Introduction to Machine Learning
Lecture 11

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Boosting

Boosting Ideas

- Main idea: use weak learner to create strong learner.
- Ensemble method: combine base classifiers returned by weak learner.
- Finding simple relatively accurate base classifiers often not hard.
- But, how should base classifiers be combined?

AdaBoost

(Freund and Schapire, 1997)

$H \subseteq \{-1,+1\}^X$.

$\textsc{AdaBoost}(S=((x_1,y_1),\ldots,(x_m,y_m)))$

for $i \leftarrow 1$ **to** m **do**
    $D_1(i) \leftarrow \frac{1}{m}$
for $t \leftarrow 1$ **to** T **do**
    $h_t \leftarrow$ base classifier in H with small error $\epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$
    $\alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
    $Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}}$ $\quad \triangleright$ normalization factor
    for $i \leftarrow 1$ **to** m **do**
        $D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
$f \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
return $h = \mathrm{sgn}(f)$

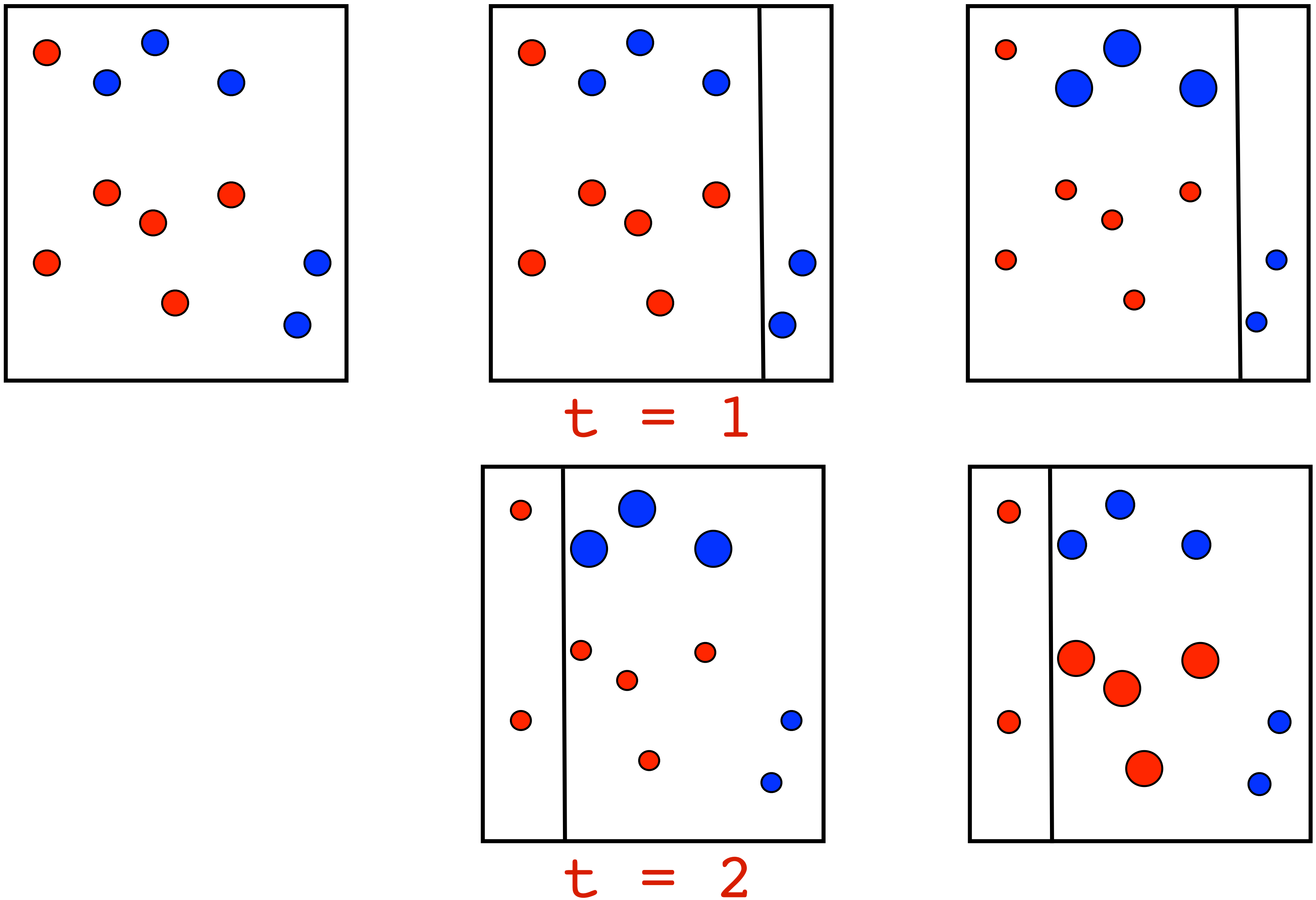
Notes

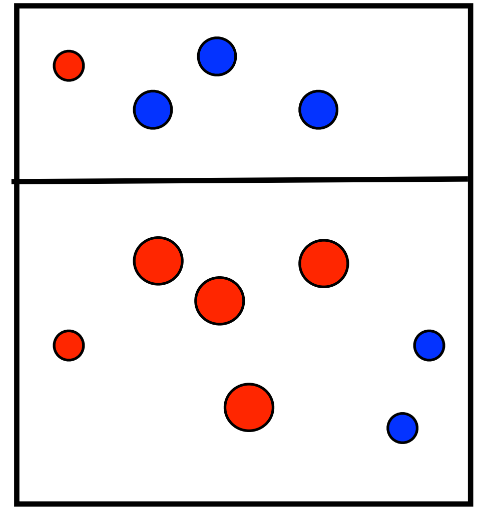
- Distributions D_t over training sample:
  - originally uniform.
  - at each round, the weight of a misclassified example is increased.
  - observation: $D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m \prod_{s=1}^t Z_s}$, since

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} = \frac{D_{t-1}(i) e^{-\alpha_{t-1} y_i h_{t-1}(x_i)} e^{-\alpha_t y_i h_t(x_i)}}{Z_{t-1} Z_t} = \frac{1}{m} \frac{e^{-y_i \sum_{s=1}^t \alpha_s h_s(x_i)}}{\prod_{s=1}^t Z_s}.$$

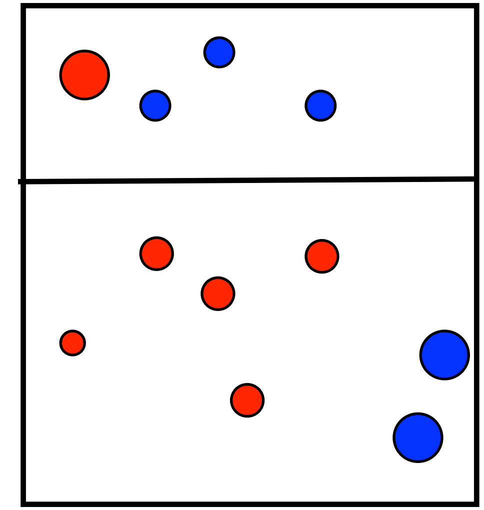
- Weight assigned to base classifier h_t: α_t directy depends on the accuracy of h_t at round t .

Illustration



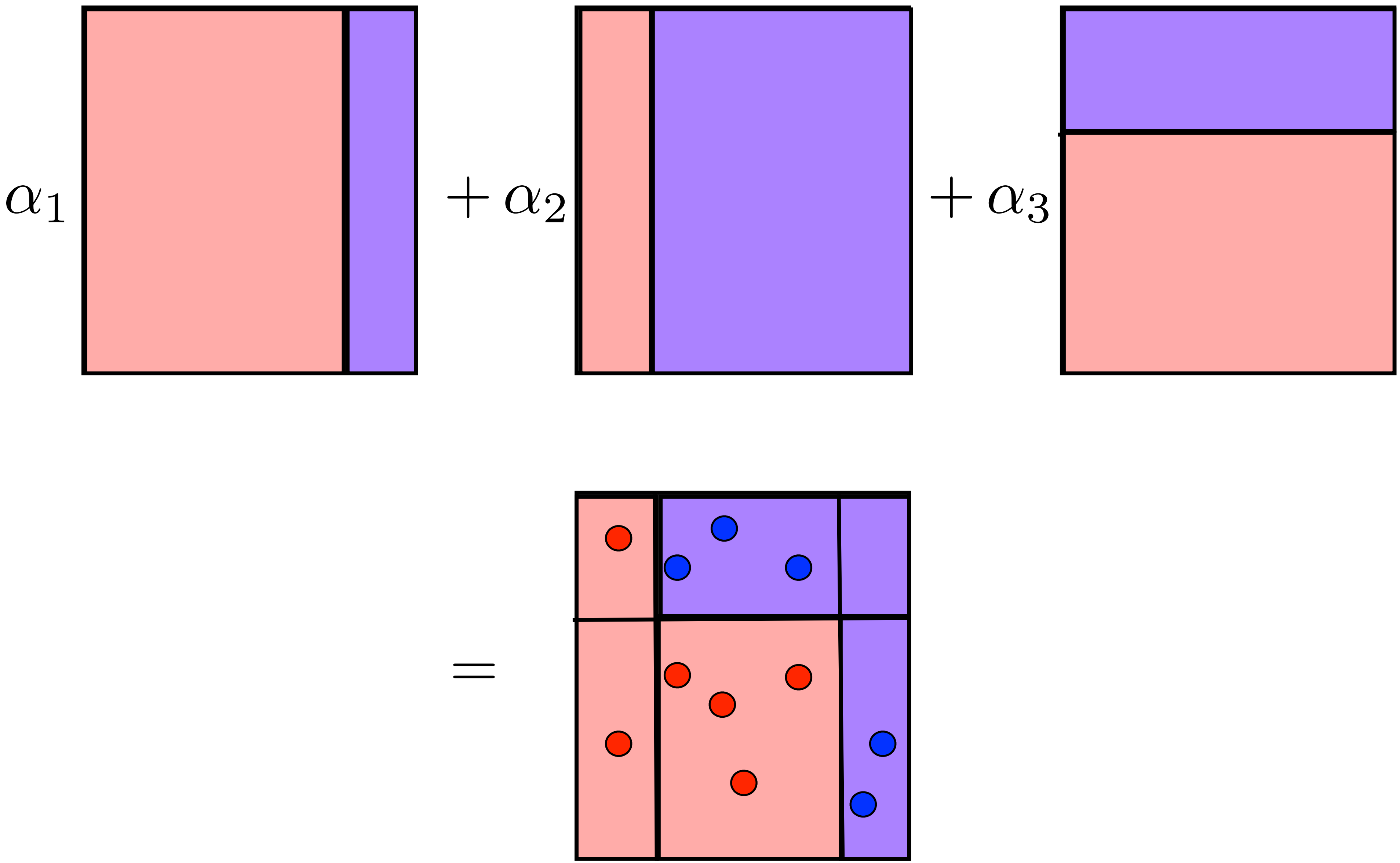


t = 3



. . .

. . .


α_1
$+ \alpha_2$
$+ \alpha_3$
$=$

Bound on Empirical Error

(Freund and Schapire, 1997)

- Theorem: The empirical error of the classifier output by AdaBoost verifies:

$$\widehat{R}(h) \leq \exp\left[-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_t\right)^2\right].$$

  - If further for all $t \in [1, T]$, $\gamma \leq \left(\frac{1}{2} - \epsilon_t\right)$, then

$$\widehat{R}(h) \leq \exp(-2\gamma^2 T).$$

  - γ does not need to be known in advance: adaptive boosting.

- Proof: Since, as we saw, $D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m \prod_{s=1}^{t} Z_s}$,

$$\widehat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{y_i f(x_i) \leq 0} \leq \frac{1}{m} \sum_{i=1}^{m} \exp(-y_i f(x_i))$$

$$\leq \frac{1}{m} \sum_{i=1}^{m} \left[m \prod_{t=1}^{T} Z_t \right] D_{T+1}(i) = \prod_{t=1}^{T} Z_t.$$

- Now, since Z_t is a normalization factor,

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h_t(x_i)} \\ &= \sum_{i : y_i h_t(x_i) \geq 0} D_t(i) e^{-\alpha_t} + \sum_{i : y_i h_t(x_i) < 0} D_t(i) e^{\alpha_t} \\ &= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \\ &= (1 - \epsilon_t) \sqrt{\tfrac{\epsilon_t}{1 - \epsilon_t}} + \epsilon_t \sqrt{\tfrac{1 - \epsilon_t}{\epsilon_t}} = 2 \sqrt{\epsilon_t (1 - \epsilon_t)}. \end{aligned}$$

- Thus,

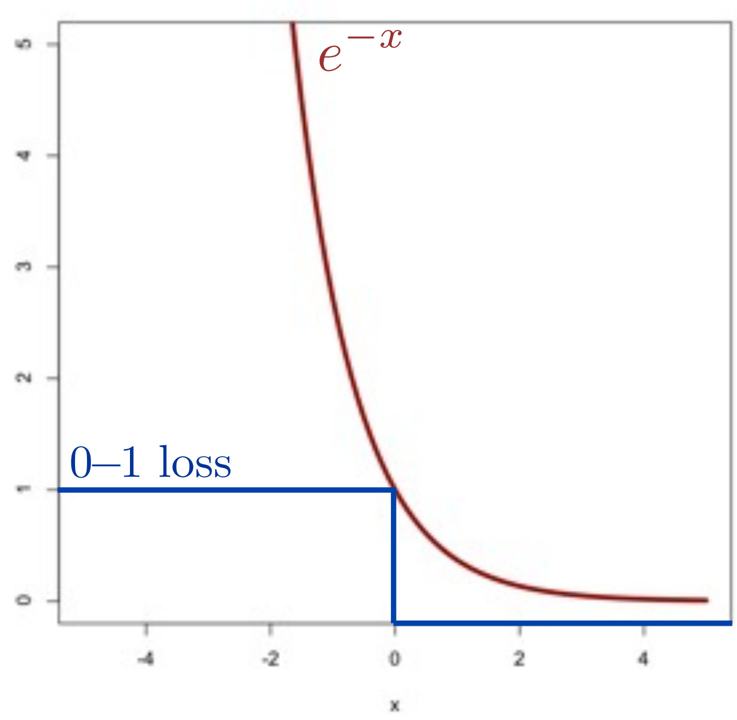
$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_{t=1}^{T} \sqrt{1 - 4\Big(\tfrac{1}{2} - \epsilon_t\Big)^2}$$
$$\leq \prod_{t=1}^{T} \exp\Big[-2\Big(\tfrac{1}{2} - \epsilon_t\Big)^2\Big] = \exp\Big[-2\sum_{t=1}^{T}\Big(\tfrac{1}{2} - \epsilon_t\Big)^2\Big].$$

- Notes:
  - α_t minimizer of $\alpha \mapsto (1-\epsilon_t)e^{-\alpha} + \epsilon_t e^{\alpha}$.
  - since $(1-\epsilon_t)e^{-\alpha_t} = \epsilon_t e^{\alpha_t}$, at each round, AdaBoost assigns the same probability mass to correctly classified and misclassified instances.
  - for base classifiers $x \mapsto [-1, +1]$, α_t can be similarly chosen to minimize Z_t.

AdaBoost = Coordinate Descent

- Objective Function: convex and differentiable.

$$F(\boldsymbol{\alpha}) = \sum_{i=1}^{m} e^{-y_i f(x_i)} = \sum_{i=1}^{m} e^{-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}.$$



- Direction: unit vector $\mathbf{e}_t$ with

$$\mathbf{e}_t = \operatorname*{argmin}_t \left.\frac{dF(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_t)}{d\eta}\right|_{\eta=0}.$$

- Since $F(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_t) = \sum_{i=1}^{m} e^{-y_i \sum_{s=1}^{t-1} \alpha_s h_s(x_i)} e^{-y_i \eta h_t(x_i)}$,

$$\begin{aligned}
\left.\frac{dF(\alpha_{t-1} + \eta \mathbf{e}_t)}{d\eta}\right|_{\eta=0} &= -\sum_{i=1}^{m} y_i h_t(x_i) \exp\Big[-y_i \sum_{s=1}^{t-1} \alpha_s h_s(x_i)\Big] \\
&= -\sum_{i=1}^{m} y_i h_t(x_i) D_t(i) \Big[m \prod_{s=1}^{t-1} Z_s\Big] \\
&= -[(1-\epsilon_t) - \epsilon_t] \Big[m \prod_{s=1}^{t-1} Z_s\Big] = \boxed{[2\epsilon_t - 1]} \Big[m \prod_{s=1}^{t-1} Z_s\Big].
\end{aligned}$$

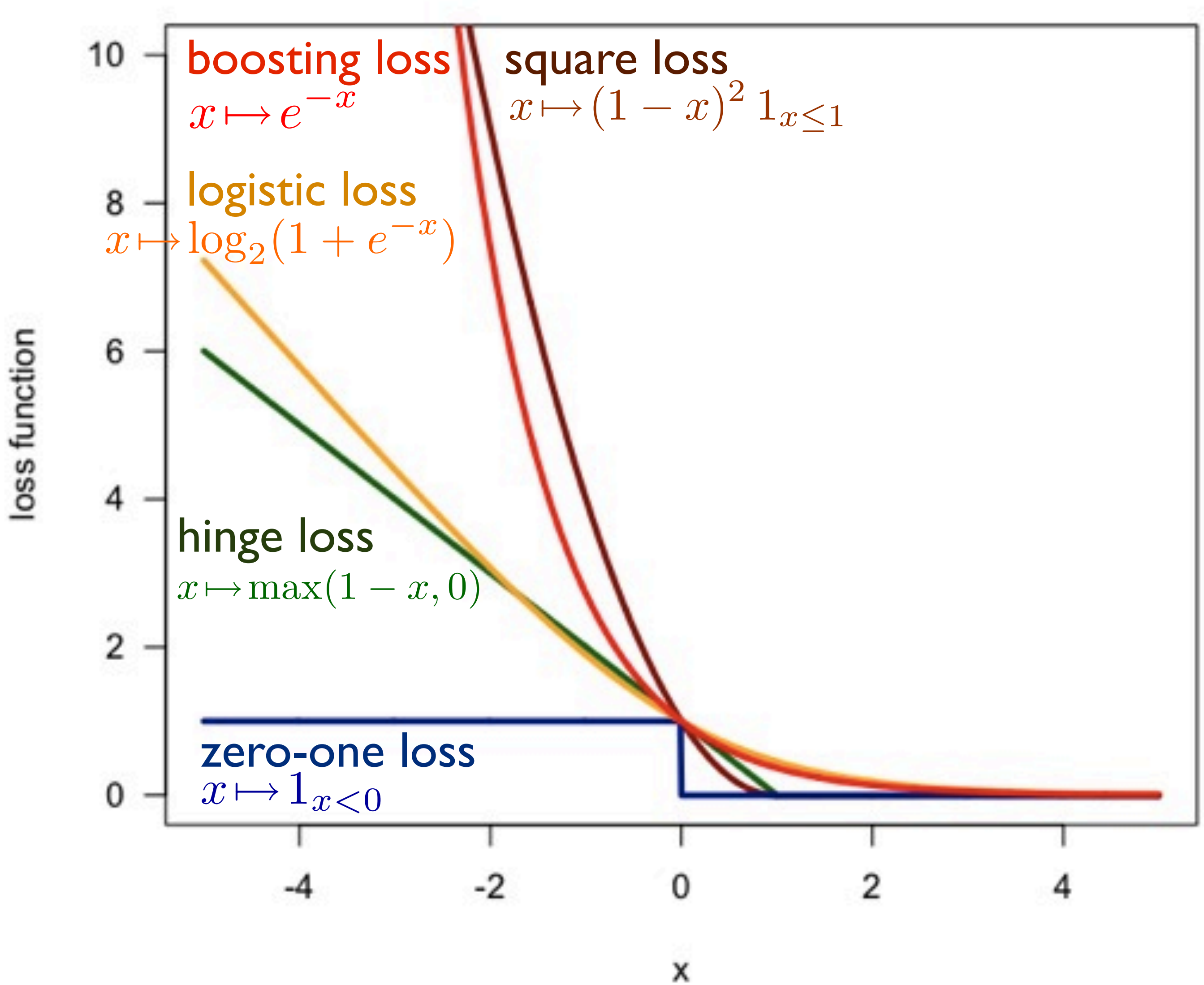
Thus, direction corresponding to base classifier with smallest error.

- Step size: obtained via

$$\begin{aligned}
\frac{dF(\boldsymbol{\alpha}_{t-1}+\eta\mathbf{e}_t)}{d\eta}=0 &\Leftrightarrow -\sum_{i=1}^{m} y_i h_t(x_i)\exp\Big[-y_i\sum_{s=1}^{t-1}\alpha_s h_s(x_i)\Big]e^{-y_i h_t(x_i)\eta}=0\\
&\Leftrightarrow -\sum_{i=1}^{m} y_i h_t(x_i) D_t(i)\Big[m\prod_{s=1}^{t-1}Z_s\Big]e^{-y_i h_t(x_i)\eta}=0\\
&\Leftrightarrow -\sum_{i=1}^{m} y_i h_t(x_i) D_t(i) e^{-y_i h_t(x_i)\eta}=0\\
&\Leftrightarrow -[(1-\epsilon_t)e^{-\eta}-\epsilon_t e^{\eta}]=0\\
&\Leftrightarrow \eta=\boxed{\frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}}.
\end{aligned}$$

Thus, step size matches base classifier weight of AdaBoost.

Alternative Loss Functions

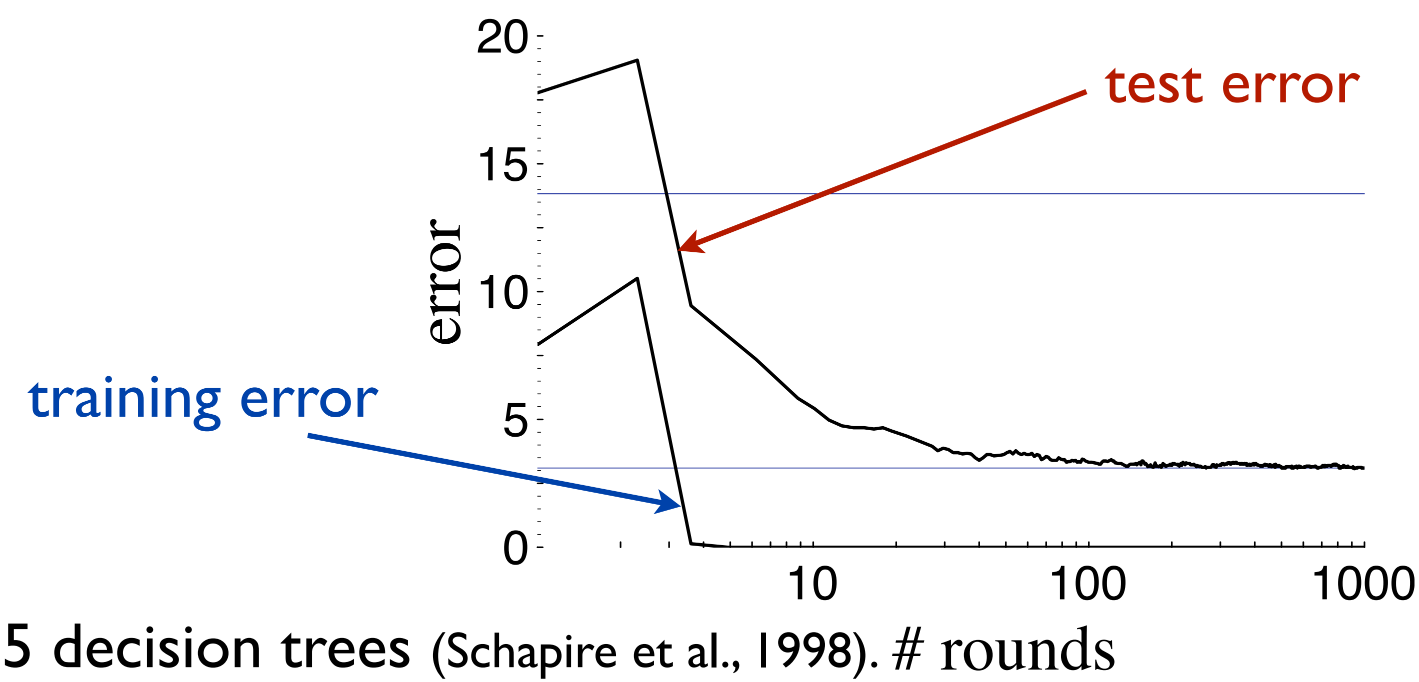


Standard Use in Practice

- Base learners: decision trees, quite often just decision stumps (trees of depth one).
- Boosting stumps:
  - data in $\mathbb{R}^N$, e.g., $N=2$, $(\text{height}(x), \text{weight}(x))$.
  - associate a stump to each component.
  - pre-sort each component: $O(Nm \log m)$.
  - at each round, find best component and threshold.
  - total complexity: $O((m \log m)N + mNT)$.
  - stumps not weak learners: think XOR example!

Overfitting?

- We could expect that AdaBoost would overfit for large values of T, and that is in fact observed in some cases, but in various others it is not!
- Several empirical observations (not all): AdaBoost does not seem to overfit, furthermore:



C4.5 decision trees (Schapire et al., 1998).

L1 Margin Definitions

- Definition: the margin of a point x with label y is (the $\|\cdot\|_\infty$ algebraic distance of $\mathbf{x} = [h_1(x), \ldots, h_T(x)]^\top$ to the hyperplane $\boldsymbol{\alpha} \cdot \mathbf{x} = 0$):

$$\rho(x) = \frac{yf(x)}{\sum_{t=T}^{m} \alpha_t} = \frac{y \sum_{t=1}^{T} \alpha_t h_t(x)}{\|\boldsymbol{\alpha}\|_1} = y \frac{\boldsymbol{\alpha} \cdot \mathbf{x}}{\|\boldsymbol{\alpha}\|_1}.$$

- Definition: the margin of the classifier for a sample $S = (x_1, \ldots, x_m)$ is the minimum margin of the points in that sample:

$$\rho = \min_{i \in [1,m]} y_i \frac{\boldsymbol{\alpha} \cdot \mathbf{x}_i}{\|\boldsymbol{\alpha}\|_1}.$$

- **Note:**
  - SVM margin: $$\rho = \min_{i \in [1,m]} y_i \frac{\mathbf{w} \cdot \Phi(x_i)}{\|\mathbf{w}\|_2}.$$
  - Boosting margin: $$\rho = \min_{i \in [1,m]} y_i \frac{\boldsymbol{\alpha} \cdot \mathbf{H}(x_i)}{\|\boldsymbol{\alpha}\|_1},$$
    with $\mathbf{H}(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_T(x) \end{bmatrix}$.
- **Distances:** $\|\cdot\|_q$ distance to hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$:
  $$\frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|_p},$$
  with $\frac{1}{p} + \frac{1}{q} = 1$.

Convex Hull of Hypothesis Set

- Definition: Let H be a set of functions mapping from X to $\mathbb{R}$. The convex hull of H is defined by

$$\operatorname{conv}(H) = \{\sum_{k=1}^{p} \mu_k h_k : p \geq 1, \mu_k \geq 0, \sum_{k=1}^{p} \mu_k \leq 1, h_k \in H\}.$$

  - ensemble methods are often based on such convex combinations of hypotheses.

Margin Bound - Ensemble Methods

(Koltchinskii and Panchenko, 2002)

- **Theorem**: Let H be a set of real-valued functions. Fix $\rho > 0$. For any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $h \in \text{conv}(H)$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\mathfrak{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

where $\mathfrak{R}_m(H)$ is a measure of the complexity of H.

Notes

- For AdaBoost, the bound applies to the functions

$$x \mapsto \frac{f(x)}{\|\boldsymbol{\alpha}\|_1} = \frac{\sum_{t=1}^{T} \alpha_t h_t(x)}{\|\boldsymbol{\alpha}\|_1} \in \operatorname{conv}(H).$$

- Note that T does not appear in the bound.

But, Does AdaBoost Maximize the Margin?

- No: AdaBoost may converge to a margin that is significantly below the maximum margin (Rudin et al., 2004) (e.g., 1/3 instead of 3/8)!
- Lower bound: AdaBoost can achieve asymptotically a margin that is at least $\frac{\rho_{\max}}{2}$ if data separable and some conditions on the base learners (Rätsch and Warmuth, 2002).
- Several boosting-type margin-maximization algorithms: but, performance in practice not clear or not reported.

Outliers

- AdaBoost assigns larger weights to harder examples.
- Application:
  - Detecting mislabeled examples.
  - Dealing with noisy data: regularization based on the average weight assigned to a point (soft margin idea for boosting) (Meir and Rätsch, 2003).

Advantages of AdaBoost

- Simple: straightforward implementation.
- Efficient: complexity $O(mNT)$ for stumps:
  - when N and T are not too large, the algorithm is quite fast.
- Theoretical guarantees: but still many questions.
  - AdaBoost not designed to maximize margin.
  - regularized versions of AdaBoost.

Weaker Aspects

- Parameters:
  - need to determine T, the number of rounds of boosting: stopping criterion.
  - need to determine base learners: risk of overfitting or low margins.
- Noise: severely damages the accuracy of Adaboost (Dietterich, 2000).
  - boosting algorithms based on convex potentials do not tolerate even low levels of random noise, even with L1 regularization or early stopping (Long and Servedio, 2010).

References

- Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2): 139-158, 2000.

- Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139, 1997.

- G. Lebanon and J. Lafferty. Boosting and maximum likelihood for exponential models. In NIPS, pages 447–454, 2001.

- Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. In *Advanced Lectures on Machine Learning (LNAI2600)*, 2003.

- J. von Neumann. Zur Theorie der Gesellschaftsspiele. Mathematische Annalen, 100:295-320, 1928.

- Cynthia Rudin, Ingrid Daubechies and Robert E. Schapire. The dynamics of AdaBoost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 5: 1557-1595, 2004.

References

- Rätsch, G., and Warmuth, M. K. (2002) "Maximizing the Margin with Boosting", in *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 02)*, Sidney, Australia, pp. 334–350, July 2002.

- Robert E. Schapire. The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.

- Robert E. Schapire, Yoav Freund, Peter Bartlett and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5): 1651-1686, 1998.