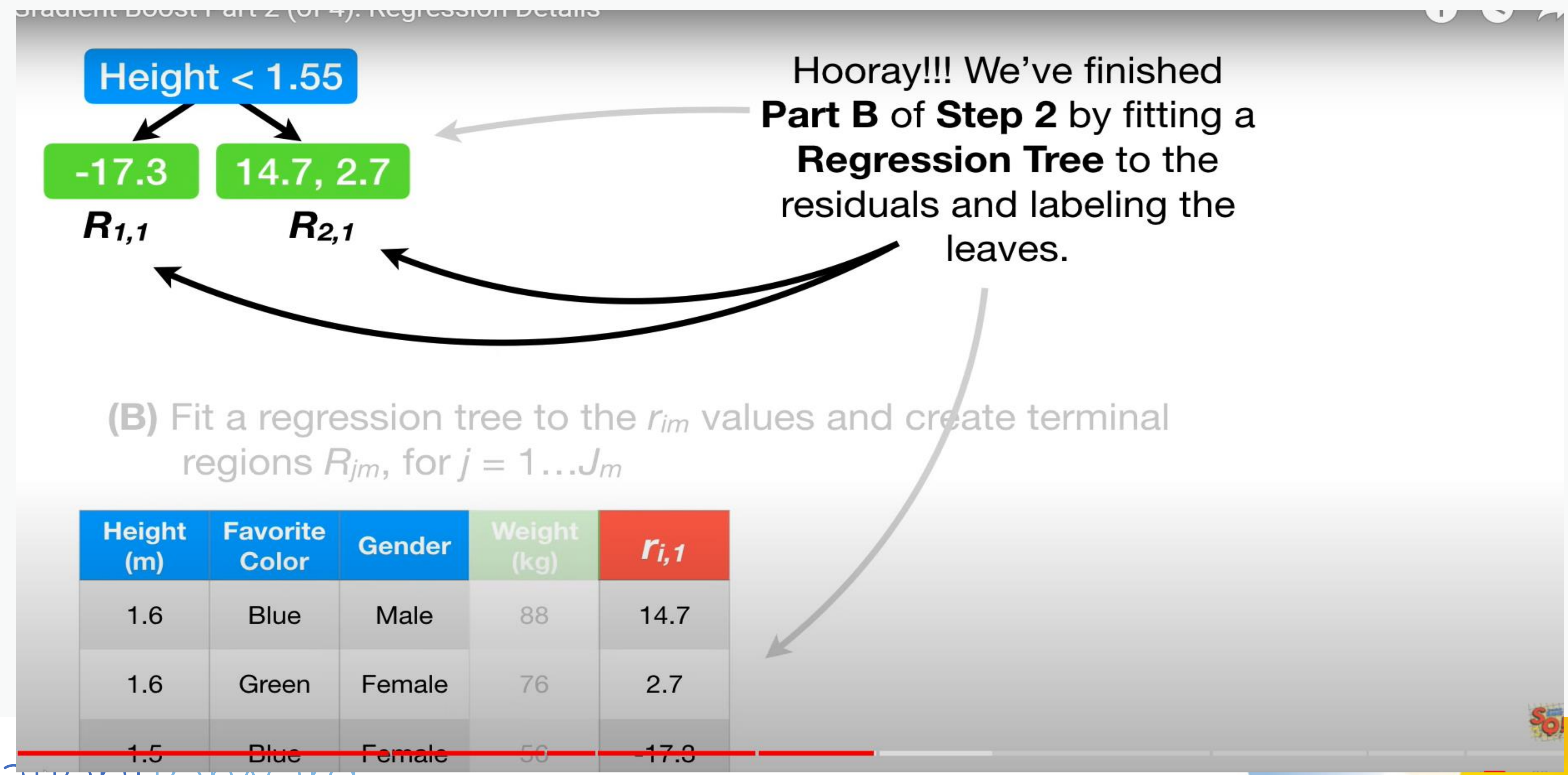
Step 2: for m = 1 to M:

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

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

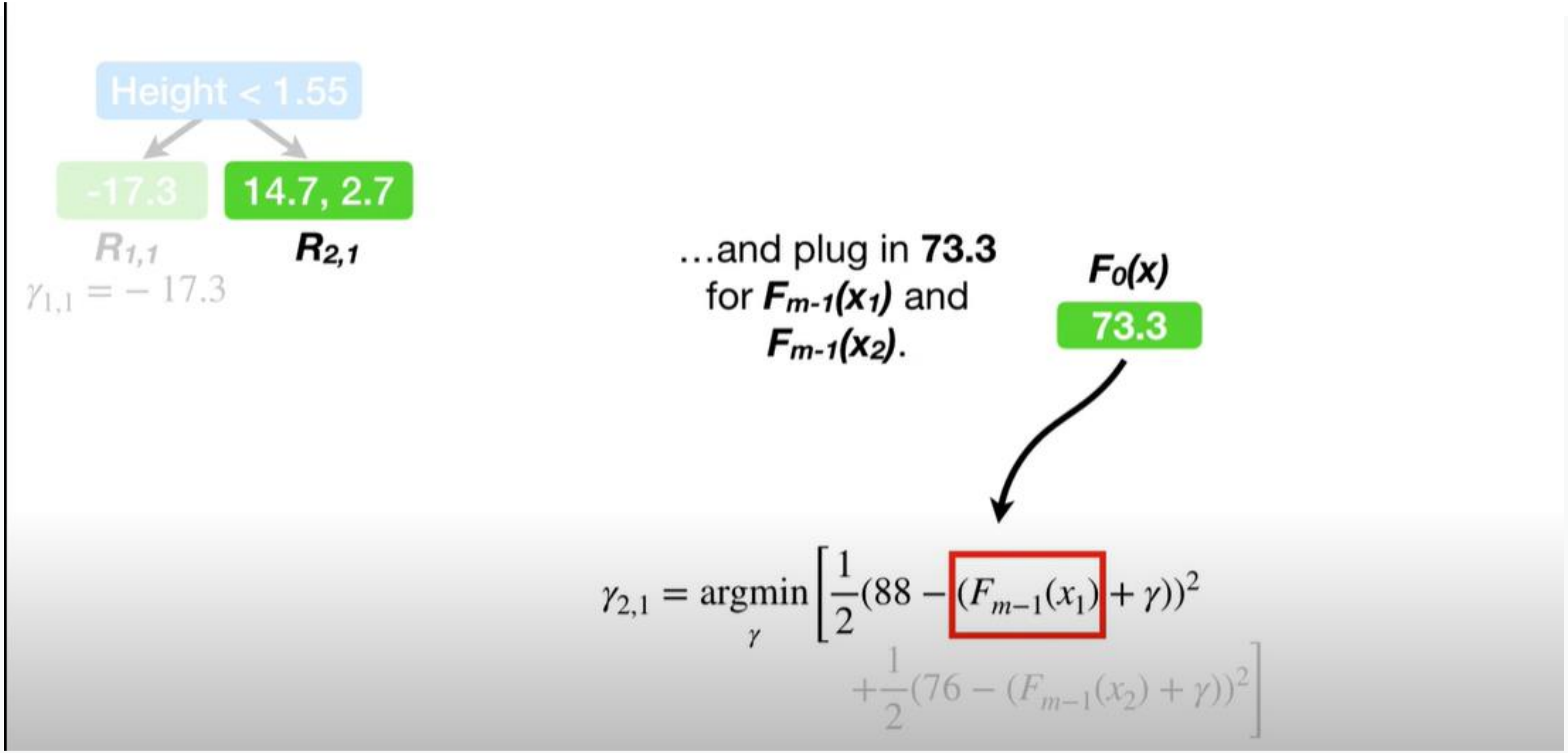


Gradient boosting : Step 2(C) Optimize leaf output values

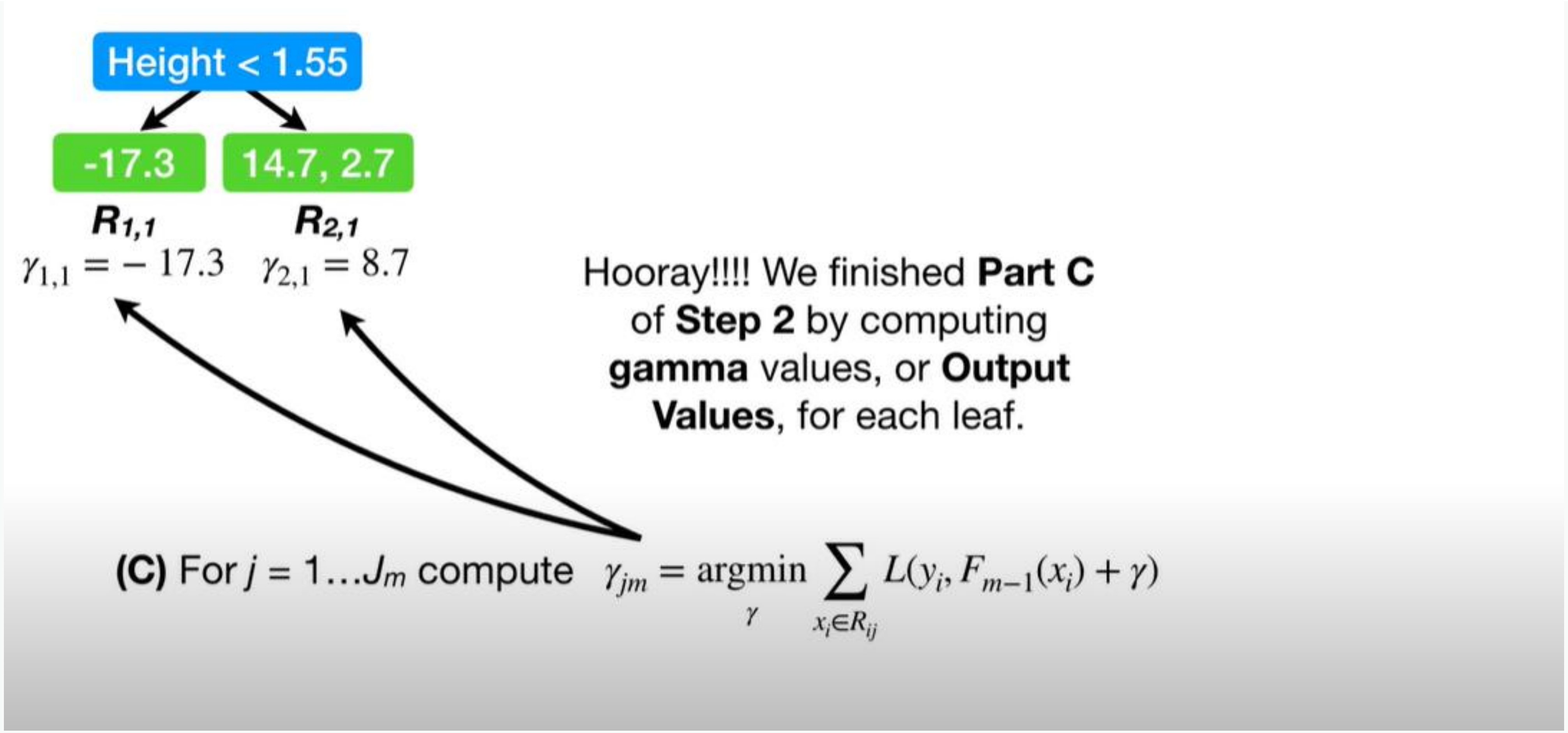
Specifically, since two residuals ended up in this leaf, it's unclear what its **Output Value** should be.

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

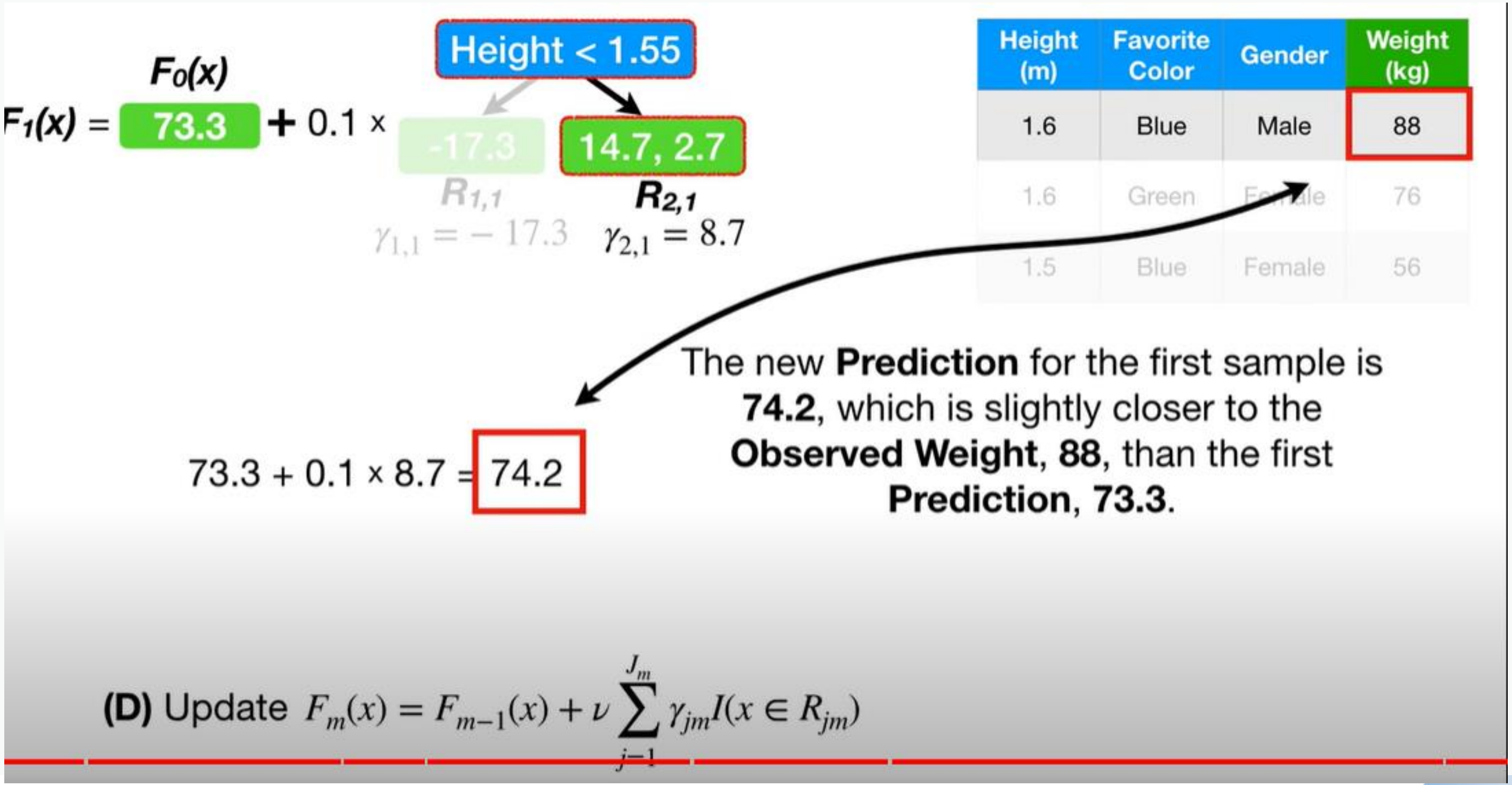
Gradient boosting : Step 2(C) Optimize leaf output values



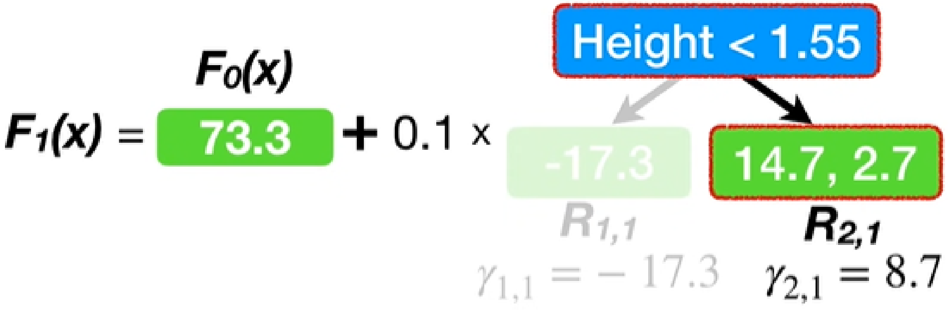
Gradient boosting : Step 2(C) Optimize leaf output values



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



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



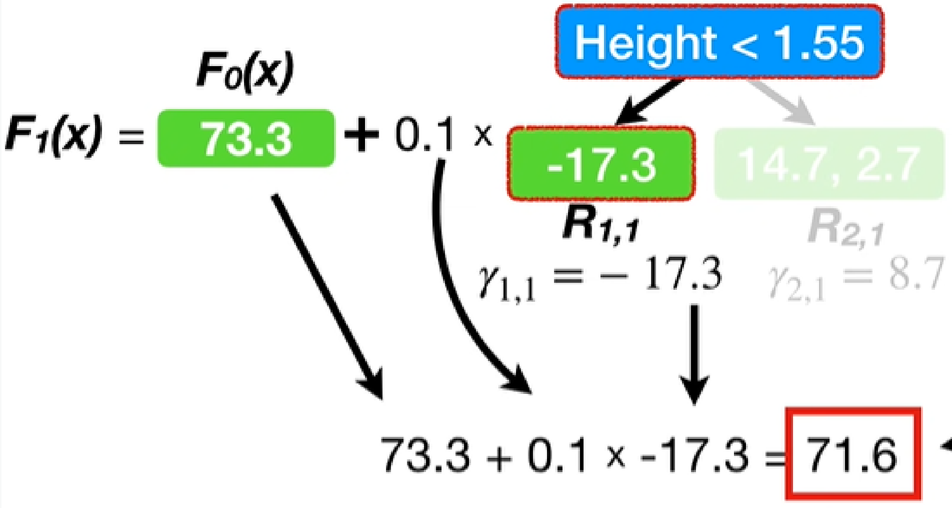
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	$\boxed{76}$
1.5	Blue	Female	56

$73.3 + 0.1 \times 8.7 = \boxed{74.2}$

The new **Prediction** for $\mathbf{x_2}$ 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



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

The new **Prediction** is **71.6**, which is an improvement over **73.3**.

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

G

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 $i = 1,\ldots,n$

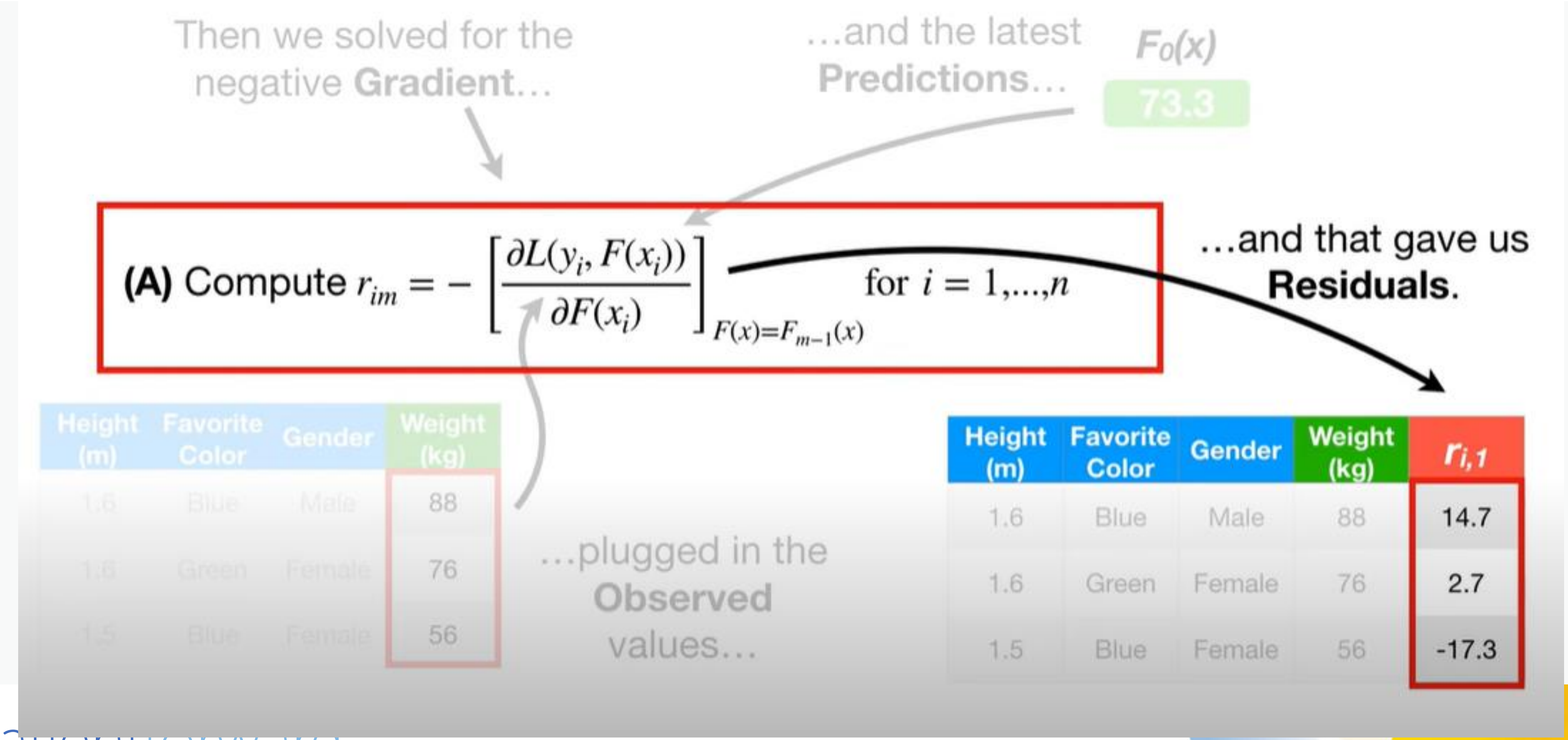
(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm}, for $j = 1\ldots J_m$

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

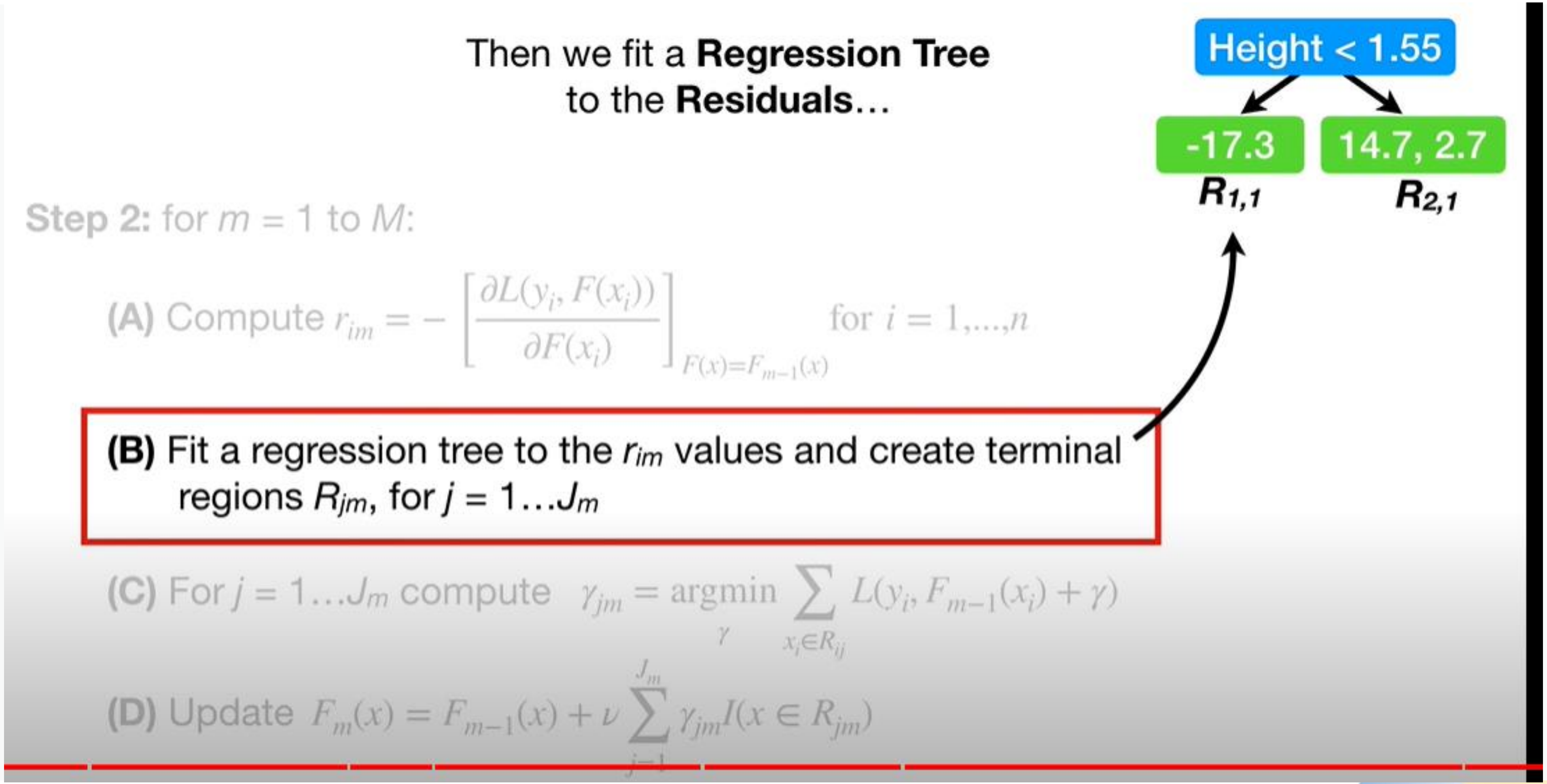
(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

mmary of step 2 >

Gradient boosting : Recap

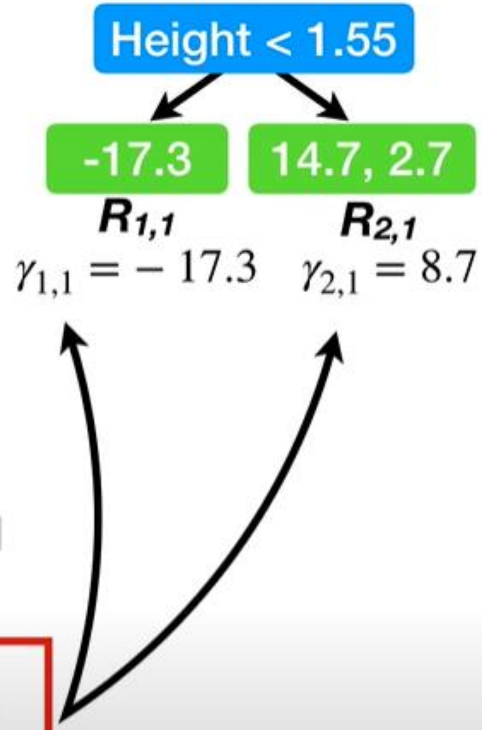


Gradient boosting : Recap



Gradient boosting : Recap

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



Step 2: for m = 1 to M:

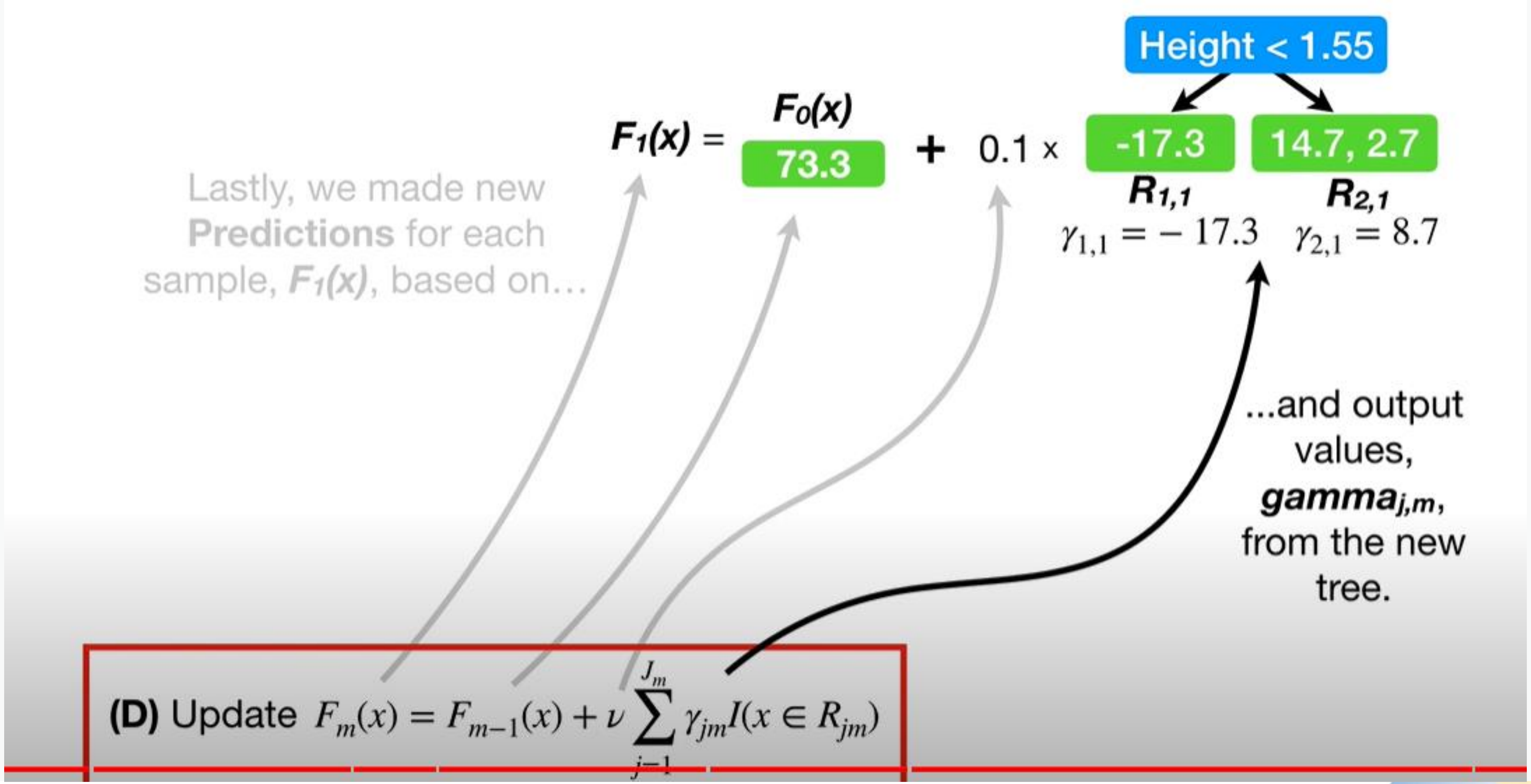
(A) Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1,\ldots,n$

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

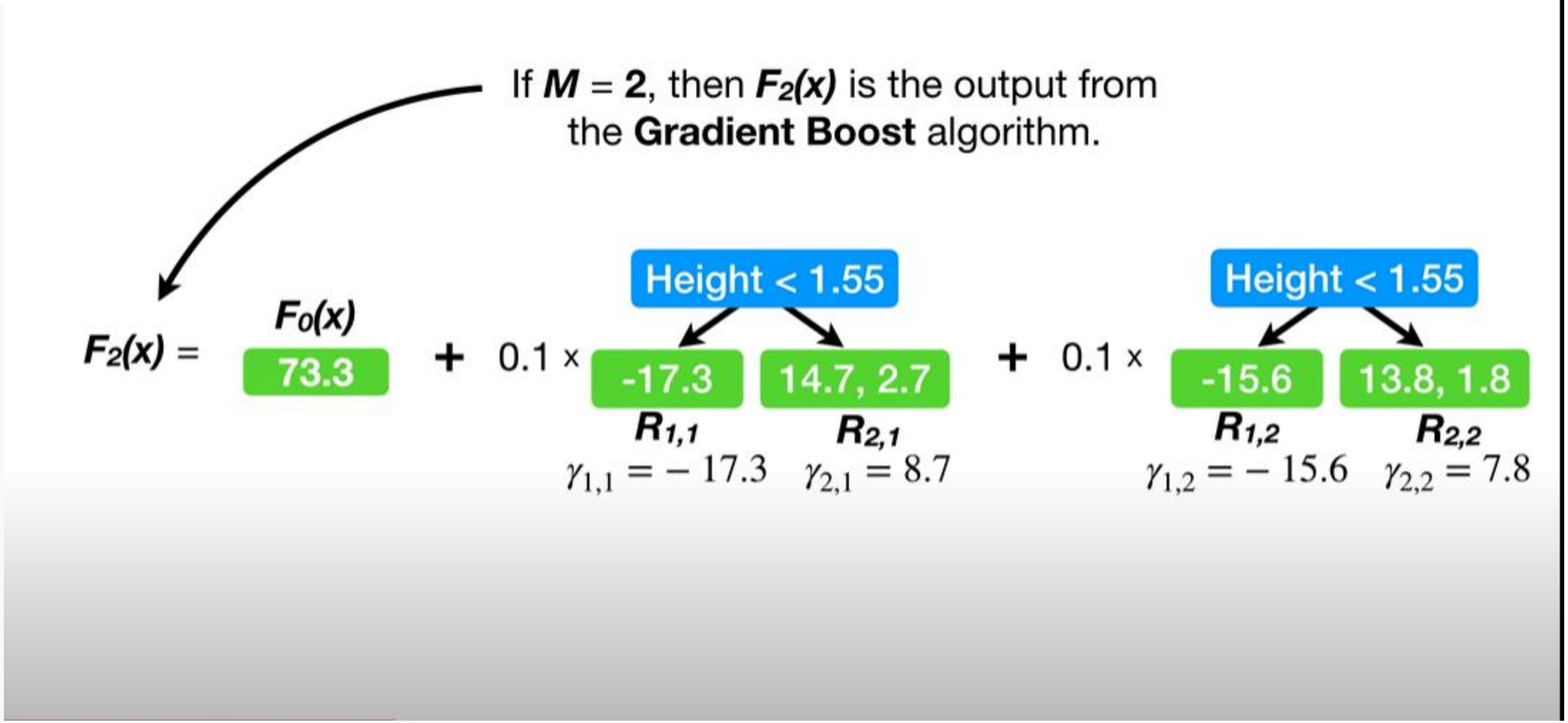
(C) For $j = 1\ldots J_m$ compute $\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

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

Gradient boosting : Recap



Gradient boosting : Recap



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:

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

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:

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

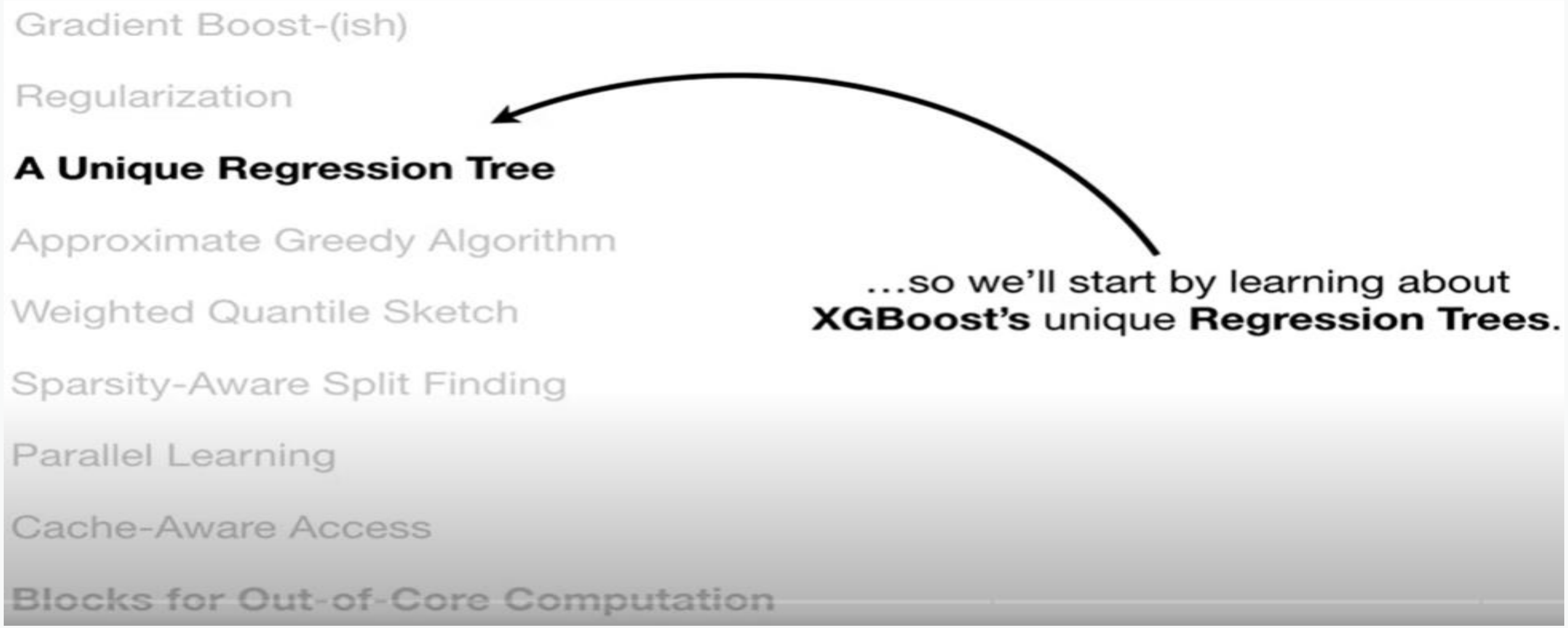
$$\hat{f}_m(x) = \alpha \hat{\phi}_m(x).$$

   3. Update the model:

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

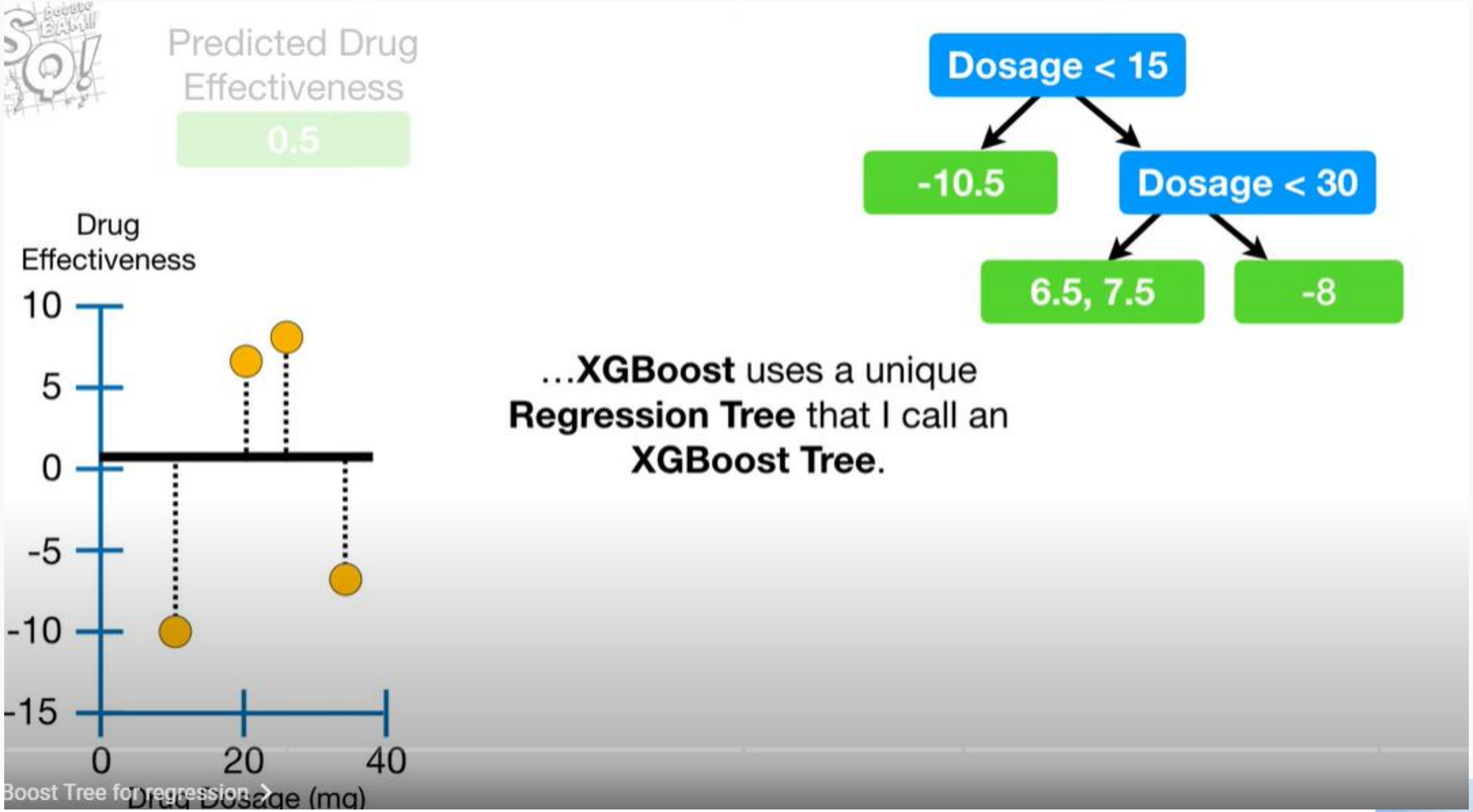
3. Output $\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x).$

Extreme !!! Gradient boosting



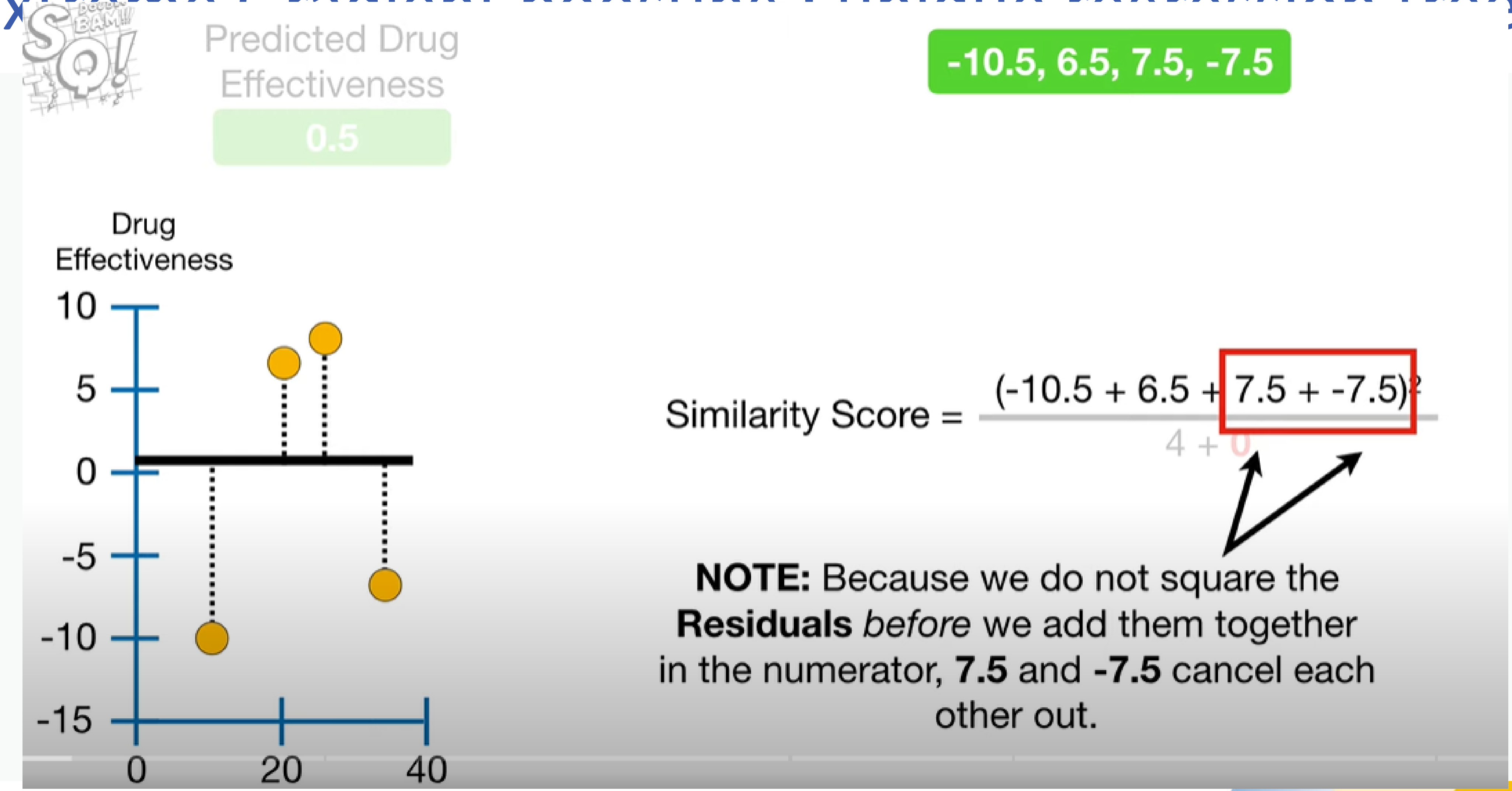
novocure™

Extreme Gradient boosting : unique regression tree

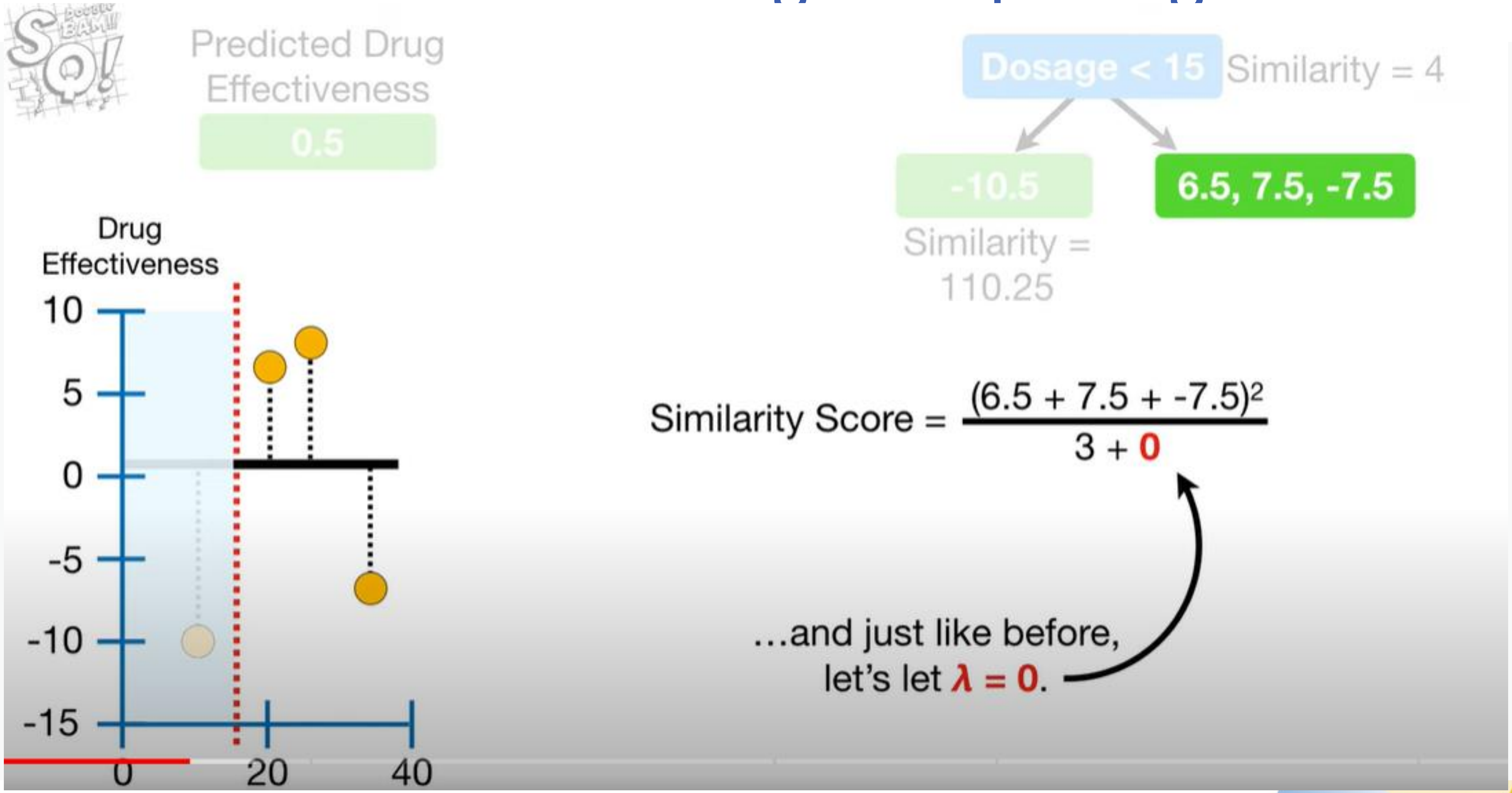


patientforward

Extreme Gradient boosting : unique regression tree

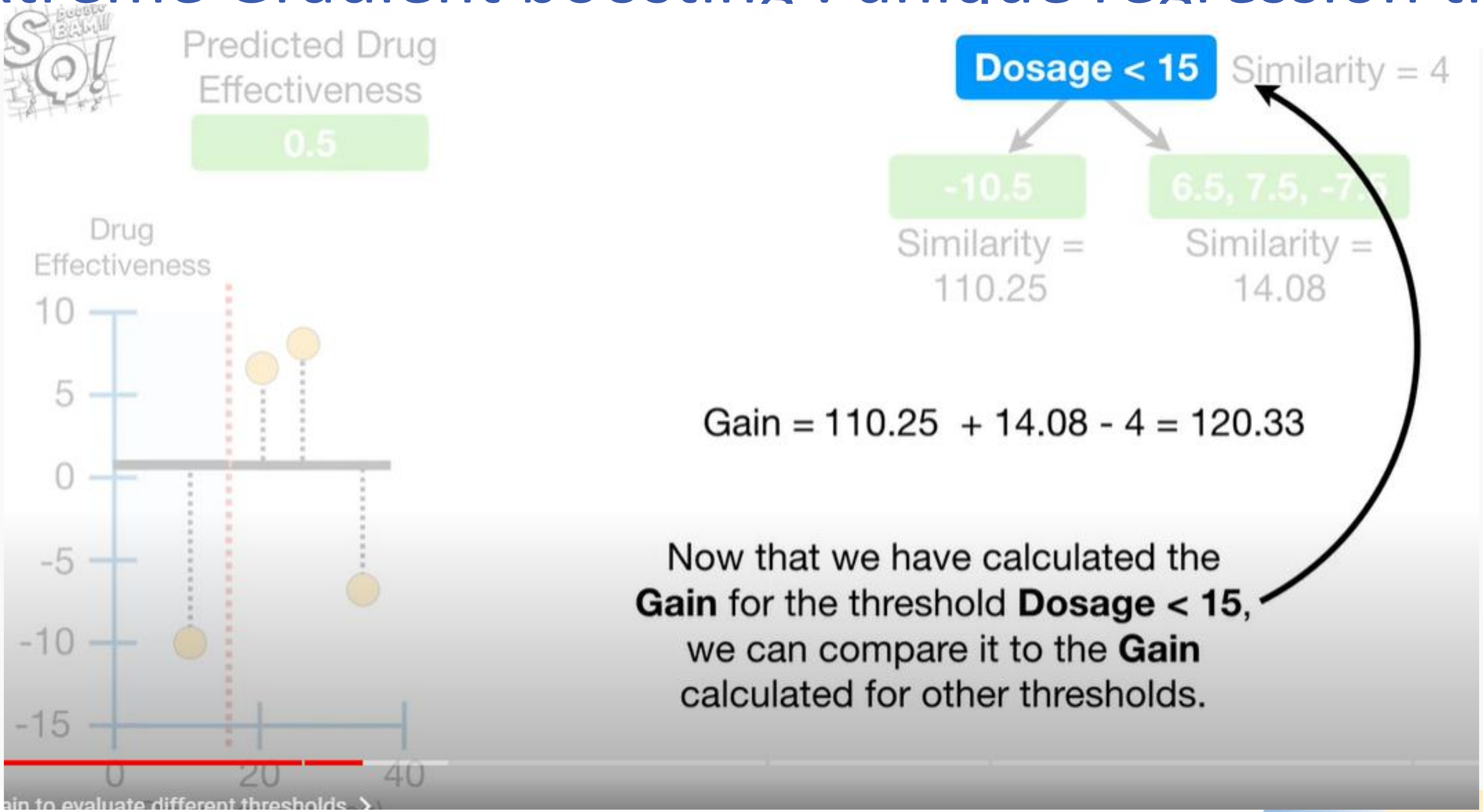


Extreme Gradient boosting : unique regression tree

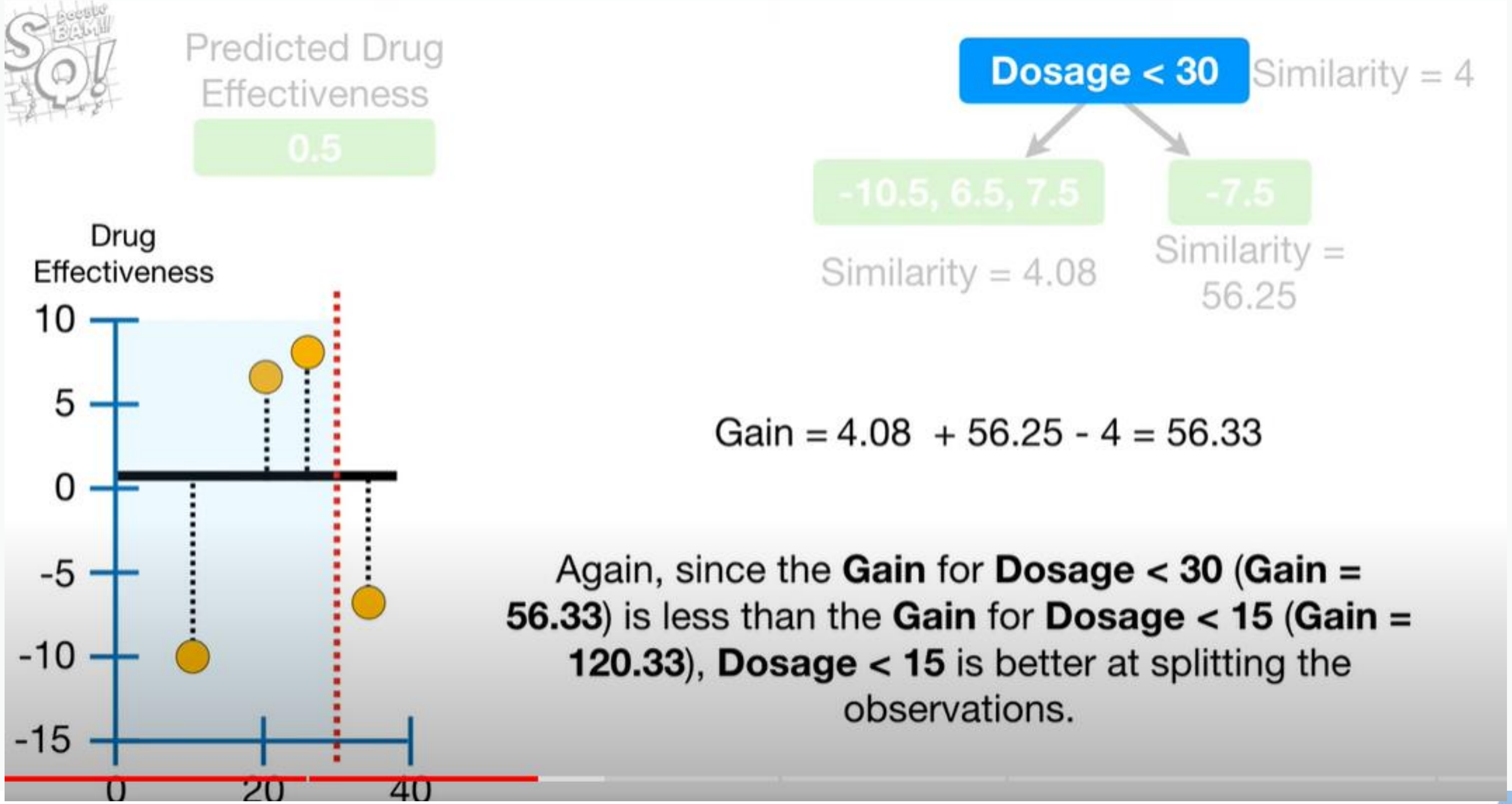


Extreme Gradient boosting : unique regression tree

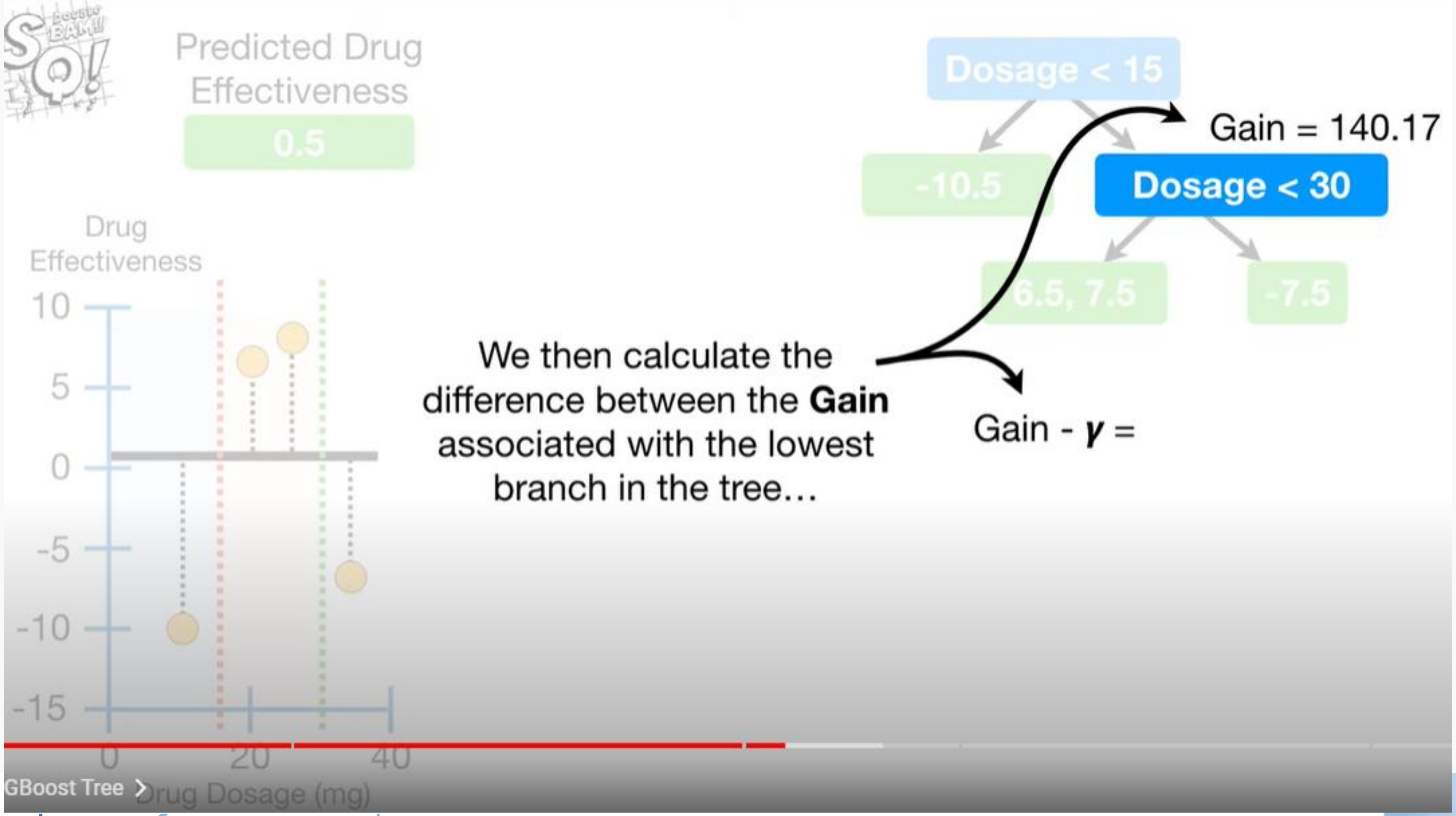
Extreme Gradient boosting : unique regression tree



Extreme Gradient boosting : unique regression tree



Extreme Gradient boosting : Pruning



Extreme Gradient boosting : Pruning