

# Ensemble Learning

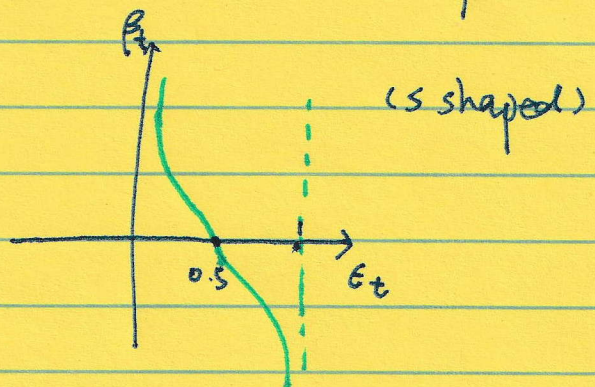
Idea: Construct a classifier  $H(x)$  that combines the individual decisions  $h_1, \dots, h_T$

## AdaBoost

1. Combines a lot of "weak learners" to make classification
2. Some weak learner get more say in the classification than others
3. Each weak learner is made by taking the previous weak learner's mistakes into account

Each sample has a "weight" or importance score. At the start, all samples get the same weight.

Amount of say of the  $t^{\text{th}}$  classifier (weak) =  $\frac{\beta_t}{2} \log \left( \frac{1 - \text{total error}}{\text{total error}} \right)$

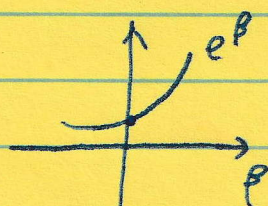
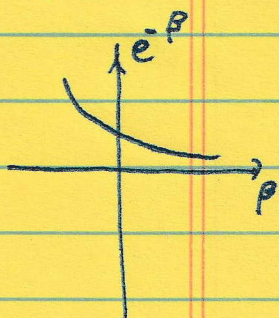


$$\epsilon_t \in [0, 1], \beta_t \in (-\infty, +\infty)$$

Update all sample weights based on if the current classifier (weak) misclassifies <sup>the samples</sup>

$$w_{t+1, i} = w_{t, i} \cdot \exp(-\beta_t y_i h_t(x_i))$$

if  $y_i = h_t(x_i) \Rightarrow h_t(x_i) \cdot y_i = 1 \Rightarrow w_{t, i} \cdot e^{-\beta} \Rightarrow e^{-\beta} \downarrow$  when  $\beta \uparrow$   
if  $y_i \neq h_t(x_i) \Rightarrow h_t(x_i) \cdot y_i = -1 \Rightarrow w_{t, i} \cdot e^{\beta} \Rightarrow e^{\beta} \uparrow$  when  $\beta \uparrow$



$$H(x) = \text{sign} \sum_{t=1}^T \beta_t h_t(x)$$



Training with weighted Instances.

$$J_{\text{reg}}(\theta) = - \sum_{i=1}^n w_i [y_i \log h_{\theta}(x_i) + (1-y_i) \log (1-h_{\theta}(x_i))] + \lambda \|\theta\|_2^2$$

Basically, prioritize samples with larger weights  $\Rightarrow$   
prioritize samples that have been misclassified repeatedly