Boosting "the statistical view"

$$\text{miscl error} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[y_i f(x_i) \le 0]}$$

$$\le \frac{1}{n}\sum_{i} e^{-y_i f(x_i)} \qquad \text{exponential loss}$$

choose f to be a linear model, a linear combination of "weak" classifiers.

$$f(x) = \sum_{j=1}^{p} \lambda_j h_j(x)$$

If I'm boring, $h_j(x) = x_{\cdot j}$, so $f(x) = \sum_{j=1}^{p} \lambda_j x_{\cdot j}$ as before.

$$R^{train}(\vec{\lambda}) = \frac{1}{n}\sum_i e^{-y_i \sum_j \lambda_j h_j(x_i)} = \frac{1}{n}\sum_i e^{-\sum_j \overbrace{y_i h_j(x_i)}^{M_{ij}} \lambda_j} = \frac{1}{n}\sum_i e^{-(\vec{M}\cdot\vec{\lambda})_i}$$

$$\text{where} \quad M = \begin{array}{c|c} & 1 \quad \cdots \quad p \\ 1 & \\ \vdots & \begin{bmatrix} & & \\ & y_i h_j(x_i) & \\ & & \end{bmatrix} \\ n & \end{array}$$

"matrix of margins"
margin of i^{th} point for j^{th} weak classifier

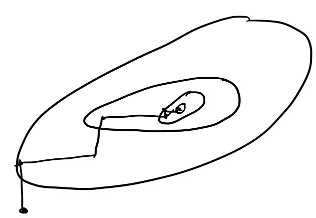
Assume all weak classifiers are binary, so $h_j(x_i) = \pm 1$
Then $M_{ij} = \pm 1$

$$R^{train}(\vec{\lambda}) = \frac{1}{n}\sum_i e^{-(\vec{M}\cdot\vec{\lambda})_i}$$

Do coordinate descent for $\vec{\lambda}$

Step 1: "coordinates" are j's. Find steepest coordinate.

Step 2: do a linesearch in that direction



Step 1. Say we are at $\bar{\lambda}_t$ and want steepest direction.

$$j_t \in \underset{j}{\operatorname{argmax}} \underbrace{\left[- \frac{d R^{train}(\vec{\lambda} + \alpha \vec{e}_j)}{d\alpha} \Bigg|_{\alpha = 0} \right]}$$

"Frechét derivative"

find steepest direction j_t

$$\vec{e}_j = \begin{array}{c} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{array} \begin{array}{c} \\ \\ \\ j \\ \\ \\ \end{array}$$

$$- \frac{d}{d\alpha} \left(\frac{1}{n} \sum_i e^{-[\bar{M}(\bar{\lambda}_t + \alpha e_j)]_i} \right) \Bigg|_{\alpha = 0}$$

$$- \frac{d}{d\alpha} \left(\frac{1}{n} \sum_i e^{-(\bar{M}\bar{\lambda}_t)_i - \alpha M_{ij}} \right) \Bigg|_{\alpha = 0}$$

$$\frac{1}{n} \sum_i \left[- \frac{d}{d\alpha} \left(e^{-(\bar{M}\bar{\lambda}_t)_i - \alpha M_{ij}} \right) \Bigg|_{\alpha = 0} \right]$$

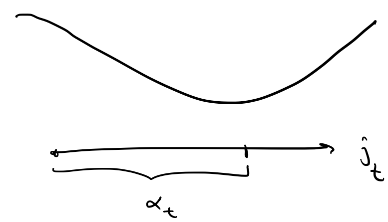
$$\frac{1}{n} \sum_i + M_{ij} \, e^{-(\bar{M}\bar{\lambda}_t)_i} + 0$$

$$j \in \underset{j}{\operatorname{argmax}} \frac{1}{n} \sum_i M_{ij} \, e^{-(\vec{M}\bar{\lambda}_t)_i}$$

"steepest direction"

Step 2: Linesearch along direction j

$$0 = \frac{dR^{train}(\vec{\lambda}_t + \alpha \vec{e}_{j_t})}{d\alpha}\Bigg|_{\alpha_t} \leftarrow \text{how far to go}$$



$$= \frac{d}{d\alpha} \frac{1}{n} \sum_i e^{-(\bar{M}(\lambda_t + \alpha e_{j_t}))_i}\Bigg|_{\alpha_t}$$

$$= \frac{1}{n} \sum_i \frac{d}{d\alpha}\left[e^{-(\bar{M}\bar{\lambda}_t)_i - \alpha M_{ij_t}}\right]\Bigg|_{\alpha_t}$$

$$= \frac{1}{n} \sum_i (-M_{ij_t}) e^{-(M\lambda_t)_i - \alpha M_{ij_t}}\Bigg|_{\alpha_t}$$

$$= -\frac{1}{n} \sum_{i:M_{ij_t}=1} M_{ij_t} e^{-(M\lambda_t)_i - \alpha M_{ij_t}} - \frac{1}{n} \sum_{i:M_{ij_t}=-1} M_{ij_t} e^{-(M\lambda_t)_i - \alpha M_{ij_t}}\Bigg|_{\alpha_t}$$

$$= -\frac{1}{n} \sum_{i:M_{ij_t}=1} e^{-(\bar{M}\bar{\lambda}_t)_i} e^{-\alpha} - \frac{1}{n} \sum_{i:M_{ij_t}=-1} -e^{-(M\lambda_t)_i} e^{\alpha}\Bigg|_{\alpha_t}$$

define $d_{t,i} = \dfrac{e^{-(M\lambda)_i}}{Z_t} \leftarrow$ normalization

$$0 = -\sum_{i:M_{ij_t}=1} d_{t,i} e^{-\alpha} - \sum_{i:M_{ij_t}=-1} -d_{t,i} e^{\alpha}\Bigg|_{\alpha_t}$$

$$= -e^{-\alpha_t} \underbrace{\sum_{i:M_{ij_t}=1} d_{t,i}}_{d_+} + e^{\alpha_t} \underbrace{\sum_{i:M_{ij_t}=-1} d_{t,i}}_{d_-}$$

$$= -e^{-\alpha_t} d_+ + e^{\alpha_t} d_-$$

$$e^{-\alpha_t} d_+ = e^{\alpha_t} d_-$$

$$\frac{d_+}{d_-} = e^{2\alpha_t}$$

$$\frac{1}{2} \ln\left(\frac{d_+}{d_-}\right) = \alpha_t \quad \leftarrow \text{Since } d\text{'s are normalized } d_+ = 1 - d_-$$

$$\boxed{\alpha_t = \frac{1}{2} \ln\left(\frac{1-d_-}{d_-}\right)} \quad ☺$$

Simplify one last thing: Step 1: $j_t \in \operatorname{argmax}_j \frac{1}{n} \sum_i M_{ij} e^{-(\bar{M}\bar{\lambda}_t)_i}$

$$\boxed{j_t \in \operatorname{argmax} \underbrace{\sum_i M_{ij} d_{t,i}}_{(\bar{d}_t^T M)_j}}$$

Finally the algorithm:

$$d_{1,i} = 1/n \quad i = 1 \ldots n$$

$$\vec{\lambda}_1 = \vec{0}$$

for $t = 1 \ldots T$

Step 1: $j_t \in \underset{j}{\operatorname{argmax}} \left(\vec{d}_t^T \bar{M}\right)_j$

Replaced in practice by weak learning algorithm.

$$j_t \in \underset{j}{\operatorname{argmax}} \left(\vec{d}_t^T \cdot M\right)_j$$

$$\underset{j}{\operatorname{argmax}} \left[\underbrace{\sum_{i: M_{ij} = 1} d_{t,i}} + \sum_{i: M_{ij} = -1} - d_{t,i} \right]$$

$$\underset{j}{\operatorname{argmax}} \left[1 - \sum_{i: M_{ij} = -1} d_{t,i} - \sum_{i: M_{ij} = -1} d_{t,i} \right]$$

$$\underset{j}{\operatorname{argmin}} \left[\sum_{i: M_{ij} = -1} d_{t,i} \right] \leftarrow$$ weak classifier minimizes "weight" of misclassified points

Notation: $d_- = \sum_{M_{ij_t} = -1} d_{t,i}$

Step 2: $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - d_-}{d_-} \right)$

Take the step: $\vec{\lambda}_{t+1} = \vec{\lambda}_t + \underset{\text{size}}{\alpha_t} \underset{\text{direction}}{\vec{e}_{j_t}}$

Notation: $d_{t+1,i} = e^{-(\bar{M}\vec{\lambda}_{t+1})_i} / Z_{t+1} \quad \forall i$

$$Z_{t+1} = \sum_i e^{-(\bar{M}\vec{\lambda}_{t+1})_i}$$

end

This ↑ is AdaBoost. Except for one thing

weight update: $d_{t+1,i} = e^{-(\bar{M}\vec{\lambda}_{t+1})_i} / Z_{t+1}$

$$\propto e^{-(\bar{M}\vec{\lambda}_t)_i} e^{-M_{ij_t}\alpha_t}$$

$$d_{t+1,i} \propto \begin{cases} d_{ti} \, e^{-\alpha_t} & \text{if } M_{ij_t} = 1 \\ d_{ti} \, e^{\alpha_t} & \text{if } M_{ij_t} = -1 \end{cases}$$