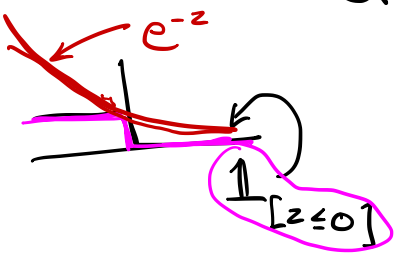
Boosting

Boosting - "the statistical view"

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

$\leq \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f(x_i)}$ exp 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 $f(x) = \sum_{j=1}^{p} \lambda_j x_{\cdot j}$

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

$(n \times p \cdot p \times 1)_i$

where $M =$ "matrix of margins"

	1		j		p
1					
i			$y_i h_j(x_i)$		
n					

margin of i^{th} point for j^{th} weak classifier.

Assume all weak classifiers are binary so $h_j(x_i) = \pm 1$

Then $M_{ij} = \pm 1$.

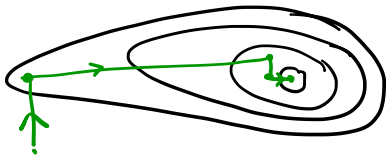
$$R^{train}(\vec{\lambda}) = \frac{+}{n} \sum_i e^{-(\tilde{M}\vec{\lambda})_i}$$

Do coordinate descent for λ

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

Step 2 - do a linesearch in that direction

Step 1: Say we are at λ_t and we want steepest direction.



$$j_t \in \underset{j}{\operatorname{argmax}} \left[- \frac{dR^{train}(\lambda_t + \alpha e_j)}{d\alpha} \Big|_{\alpha=0} \right]$$

"Frechet derivative"

find steepest direction j

$$e_j = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}_j$$

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

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

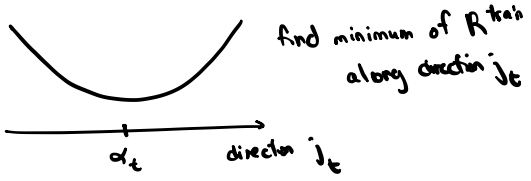
$$\frac{1}{n} \sum_i \left[- \frac{d}{d\alpha} \left(e^{-(\vec{M}\vec{\lambda}_t)_i - \alpha M_{ij}} \right) \Big|_{\alpha=0} \right.$$

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

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

"steepest direction"

Step 2: Line search along direction j_t:



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

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

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

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

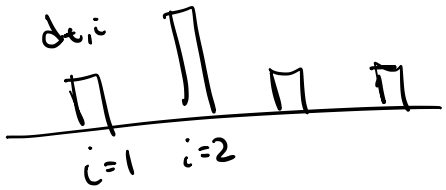
$$= -\frac{1}{n}\sum_{i: M_{ij_t}=1} M_{ij_t} e^{-(M\lambda_t)_i - \alpha M_{ij_t}} - \frac{1}{n}\sum_{i: M_{ij_t}=-1} M_{ij_t} e^{-(M\lambda_t)_i - \alpha M_{ij_t}}\Big|_{\alpha_t}$$

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

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

$$= -e^{-\alpha_t}\frac{[\quad]}{Z_t} + \frac{e^{\alpha_t}[\quad]}{Z_t} \quad \leftarrow \text{normalization}$$

define $d_{t,i} = \frac{e^{-(M\lambda_t)_i}}{Z_t}$



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

d_+: total weight for correctly classified points

$$0 = -e^{-\alpha_t} \underbrace{\sum_{i : M_{i j_t} = 1} d_{t,i}}_{d_+} + e^{\alpha_t} \underbrace{\sum_{i : M_{i j_t} = -1} d_{t,i}}_{d_-}$$

d_+: total weight for correctly classified points

d_-

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

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

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

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

Since Z_t was normalization

$$d_+ = 1 - d_-$$

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

Simplify one last thing:

Step 1: $j_t \in \underset{j}{\operatorname{argmax}} \left[\frac{1}{n} \sum_i M_{ij} e^{-(\bar{M} \bar{\lambda}_t)_i}\right]$

$$j_t \in \underset{j}{\operatorname{argmax}} \sum_i M_{ij} d_{t,i}$$

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

Finally the algorithm:

$d_{1,i} = \frac{1}{n}$ $i = 1 \dots n$

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

for $t = 1 \dots T$

Step 1: $j_t \in \underset{j}{\text{argmax}} \, (\vec{d}_t^{\,T} M)_j$

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: $\vec{\lambda}_{t+1} = \vec{\lambda}_t + \alpha_t e_{j_t}$

$$\begin{bmatrix} 3 \\ 12 \\ 17 \\ 4 \\ \vdots \end{bmatrix}_p = \begin{bmatrix} 3 \\ 12 \\ 17 \\ \vdots \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha_t \\ 0 \\ 0 \end{bmatrix}$$

λ_{t+1}

Notation: $d_{t+1,i} = e^{-(M\lambda_{t+1})_i} / Z_{t+1}$

$Z_{t+1} = \sum_i e^{-(M\lambda_{t+1})_i}$

This ↑ is AdaBoost. Except for one thing.

$j_t \in \underset{j}{\text{argmax}} \, (d_t^T M)_j$ is replaced by a "weak learning algorithm"

$$= \underset{j}{\text{argmax}} \left[\sum_{i : M_{ij} = 1} d_{ti} + \sum_{i : M_{ij} = -1} -d_{ti} \right]$$

$$= \underset{j}{\text{argmax}} \left[\overbrace{1 - \sum_{i : M_{ij} = -1} d_{ti}}^{i : M_{ij} = 1} - \sum_{i : M_{ij} = -1} d_{ti} \right]$$

$$= \underset{j}{\text{argmin}} \left[\sum_{i : M_{ij} = -1} d_{ti} \right]$$

← pick the weak classifier that minimizes the "weight" of misclassified points

One last bit of notation:

weight update: $d_{t+1} = e^{-(M\lambda_{t+1})_i} / Z_{t+1}$

$$\propto \left[e^{-(M\lambda_t)_i} e^{-M_{ij_t}\alpha_t} \right]$$

$$\propto d_{ti} \cdot e^{-M_{ij_t}\alpha_t}$$

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

multiplicative weight update