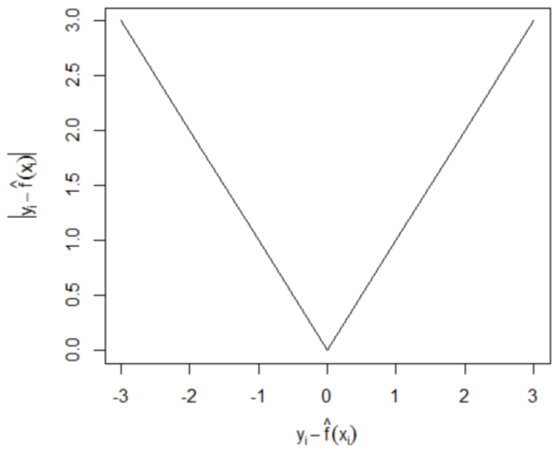
Computing the Negative Gradient of Absolute Error

$$
\begin{aligned}
-\frac{\partial}{\partial \hat{f}(x_i)}\left|y_i - \hat{f}(x_i)\right| &= -\frac{y_i - \hat{f}(x_i)}{\left|y_i - \hat{f}(x_i)\right|}\left(\frac{\partial}{\partial \hat{f}(x_i)}\left(y_i - \hat{f}(x_i)\right)\right) \\
&= -\frac{y_i - \hat{f}(x_i)}{\left|y_i - \hat{f}(x_i)\right|}\left(\frac{\partial}{\partial \hat{f}(x_i)}\left(y_i\right) - \frac{\partial}{\partial \hat{f}(x_i)}\left(\hat{f}(x_i)\right)\right) \\
&= -\frac{y_i - \hat{f}(x_i)}{\left|y_i - \hat{f}(x_i)\right|}\left(0-1\right) \\
&= \frac{y_i - \hat{f}(x_i)}{\left|y_i - \hat{f}(x_i)\right|}
\end{aligned}
$$

While the derivative is not defined when the prediction equals the actual response (because 0/0 is not defined), the gradient boosting machine (gbm) package uses the following function for the negative gradient of absolute error:

$$
sign\left(y_i - \hat{f}(x_i)\right) = \begin{Bmatrix} -1 & \text{if } y_i < \hat{f}(x_i) \\ 0 & \text{if } y_i == \hat{f}(x_i) \\ 1 & \text{if } y_i > \hat{f}(x_i) \end{Bmatrix}
$$

If our prediction is larger than the actual response, we want to decrease our prediction. If our prediction is precisely equal to the actual value, we want to use zero as our residual error (no adjustment needed). If our prediction is smaller than the actual response, we want to increase our prediction. Reminder: the gradient points us in the direction of greatest increase for our function, so we use the negative gradient to minimize the value of the error function.



P.S. The Huber loss function was designed to behave like mean squared error in a small area around 0 and like absolute error when outside the small area around 0. This makes it less sensitive to outliers than mean squared error, but also yields a smooth function around 0.