Boğaziçi University EE Department

ee58J | 2012 – 2013 spring

Data Mining for Visual Media

Lecture VIII

Boosting

Ceyhun Burak Akgül, PhD in EE

www.cba-research.com

Boosting: Principles

- Boosting is a blueprint for supervised learning algorithms.

  Aggregate several weak classifiers into a single strong classifier with higher accuracy

- A **weak classifier** is one which performs only **slightly better than chance**.
- It's always easy to train a single classifier

AdaBoost

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

$h_t(x) : X \to \{-1, +1\}$ "Weak" or base classifier

$H(x) = sign\big(f(x)\big)$ "Strong" or final classifier

Properties

- AdaBoost avoids overfitting to a considerable extent: **reduces bias for simple classifiers** (e.g., stumps) and **variance for complex classifiers** (e.g., decision trees)
- AdaBoost also **maximizes the margin** (as in SVM)
- AdaBoost has a **built-in feature selection** mechanism

AdaBoost: The Algorithm

Given : $\{(x_i, y_i)\}_{i=1}^{m}$; $x_i \in X$ and $y_i \in \{-1, +1\}$

Initialize weights $D_1(i) = 1/m,$ for all $i = 1, \ldots, m$

For $t = 1, \ldots, T$:

1. **(Call WeakLearn) Find** $h_t = \arg\min_{h \in \mathrm{H}} \sum_{i=1}^{m} D_t(i)\left[y_i \neq h(x_i)\right]$

2. $\varepsilon_t = \sum_{i=1}^{m} D_t(i)\left[y_i \neq h_t(x_i)\right]$ then if $\varepsilon_t \geq 1/2$ then stop else continue

3. **Set** $\alpha_t = \frac{1}{2}\log\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$ then $D_{t+1}(i) = D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)$

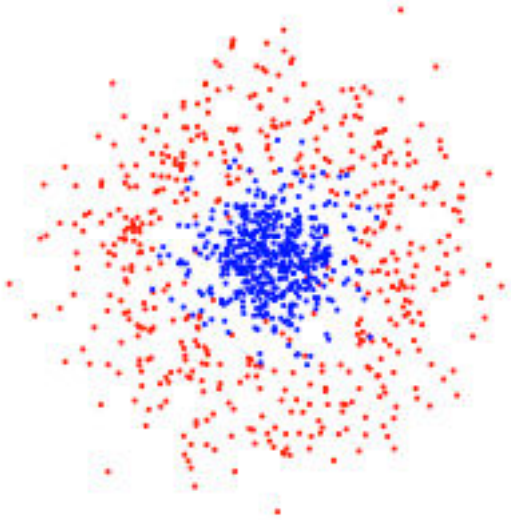
4. **Compute** $Z_t = \sum_{i=1}^{m} D_{t+1}(i)$ then **normalize** $D_{t+1}(i) \leftarrow D_{t+1}(i)/Z_t$ for all $i = 1, \ldots, m$

Output the final classifier :

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

AdaBoost: In Pictures

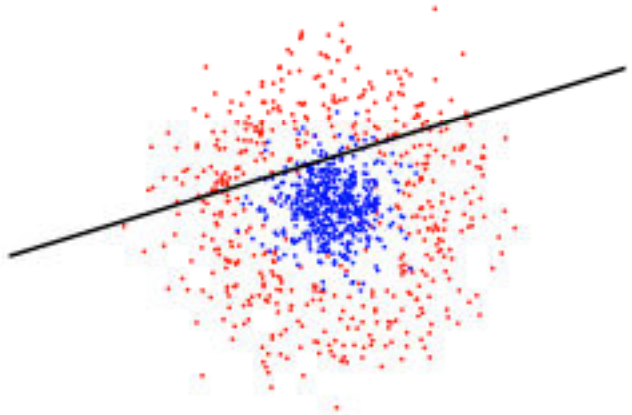
Training dataset



AdaBoost: In Pictures

Classifier at iteration

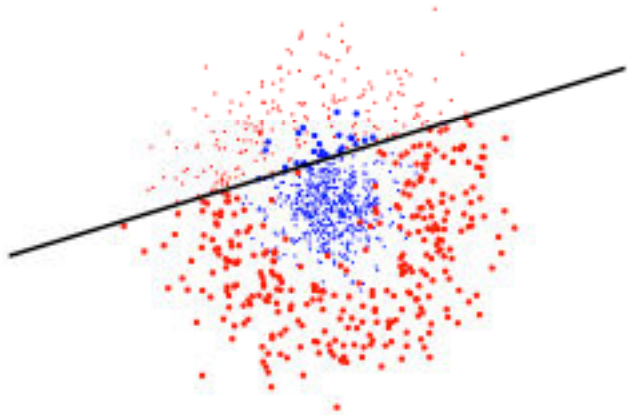
$$t = 1$$



AdaBoost: In Pictures

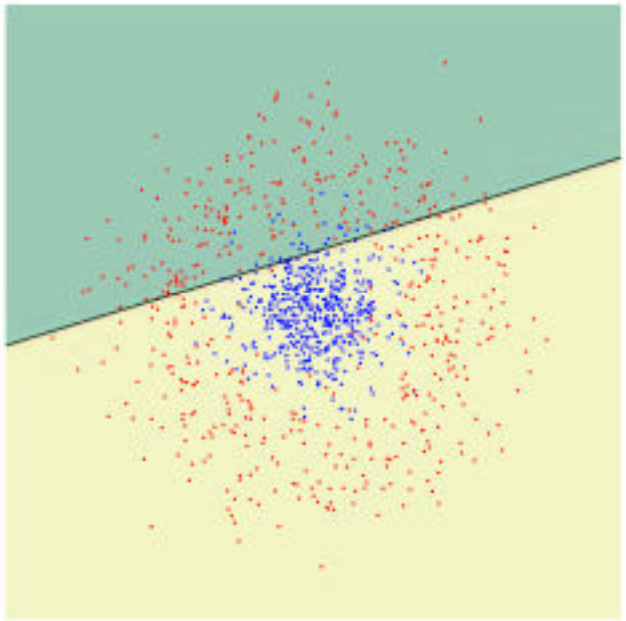
Training instances reweighted

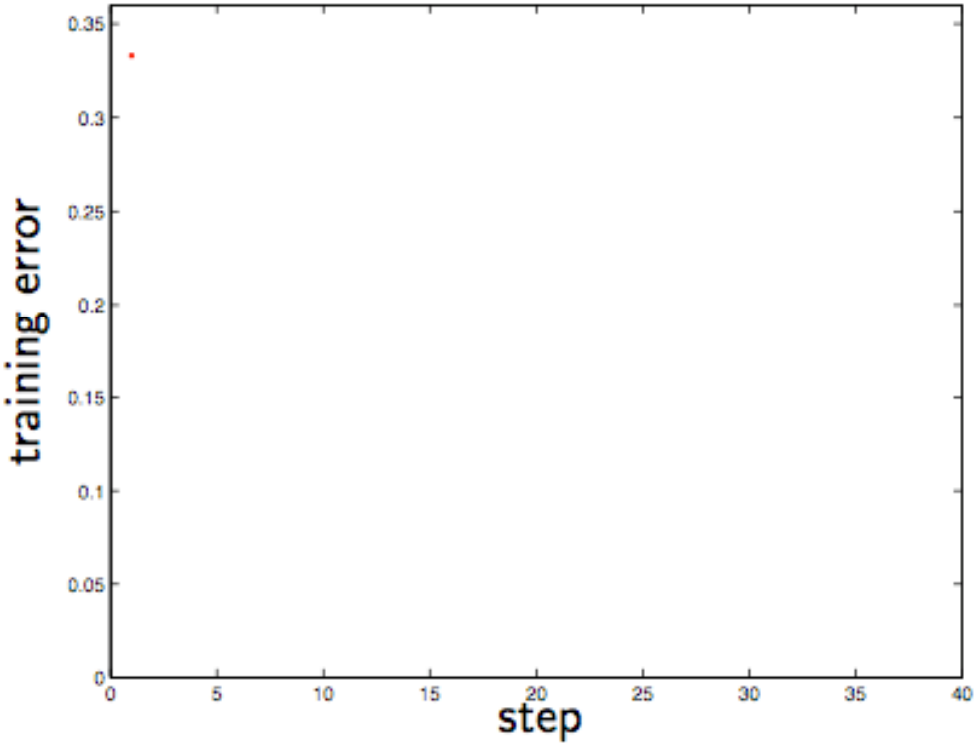
$$t = 1$$



AdaBoost: In Pictures

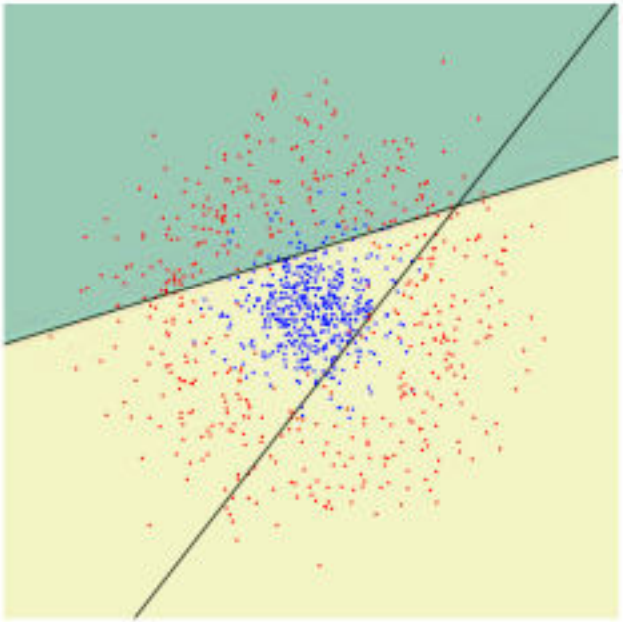
$t = 1$

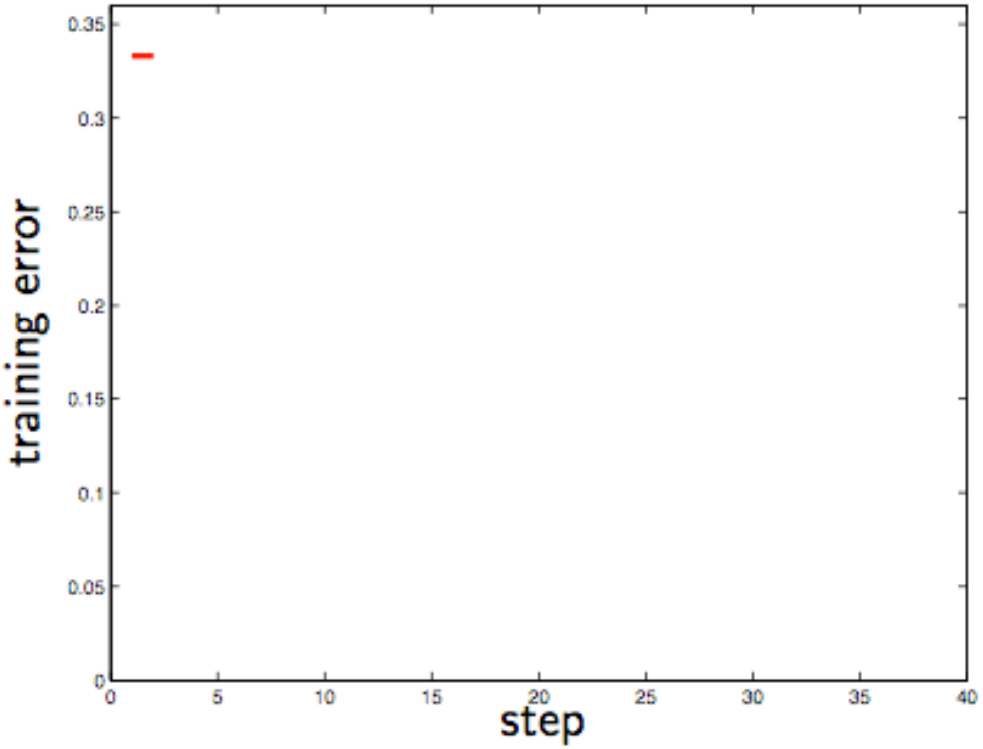




AdaBoost: In Pictures

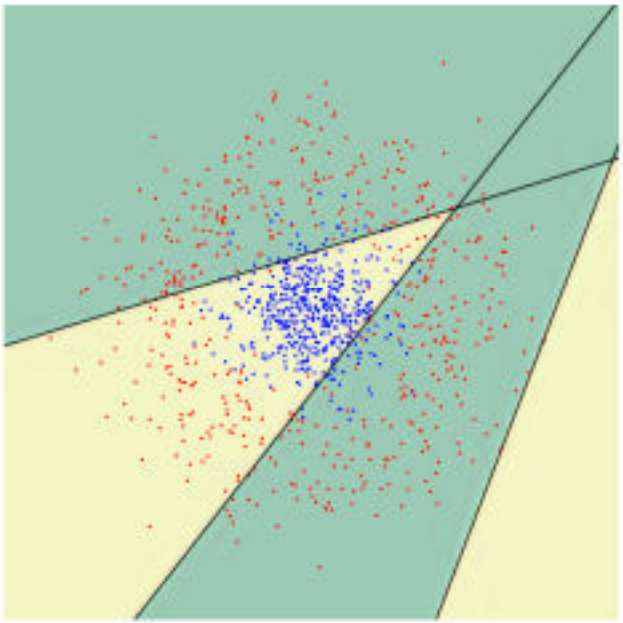
$t = 2$

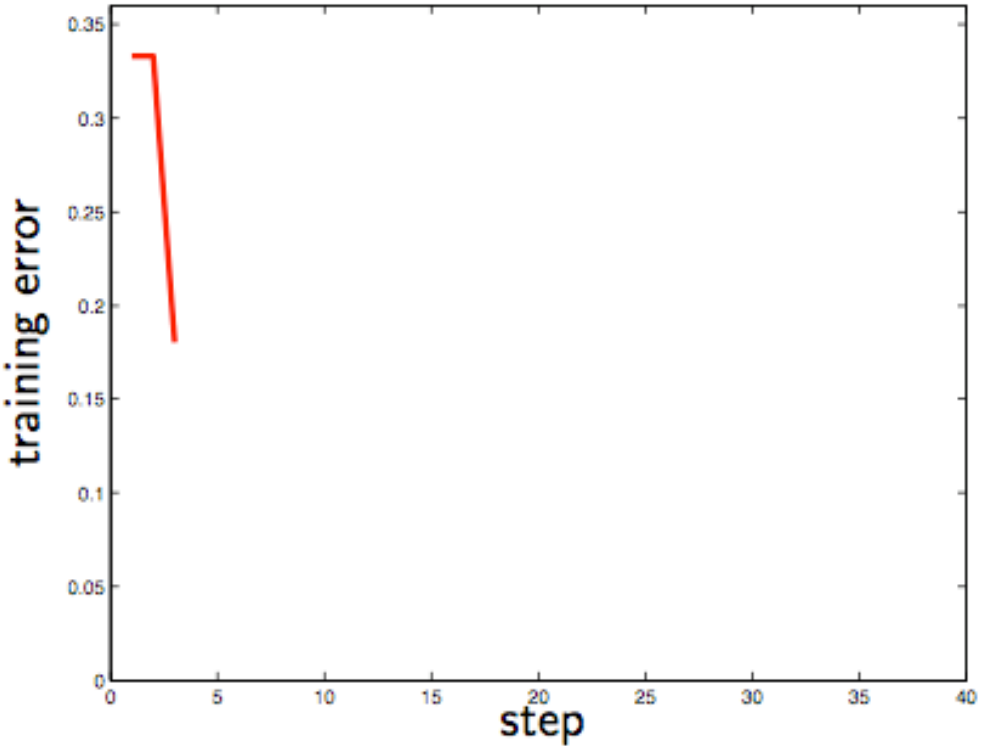




AdaBoost: In Pictures

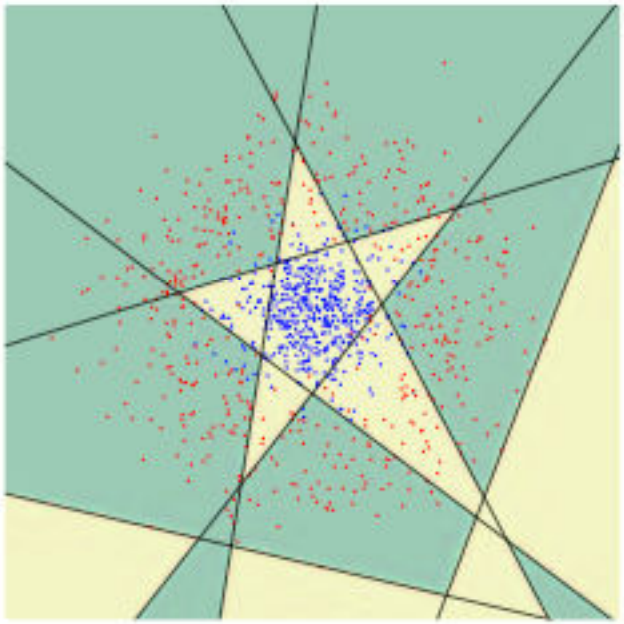
$t = 3$

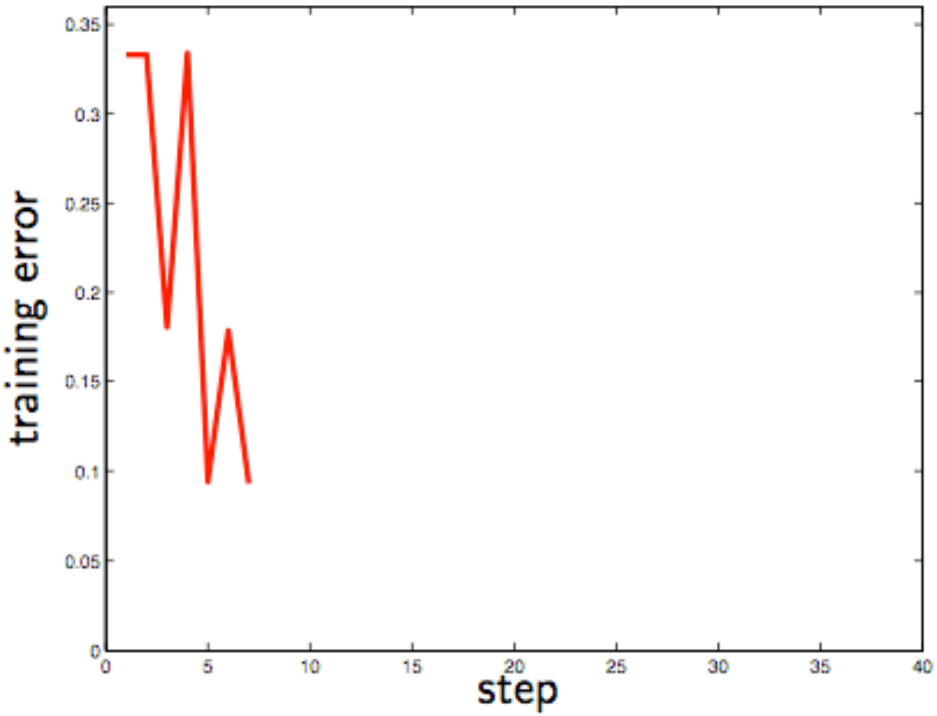




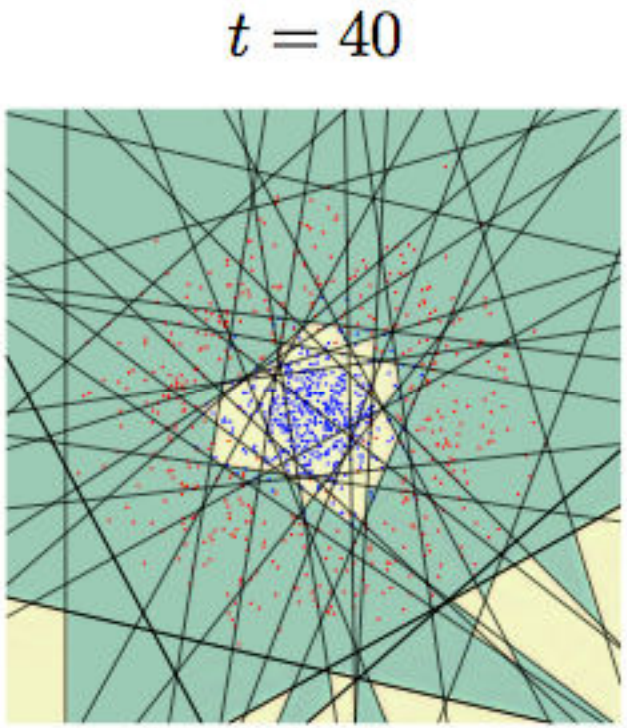
AdaBoost: In Pictures

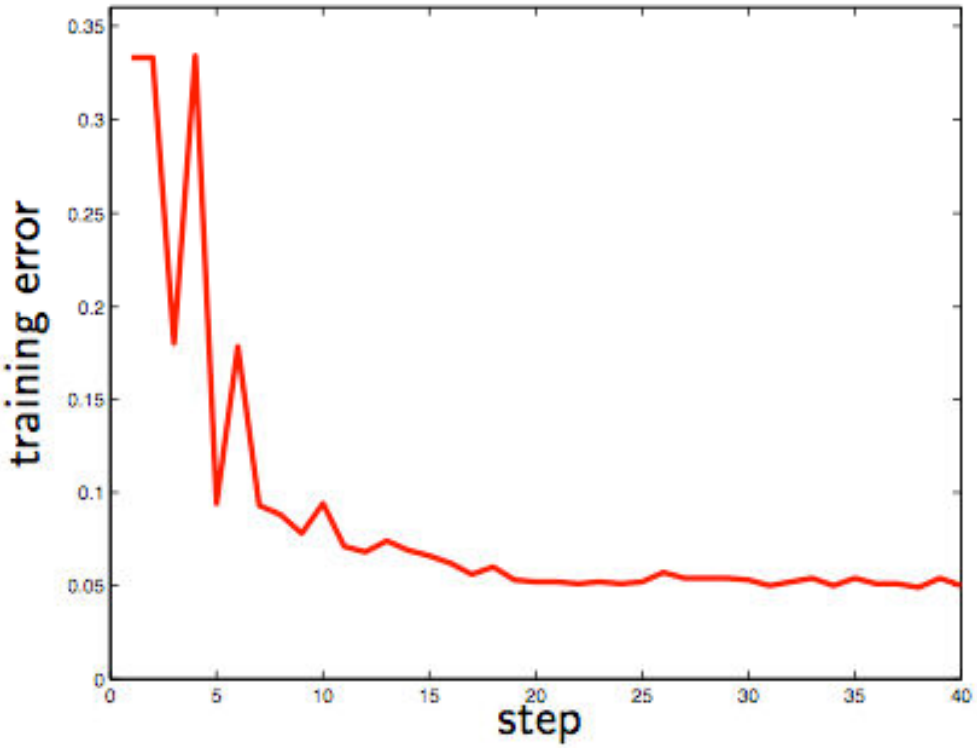
$t = 7$





AdaBoost: In Pictures





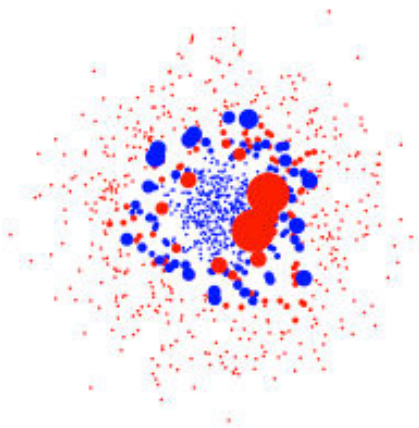
AdaBoost: In Detail

Reweighting

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$exp(-\alpha_t y_i h_t(x_i)) \begin{cases} < 1, & y_i = h_t(x_i) \\ > 1, & y_i \neq h_t(x_i) \end{cases}$$

AdaBoost: In Detail



Reweighting

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

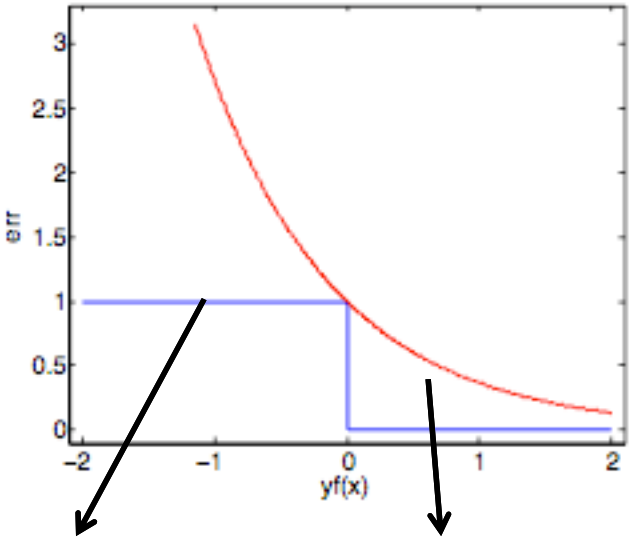
$$exp(-\alpha_t y_i h_t(x_i)) \begin{cases} < 1, & y_i = h_t(x_i) \\ > 1, & y_i \neq h_t(x_i) \end{cases}$$

- Increase (decrease) the weight of wrongly (correctly) classified examples
- An upper bound on the training error can be derived using the normalization factors
- All information about the selected hypotheses (or “features”) are captured in the distribution

AdaBoost: In Detail

The Upper Bound Theorem

$$\frac{1}{m}|\{i : H(x_i) \neq y_i\}| \leq \prod_{t=1}^{T} Z_t$$



0-1 loss **exponential loss**

Proof. *see whiteboard*

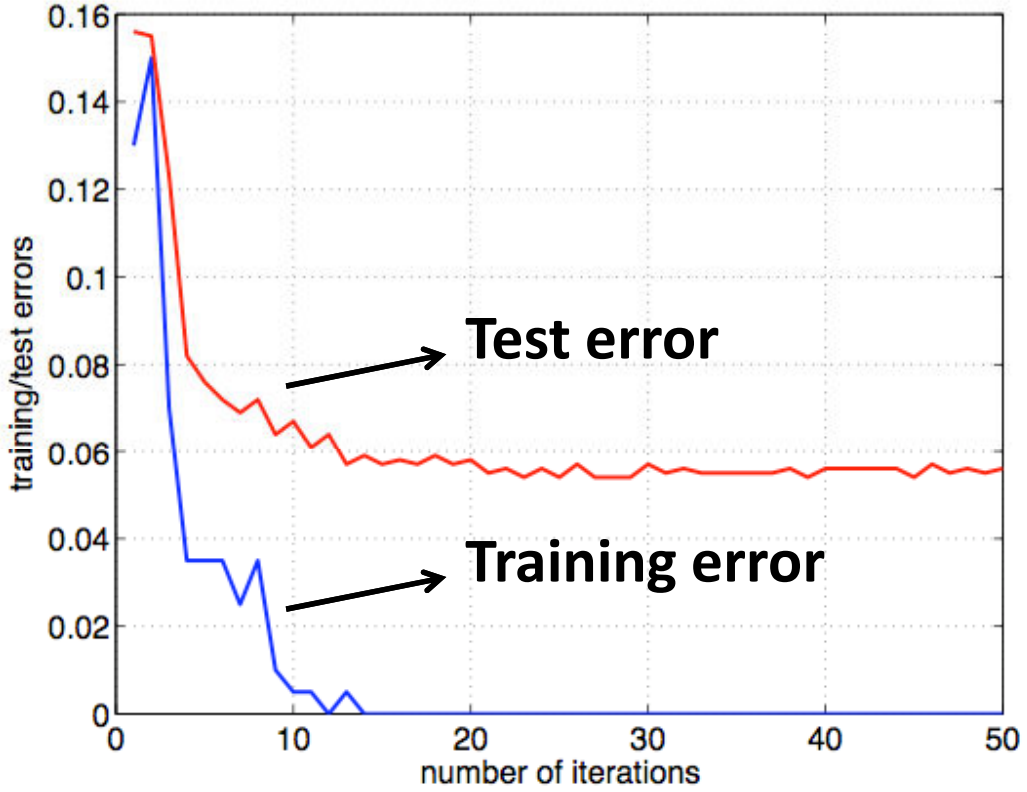
Comments/Consequences

- In AdaBoost, one minimizes an upper bound on the training error by suitable selection of the weights and hypotheses
- AdaBoost is a large-margin algorithm
- AdaBoost performs iterative loss minimization

AdaBoost: Technical Issues

- Modularity of the upper bound
- What should be the weights?
- What should be the hypotheses?
- Why and how weighted error minimization?

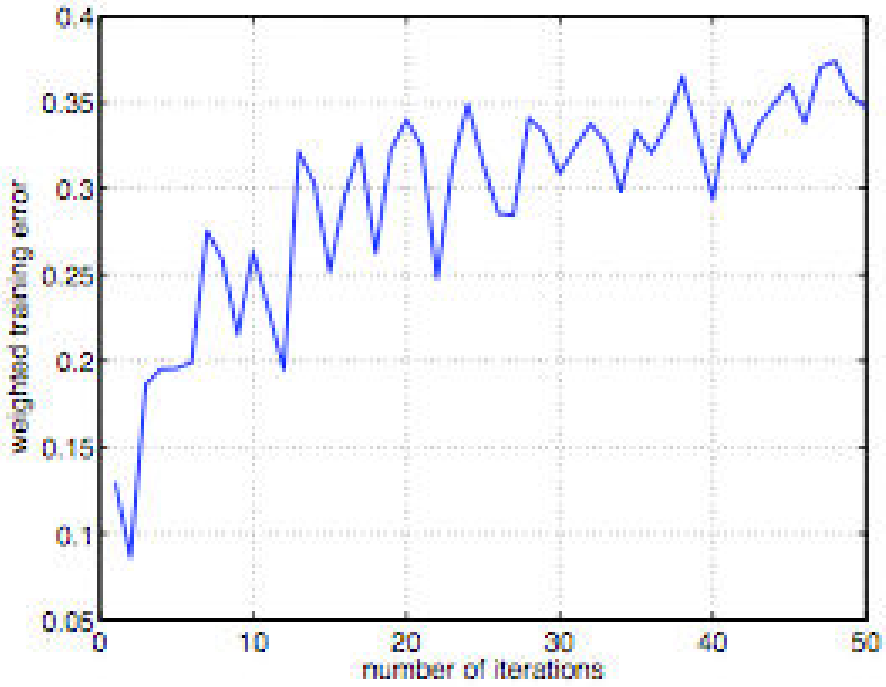
AdaBoost: Overfitting Avoidance



Isn't it a bit confusing?!!

AdaBoost: Overfitting Avoidance

- Weak Learner faces increasingly difficult classification problems through AdaBoost iterations because…



AdaBoost: Overfitting Avoidance

- In SVM, we have seen that the **margin** has something to do with **generalization**
- Very roughly:

  the larger the margin, the better the generalization

- **Better generalization** means **less overfitting**
- AdaBoost also maximizes the margin because it minimizes the exponential loss >>

AdaBoost: Overfitting Avoidance

AdaBoost Margin

In SVM

$$\bar{\gamma} = \frac{y\left(w_1 x_1 + \ldots + w_n x_n\right)}{\left\|w\right\|_2}$$

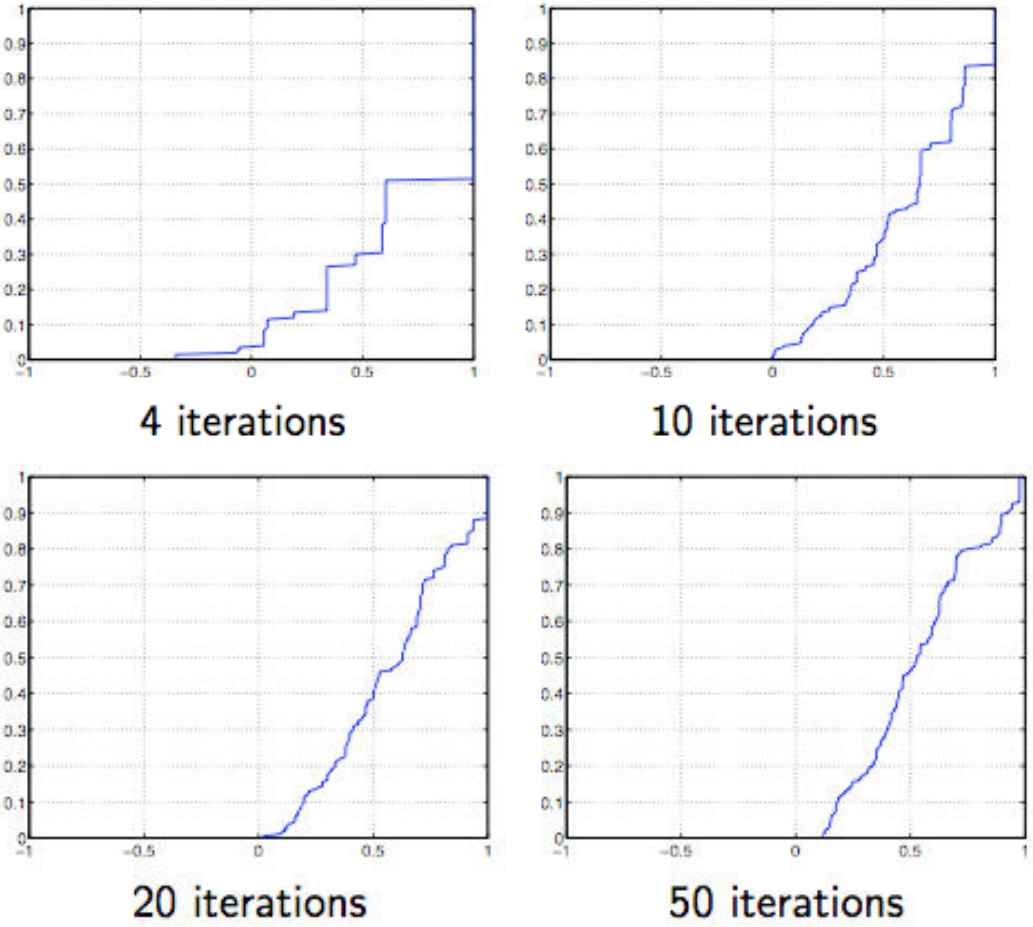
In AdaBoost

$$\bar{\gamma} = \frac{y\left(\alpha_1 h_1(x) + \ldots + \alpha_T h_T(x)\right)}{\alpha_1 + \ldots + \alpha_T}$$

Do you see the analogy?

AdaBoost: Overfitting Avoidance

Cumulative distribution of margin values



AdaBoost: Overfitting Avoidance

Intuitively, AdaBoost doesn't overfit because

1. Weighted errors increase through AdaBoost iterations
2. It's a large margin algorithm

AdaBoost: Variants

Freund & Schapire 1995

- Discrete ($h : \mathcal{X} \to \{0, 1\}$)
- Multiclass AdaBoost.M1 ($h : \mathcal{X} \to \{0, 1, ..., k\}$)
- Multiclass AdaBoost.M2 ($h : \mathcal{X} \to [0, 1]^k$)
- Real valued AdaBoost.R ($Y = [0, 1]$, $h : \mathcal{X} \to [0, 1]$)

Schapire & Singer 1999

- Confidence rated prediction ($h : \mathcal{X} \to R$, two-class)
- Multilabel AdaBoost.MR, AdaBoost.MH (different formulation of minimised loss)

Oza 2001

- Online AdaBoost

Boosting: Example Applications

Boosting Image Retrieval

Tieu and Jones. *IJCV* 56(1-2):17-36, 2004.

TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context

Shotton et al. *IJCV* 81:2-23, 2009.

Generic Object Recognition with Boosting

Opelt et al. *PAMI* 28(3):416-431, 2006.

Boosting Sex Identification Performance

Baluja and Rowley. *IJCV* 71(1):111-119, 2007.

Sharing visual features for multiclass …

Torralba et al. *PAMI* 29(5):854-869, 2007.