Lecture 23:

$$\hat{G}_{m-1}(x) = \sum_{i=1}^{m-1} \beta_i \hat{f}_i(x)$$

Solving $\beta_m, \hat{f}_m = \underset{\beta, f_m}{\arg\min} \sum_n L(y_n, \hat{G}_{m-1}(x_n) + \beta f_m(x_n))$

$L(y,h) = e^{-yh}$

$$\sum_n \exp\left(-y_n\left(\hat{G}_{m-1}(x) + \beta f_m(x_n)\right)\right)$$

$$= \sum_n \underbrace{\exp\left(-y_n \hat{G}_{m-1}(x)\right)}_{\text{no } \beta,\, f_m} \exp\left(-y_n \beta f_m(x_n)\right)$$

So I can regard as a weight w_{nm}

$$= \sum_n w_{nm} \underbrace{\exp(-\beta \underbrace{y_n f_m(x_n)}_{\pm 1})}_{e^{\pm\beta}}$$

$y_n = \pm 1$

$f_m(x_n) = \pm 1$

$$= \sum_{y_n = f_m(x_n)} w_{nm} e^{-\beta} + \sum_{y_n \neq f_m(x_n)} w_{nm} e^{\beta}$$

$$= \underbrace{e^{-\beta} \sum_{y_n = f_m(x_n)} w_{nm}}_{1} \underbrace{- e^{-\beta} \sum_{y_n \neq f_m(x_n)} w_{nm}}_{3} + \underbrace{e^{-\beta} \sum_{y_n \neq f_m(x_n)} w_{nm}}_{2} + \underbrace{e^{\beta} \sum_{y_n \neq f_m(x_n)} w_{nm}}_{4}$$

$$= ① + ② + ③ + ④$$

$$= e^{-\beta} \sum_n w_{nm} + (e^{\beta} - e^{-\beta}) \sum_{y_n \neq f_m(x_n)} w_{nm}$$

$$(*) = e^{-\beta} \sum_n w_{nm} + (e^{\beta} - e^{-\beta}) \underbrace{\sum_n w_{nm} \mathbb{1}(y_n \neq f_m(x_n))}_{\propto \text{ weighted mis-class err of } f_m}$$

Want to find opt. β, f_m to minimize

Fix β, find f_m to minimize:

$\hat{f}_m$ should minimize weighted mis-class err rate.

Given $\hat{f}_m$, find best β:

$$\frac{\partial}{\partial \beta}(*) = -e^{-\beta} \sum_n w_n + (e^{\beta} + e^{-\beta}) \sum_n w_{nn} \mathbb{1}(y_n \neq \hat{f}_m(x_n))$$

$= 0$ and solve for β

$$\Leftrightarrow \sum_n w_n = (e^{2\beta} + 1) \sum_n w_{nm} \mathbb{1}(y_n \neq \hat{f}_m(x_n))$$

$$\vdots$$

$$\beta_m = \frac{1}{2} \log\left(\frac{1 - err_m}{err_m}\right)$$

err_m = weighted mis-class

$$\alpha_m \triangleq 2\beta_m$$

Just need to show that we update weights as claimed.

$$\boxed{\text{AdaBoost: } w_{n,m+1} \leftarrow w_{nm} \exp\left(\alpha_m \mathbb{1}(y_n \neq \hat{f}_m(x_n))\right)}$$

Notice:

$$w_{n,m+1} = \exp\left(-y_n \hat{G}_m(x_n)\right)$$

$$= \exp\left(-y_n \hat{G}_{m-1}(x_n)\right) \exp\left(-\beta_m \hat{f}_m(x_n) y_n\right)$$

$$= w_{nm} \exp\left(-\beta_m \hat{f}_m(x_n) y_n\right)$$

notice that $\underbrace{-y_n \hat{f}_m(x_n)}_{\pm 1} = 2\mathbb{1}(y_n \neq \hat{f}_m(x_n)) - 1$

So

$$= w_{nm} \exp\left(2\beta_m \mathbb{1}(y_n \neq \hat{f}_m(x_n)) - \beta_m\right)$$

$$= w_{nm} \exp\left(\alpha_m \mathbb{1}(y_n \neq \hat{f}_m(x_n))\right) \underbrace{\exp(-\beta_m)}_{\text{doesn't matter}}$$

This is exactly AdaBoost.

For continuous y_n, can do a similar thing, by fitting each model to the residual of the previous.

This is equivalent to SAM using a squared err loss.

<u>A couple tuning choices:</u>

- loss *
- number of trees M
- Shrinkage factor: $\nu \in [0, 1]$

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

learning rate

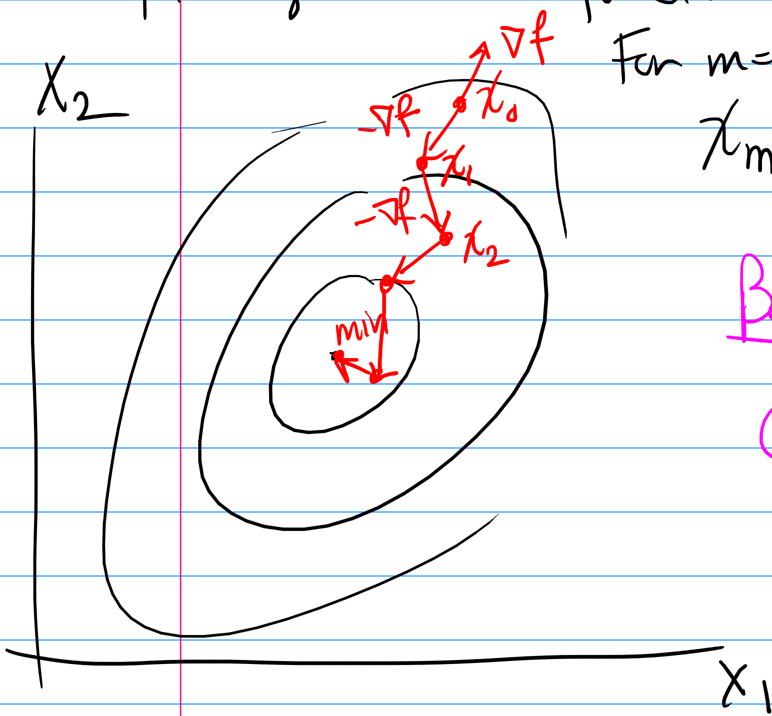
- number of splits in each of my trees

<u>Gradient Boosting</u>

Boosting using something other than SE loss (regr.) or exp. loss (class.)

Gradient Descent:

Optimize (minimize) some function f



For $m = 1, \ldots, M$

$$x_m = x_{m-1} - \alpha_m \nabla f \Big|_{x_{m-1}}$$

Boosting:

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

Looks like gradient descent if $\hat{f}_m \approx -\nabla f = -$gradient

$$\hat{G}_0 \xrightarrow{\hat{f}_1} \hat{G}_1 \xrightarrow{\hat{f}_2} \hat{G}_2 \xrightarrow{\hat{f}_3} \hat{G}_3 \quad \cdots \quad \bullet$$

arrive close to minimum of L

Gradient Boosting: (regression)

(0) $\hat{G}_0(x) = 0$

(2) For $m = 1, \ldots, M$

(a) $r_{nm} = -\left.\dfrac{\partial L}{\partial f(x_n)}\right|_{\hat{G}_{m-1}(x_n)}$ pseudo residuals

(b) fit $\hat{f}_m$ to predict r_{nm} from x_n

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

learning rate