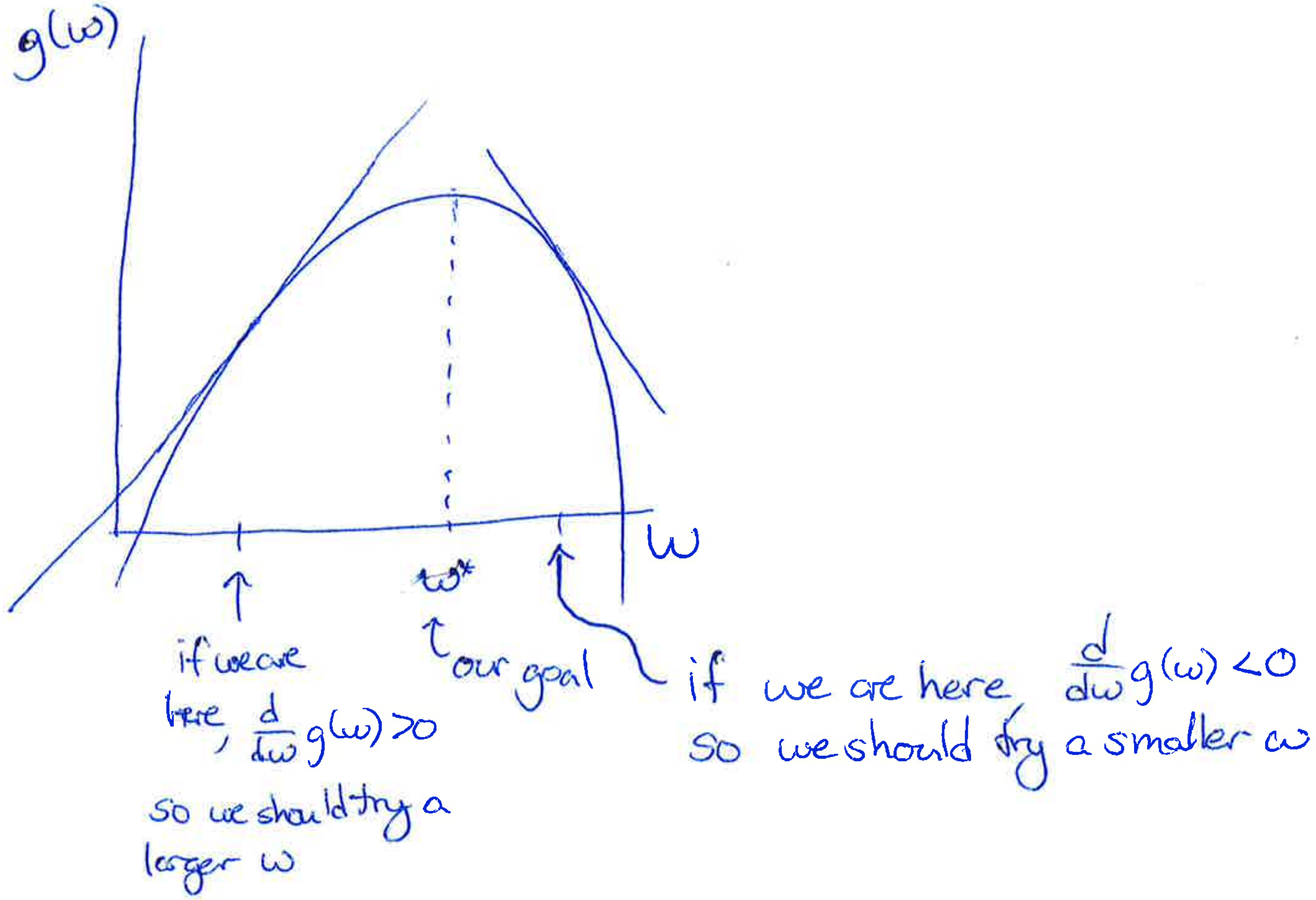
Mathematical Details for Gradient Boosting (Regression)

Note: If we're trying to maximize a function of ω, $g(\omega)$, the derivative tells us the direction we should move ω:



Above we fit each new component model to the residuals from the current ensemble.

What does this have to do with a gradient?

We want to minimize $RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

Equivalent to maximizing $-\frac{1}{2} RSS = -\frac{1}{2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$$= -\frac{1}{2} \{ (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \cdots + (y_n + \hat{y}_n)^2 \}$$

Taking the derivative of this wrt $\hat{y}_{i^*}$ tells us how we can improve fit to training data by changing the predicted value for observation i^*.

If $\frac{d}{d\hat{y}_{i^*}} -\frac{1}{2} RSS > 0$, fit would be improved by making the current prediction for $\hat{y}_{i^*}$ larger.

$$\frac{\partial}{\partial \hat{y}_{i^*}} -\frac{1}{2} RSS = \frac{\partial}{\partial \hat{y}_{i^*}} \frac{-1}{2} \left[(y_1 - \hat{y}_1)^2 + \cdots + (y_{i^*} - \hat{y}_{i^*})^2 + \cdots + (y_n - \hat{y}_n)^2 \right]$$

$$= \frac{-1}{2} \cdot 2 (y_{i^*} - \hat{y}_{i^*}) (-1$$

$$= y_{i^*} - \hat{y}_{i^*}$$

= residual for obs. i^* based on current ensemble.

Suppose our current prediction is too small:

- $y_{i*} > \hat{y}_{i*}$
- $y_{i*} - \hat{y}_{i*} > 0$ (positive residual)
- derivative of $\frac{-1}{2}$ RSS > 0
  - $\rightarrow$ we can increase $\frac{-1}{2}$ RSS by making predicted value larger
- A bigger difference between y_{i*} and $\hat{y}_{i*}$ means a larger derivative, bigger change in $\hat{y}_{i*}$ needed.

Suppose current prediction is too large:

- $y_{i*} < \hat{y}_{i*}$
- $y_{i*} - \hat{y}_{i*} < 0$ (negative residual)
- derivative of $\frac{-1}{2}$ RSS < 0
- we can increase $\frac{-1}{2}$ RSS by making predicted value smaller.

Another view:

After iteration b, our predicted value for y_{i*} is

$$\hat{f}^{(b)}(x_{i*}) = \hat{g}^{(1)}(x_{i*}) + \hat{g}^{(2)}(x_{i*}) + \cdots + \hat{g}^{(b)}(x_{i*}) = \sum_{j=1}^{b} \hat{g}^{(j)}(x_{i*})$$

In iteration b+1, we will add one more component model with prediction $\hat{g}^{(b+1)}(x_{i*})$

If we fit the training data perfectly, we would have

$$\hat{f}^{(b)}(x_{i*}) + \hat{g}^{(b+1)}(x_{i*}) = y_{i*} \Rightarrow \hat{g}^{(b+1)}(x_{i*}) = y_{i*} - \hat{f}^{(b)}(x_{i*})$$

$\Rightarrow$ we fit the residual from current ensemble!

Full Statement of Gradient Boosting Procedure:

1. Start with a "null ensemble"
   - just predicts mean training set response or 0 for all observations

2. For $b = 1, \ldots, B$ ← # of boosting iterations

   a. Calculate gradient vector of $\frac{-1}{2}$ RSS with respect to predicted values evaluated at current ensemble predictions

   $$\nabla_{\hat{y}} \frac{-1}{2} RSS = \frac{-1}{2}\left(\frac{\partial}{\partial \hat{y}_1} RSS, \ldots, \frac{\partial}{\partial \hat{y}_n} RSS\right)$$

   $$= (y_1 - \hat{y}_1, \ldots, y_n - \hat{y}_n)$$

   from current ensemble

   $$= (r_1, \ldots, r_n)$$

   b. Fit a new component model using the vector $(r_1, \ldots, r_n)$ as the response

   c. Add new component model to ensemble.

Tuning parameters to prevent overfitting:

- Learning rate: multiply predictions from each new component model by a small weight like 0.01.
  Prevents immediate overfitting
- Number of boosting iterations:
  The more boosting iterations higher potential for overfitting
- Minimum reduction in RSS:
  When growing a tree, how big does reduction in RSS need to be to make that split?
- Tree depth: deeper tree means more capacity to overfit
- Train on fewer observations: similar to bagging
- Train on fewer features: each component model trained using a subset of available explanatory variables.

Note: scaling factor of $\frac{1}{2}$ in our derivation above is not important, especially if we use a learning rate.