Foundations of Machine Learning
Boosting

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

Weak Learning

(Kearns and Valiant, 1994)

- Definition: concept class C is weakly PAC-learnable if there exists a (weak) learning algorithm L and $\gamma > 0$ such that:
  - for all $\delta > 0$, for all $c \in C$ and all distributions D,

$$\Pr_{S \sim D}\left[R(h_S) \leq \frac{1}{2} - \gamma\right] \geq 1 - \delta,$$

  - for samples S of size $m = poly(1/\delta)$ for a fixed polynomial.

Boosting Ideas

- Finding simple relatively accurate base classifiers often not hard ⟵ weak learner.
- Main ideas:
  - use weak learner to create a strong learner.
  - combine base classifiers returned by weak learner (ensemble method).
- But, how should the 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 $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
  $\alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
  $Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}}$ ▷ normalization factor
  for $i \leftarrow 1$ **to** m **do**
    $D_{t+1}(i) \leftarrow \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
  $f_t \leftarrow \sum_{s=1}^{t} \alpha_s h_s$
return $h = \operatorname{sgn}(f_T)$

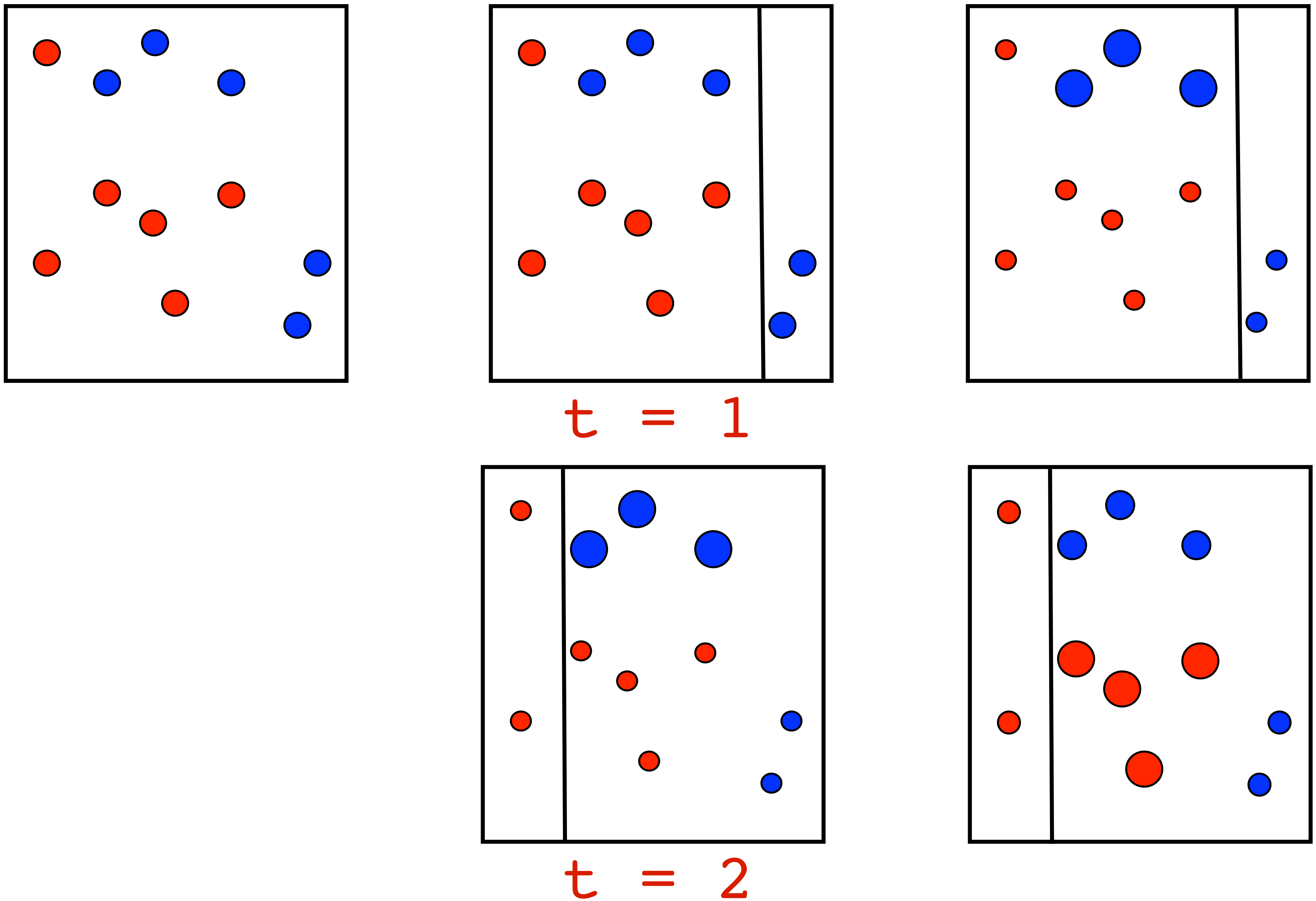
Notes

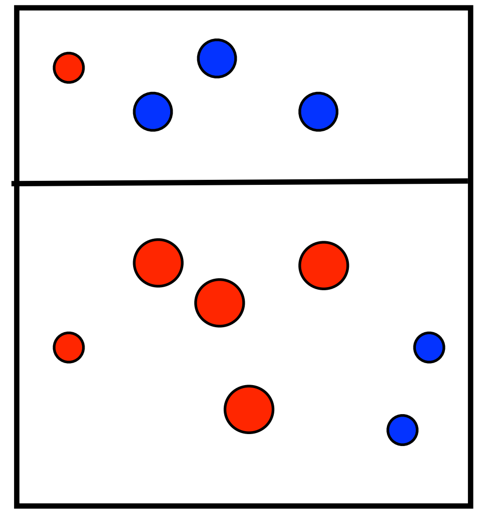
- Distributions D_t over training sample:
  - originally uniform.
  - at each round, the weight of a misclassified example is increased.
  - observation: $D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m \prod_{s=1}^t Z_s}$, since

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

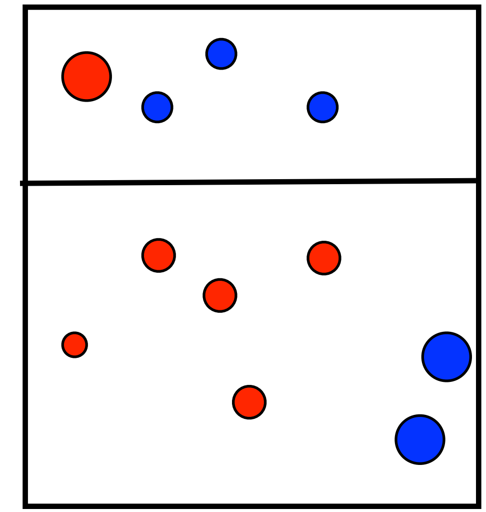
- Weight assigned to base classifier h_t : α_t directly depends on the accuracy of h_t at round t .

Illustration



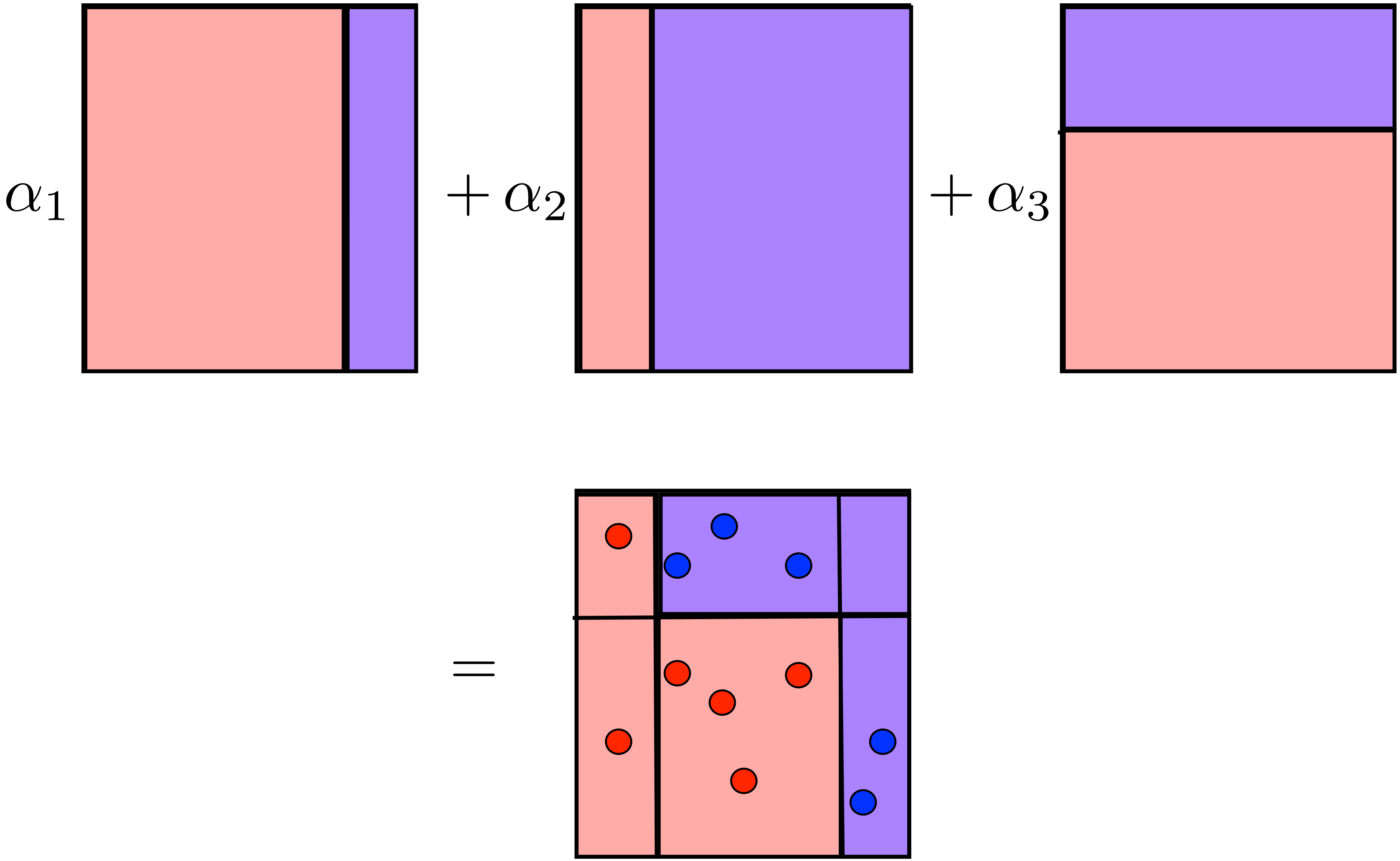


t = 3



. . .

. . .


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,

$$Z_t = \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h_t(x_i)}$$

$$= \sum_{i: y_i h_t(x_i) \geq 0} D_t(i) e^{-\alpha_t} + \sum_{i: y_i h_t(x_i) < 0} D_t(i) e^{\alpha_t}$$

$$= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t}$$

$$= (1 - \epsilon_t) \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = 2\sqrt{\epsilon_t (1 - \epsilon_t)}.$$

- Thus,

$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_{t=1}^{T} \sqrt{1 - 4\Big(\frac{1}{2} - \epsilon_t\Big)^2}$$

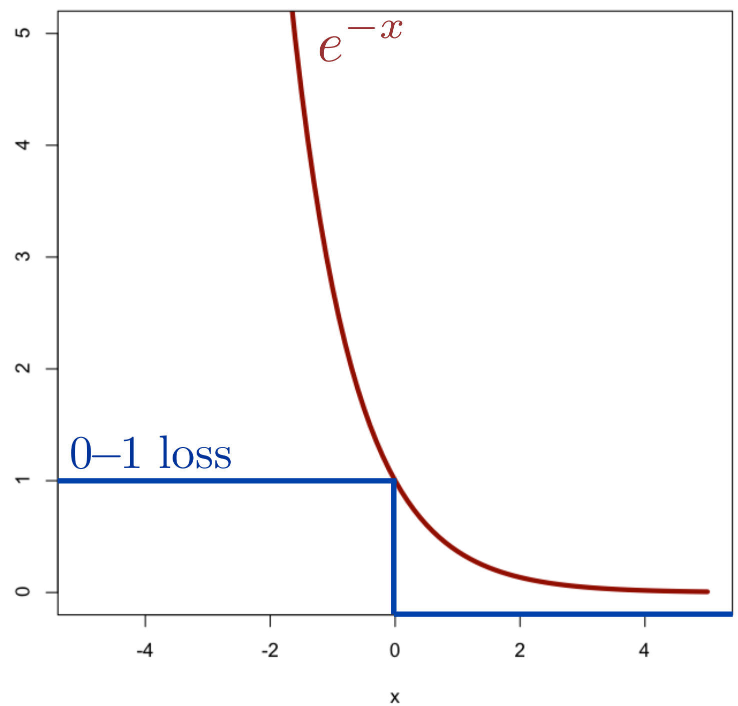
$$\leq \prod_{t=1}^{T} \exp\Big[-2\Big(\frac{1}{2} - \epsilon_t\Big)^2\Big] = \exp\Big[-2\sum_{t=1}^{T}\Big(\frac{1}{2} - \epsilon_t\Big)^2\Big].$$

- 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.

$$F(\bar{\boldsymbol{\alpha}}) = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f(x_i)} = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_j h_j(x_i)}.$$



- Direction: unit vector $\mathbf{e}_k$ with best directional derivative:

$$F'(\bar{\boldsymbol{\alpha}}_{t-1}, \mathbf{e}_k) = \lim_{\eta \to 0} \frac{F(\bar{\boldsymbol{\alpha}}_{t-1} + \eta \mathbf{e}_k) - F(\bar{\boldsymbol{\alpha}}_{t-1})}{\eta}.$$

- Since $F(\bar{\boldsymbol{\alpha}}_{t-1} + \eta \mathbf{e}_k) = \frac{1}{m} \sum_{i=1}^{m} e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i) - \eta y_i h_k(x_i)}$,

$$\begin{aligned} F'(\bar{\boldsymbol{\alpha}}_{t-1}, \mathbf{e}_k) &= -\frac{1}{m} \sum_{i=1}^{m} y_i h_k(x_i) e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i)} \\ &= -\frac{1}{m} \sum_{i=1}^{m} y_i h_k(x_i) \bar{D}_t(i) \bar{Z}_t \\ &= -\left[\sum_{i=1}^{m} \bar{D}_t(i) 1_{y_i h_k(x_i)=+1} - \sum_{i=1}^{m} \bar{D}_t(i) 1_{y_i h_k(x_i)=-1} \right] \frac{\bar{Z}_t}{m} \\ &= -\left[(1 - \bar{\epsilon}_{t,k}) - \bar{\epsilon}_{t,k} \right] \frac{\bar{Z}_t}{m} = \left[2\bar{\epsilon}_{t,k} - 1 \right] \frac{\bar{Z}_t}{m}. \end{aligned}$$

Thus, direction corresponding to base classifier with smallest error.

- **Step size:** η chosen to minimize $F(\bar{\boldsymbol{\alpha}}_{t-1} + \eta\, \mathbf{e}_k)$;

$$\frac{dF(\bar{\boldsymbol{\alpha}}_{t-1} + \eta \mathbf{e}_k)}{d\eta} = 0 \Leftrightarrow -\sum_{i=1}^{m} y_i h_k(x_i) e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i)} e^{-\eta y_i h_k(x_i)} = 0$$

$$\Leftrightarrow -\sum_{i=1}^{m} y_i h_k(x_i) \bar{D}_t(i) \bar{Z}_t e^{-\eta y_i h_k(x_i)} = 0$$

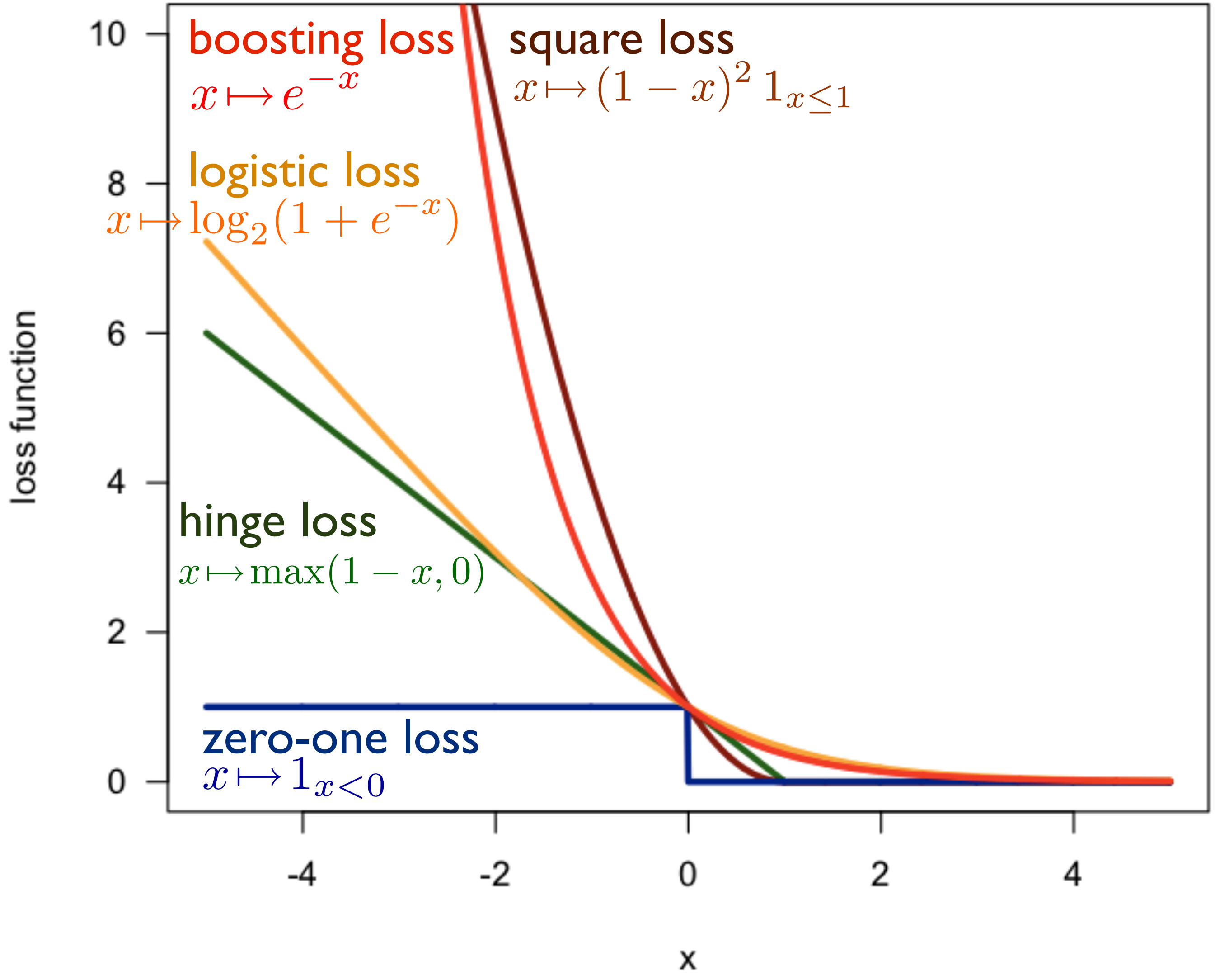
$$\Leftrightarrow -\sum_{i=1}^{m} y_i h_k(x_i) \bar{D}_t(i) e^{-\eta y_i h_k(x_i)} = 0$$

$$\Leftrightarrow -\Big[(1 - \bar{\epsilon}_{t,k}) e^{-\eta} - \bar{\epsilon}_{t,k} e^{\eta}\Big] = 0$$

$$\Leftrightarrow \boxed{\eta = \frac{1}{2} \log \frac{1 - \bar{\epsilon}_{t,k}}{\bar{\epsilon}_{t,k}}.}$$

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?

- Assume that $\text{VCdim}(H) = d$ and for a fixed T, define

$$\mathcal{F}_T = \left\{ \text{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t - b \right) : \alpha_t, b \in \mathbb{R}, h_t \in H \right\}.$$

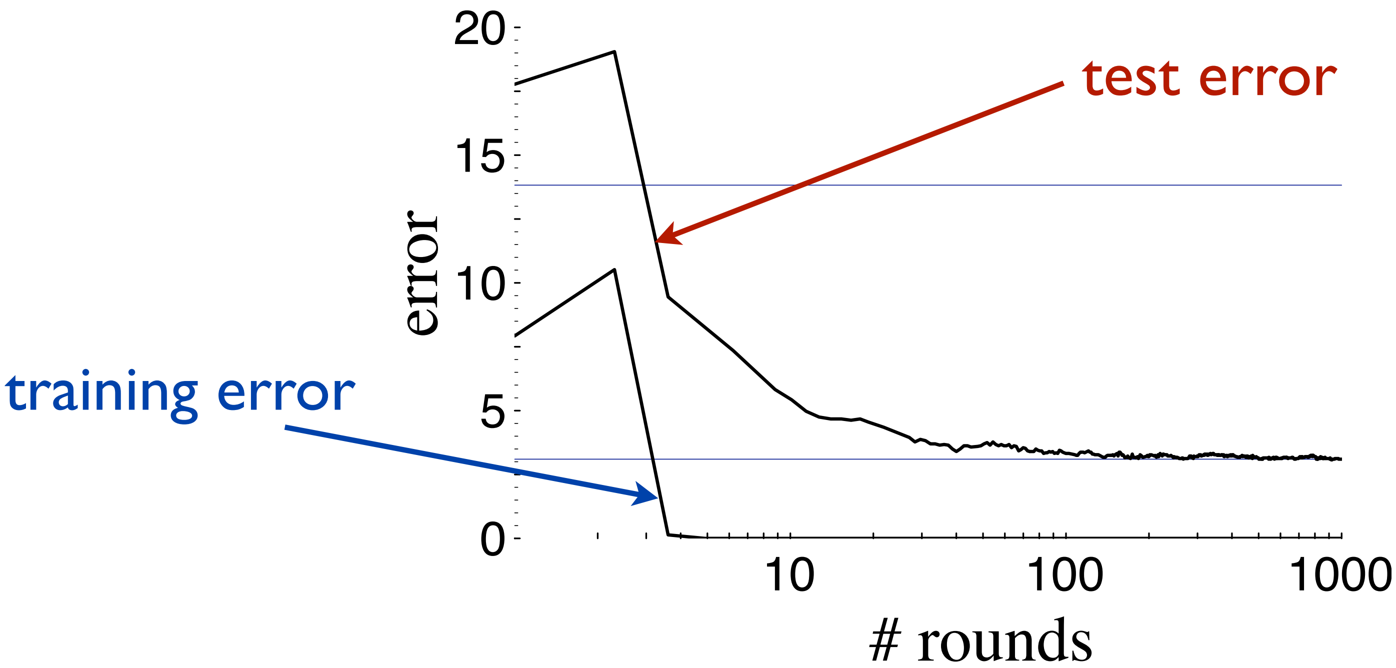
- $\mathcal{F}_T$ can form a very rich family of classifiers. It can be shown (Freund and Schapire, 1997) that:

$$\text{VCdim}(\mathcal{F}_T) \leq 2(d+1)(T+1)\log_2((T+1)e).$$

- This suggests that AdaBoost could overfit for large values of T, and that is in fact observed in some cases, but in various others it is not!

Empirical Observations

- Several empirical observations (not all): AdaBoost does not seem to overfit, furthermore:



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

Rademacher Complexity of Convex Hulls

- **Theorem**: Let H be a set of functions mapping from X to $\mathbb{R}$. Let the convex hull of H be defined as

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

Then, for any sample S, $\widehat{\mathfrak{R}}_S(\operatorname{conv}(H)) = \widehat{\mathfrak{R}}_S(H)$.

- **Proof**: $$\begin{aligned}\widehat{\mathfrak{R}}_S(\operatorname{conv}(H)) &= \frac{1}{m} \operatorname*{E}_{\sigma}\Big[\sup_{h_k \in H, \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 \leq 1} \sum_{i=1}^{m} \sigma_i \sum_{k=1}^{p} \mu_k h_k(x_i)\Big] \\ &= \frac{1}{m} \operatorname*{E}_{\sigma}\Big[\sup_{h_k \in H} \sup_{\boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 \leq 1} \sum_{k=1}^{p} \mu_k \Big(\sum_{i=1}^{m} \sigma_i h_k(x_i)\Big)\Big] \\ &= \frac{1}{m} \operatorname*{E}_{\sigma}\Big[\sup_{h_k \in H} \max_{k \in [1,p]} \Big(\sum_{i=1}^{m} \sigma_i h_k(x_i)\Big)\Big] \\ &= \frac{1}{m} \operatorname*{E}_{\sigma}\Big[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i)\Big] = \widehat{\mathfrak{R}}_S(H).\end{aligned}$$

Margin Bound - Ensemble Methods

(Koltchinskii and Panchenko, 2002)

- Corollary: 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 $h \in \mathrm{conv}(H)$:

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

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\widehat{\mathfrak{R}}_S(H) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

- Proof: Direct consequence of margin bound of Lecture 4 and $\widehat{\mathfrak{R}}_S(\mathrm{conv}(H)) = \widehat{\mathfrak{R}}_S(H)$.

Margin Bound - Ensemble Methods

(Koltchinskii and Panchenko, 2002); see also (Schapire et al., 1998)

- Corollary: Let H be a family of functions taking values in $\{-1, +1\}$ with VC dimension d. Fix $\rho > 0$. For any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $h \in \text{conv}(H)$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- Proof: Follows directly previous corollary and VC dimension bound on Rademacher complexity (see lecture 3).

Notes

- All of these bounds can be generalized to hold uniformly for all $\rho \in (0, 1)$, at the cost of an additional term $\sqrt{\frac{\log \log_2 \frac{2}{\rho}}{m}}$ and other minor constant factor changes (Koltchinskii and Panchenko, 2002).
- 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.

Margin Distribution

- **Theorem**: For any $\rho > 0$, the following holds:

$$\widehat{\Pr}\left[\frac{yf(x)}{\|\boldsymbol{\alpha}\|_1} \leq \rho\right] \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}}.$$

- **Proof**: Using the identity $D_{t+1}(i) = \frac{e^{-y_i f(x_i)}}{m \prod_{t=1}^{T} Z_t}$,

$$\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m} 1_{y_i f(x_i) - \|\boldsymbol{\alpha}\|_1 \rho \leq 0} &\leq \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i f(x_i) + \|\boldsymbol{\alpha}\|_1 \rho) \\
&= \frac{1}{m}\sum_{i=1}^{m} e^{\|\boldsymbol{\alpha}\|_1 \rho}\left[m \prod_{t=1}^{T} Z_t\right] D_{T+1}(i) \\
&= e^{\|\boldsymbol{\alpha}\|_1 \rho} \prod_{t=1}^{T} Z_t = 2^T \prod_{t=1}^{T} \left[\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\right]^{\rho} \sqrt{\epsilon_t(1-\epsilon_t)}.
\end{aligned}$$

Notes

- If for all $t \in [1, T]$, $\gamma \leq (\frac{1}{2} - \epsilon_t)$, then the upper bound can be bounded by

$$\widehat{\Pr}\left[\frac{yf(x)}{\|\boldsymbol{\alpha}\|_1} \leq \rho\right] \leq \left[(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho}\right]^{T/2}.$$

  For $\rho < \gamma$, $(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho} < 1$ and the bound decreases exponentially in T.

- For the bound to be convergent: $\rho \gg O(1/\sqrt{m})$, thus $\gamma \gg O(1/\sqrt{m})$ is roughly the condition on the edge value.

L1-Geometric Margin

- Definition: the L_1-margin $\rho_f(x)$ of a linear function $f = \sum_{t=1}^T \alpha_t h_t$ with $\boldsymbol{\alpha} \neq 0$ at a point $x \in \mathcal{X}$ is defined by

$$\rho_f(x) = \frac{|f(x)|}{\|\boldsymbol{\alpha}\|_1} = \frac{|\sum_{t=1}^T \alpha_t h_t(x)|}{\|\boldsymbol{\alpha}\|_1} = \frac{\big|\boldsymbol{\alpha} \cdot \mathbf{h}(x)\big|}{\|\boldsymbol{\alpha}\|_1}.$$

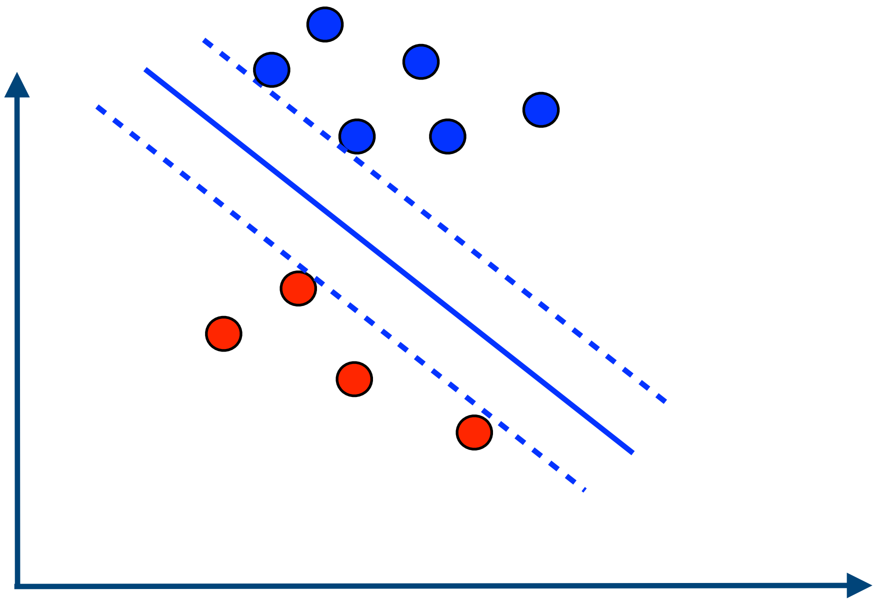
  - the L_1-margin of f over a sample $S = (x_1, \ldots, x_m)$ is its minimum margin at points in that sample:

$$\rho_f = \min_{i \in [1,m]} \rho_f(x_i) = \min_{i \in [1,m]} \frac{\big|\boldsymbol{\alpha} \cdot \mathbf{h}(x_i)\big|}{\|\boldsymbol{\alpha}\|_1}.$$

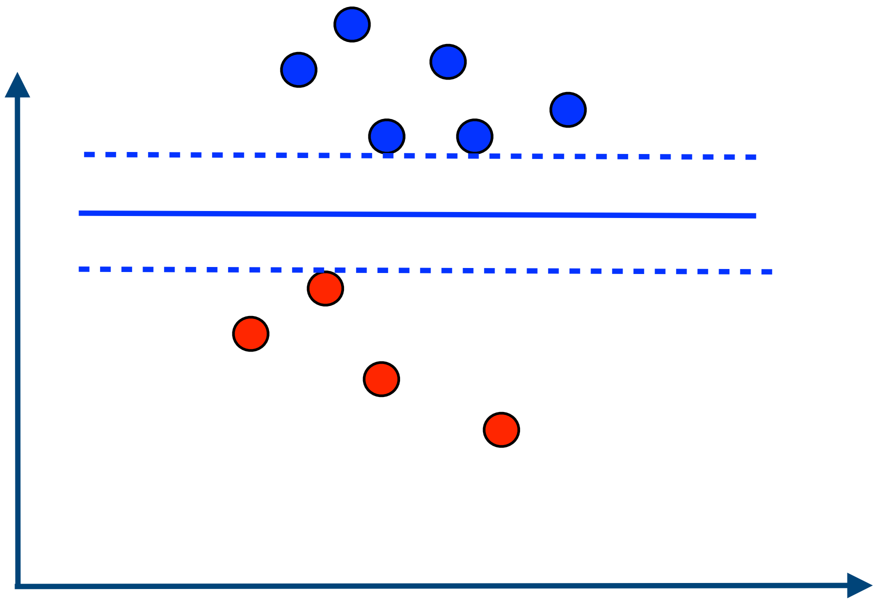
SVM vs AdaBoost

	SVM	AdaBoost
features or base hypotheses	$\mathbf{\Phi}(x) = \begin{bmatrix} \Phi_1(x) \\ \vdots \\ \Phi_N(x) \end{bmatrix}$	$\mathbf{h}(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_N(x) \end{bmatrix}$
predictor	$x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x)$	$x \mapsto \boldsymbol{\alpha} \cdot \mathbf{h}(x)$
geom. margin	$\frac{\lvert \mathbf{w} \cdot \mathbf{\Phi}(x) \rvert}{\lVert \mathbf{w} \rVert_2} = d_2(\mathbf{\Phi}(x), \text{hyperpl.})$	$\frac{\lvert \boldsymbol{\alpha} \cdot \mathbf{h}(x) \rvert}{\lVert \boldsymbol{\alpha} \rVert_1} = d_\infty(\mathbf{h}(x), \text{hyperpl.})$
conf. margin	$y(\mathbf{w} \cdot \mathbf{\Phi}(x))$	$y(\boldsymbol{\alpha} \cdot \mathbf{h}(x))$
regularization	$\lVert \mathbf{w} \rVert_2$	$\lVert \boldsymbol{\alpha} \rVert_1$ (L1-AB)

Maximum-Margin Solutions



Norm $\|\cdot\|_2$.



Norm $\|\cdot\|_\infty$.

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 the data is separable and some conditions on the base learners hold (Rätsch and Warmuth, 2002).
- Several boosting-type margin-maximization algorithms: but, performance in practice not clear or not reported.

AdaBoost's Weak Learning Condition

- **Definition**: the edge of a base classifier h_t for a distribution D over the training sample is

$$\gamma(t) = \frac{1}{2} - \epsilon_t = \frac{1}{2} \sum_{i=1}^{m} y_i h_t(x_i) D(i).$$

- **Condition**: there exists $\gamma > 0$ for any distribution D over the training sample and any base classifier

$$\gamma(t) \geq \gamma.$$

Zero-Sum Games

- Definition:
  - payoff matrix $\mathbf{M} = (\mathbf{M}_{ij}) \in \mathbb{R}^{m \times n}$.
  - m possible actions (pure strategy) for row player.
  - n possible actions for column player.
  - $\mathbf{M}_{ij}$ payoff for row player ($=$ loss for column player) when row plays i, column plays j.
- Example:

	rock	paper	scissors
rock	0	-1	1
paper	1	0	-1
scissors	-1	1	0

Mixed Strategies

(von Neumann, 1928)

- Definition: player row selects a distribution $\mathbf{p}$ over the rows, player column a distribution $\mathbf{q}$ over columns. The expected payoff for row is

$$\mathop{\mathrm{E}}_{\substack{i \sim p \\ j \sim q}}[\mathbf{M}_{ij}] = \sum_{i=1}^{m} \sum_{j=1}^{n} p_i \mathbf{M}_{ij} q_j = \mathbf{p}^\top \mathbf{M} \mathbf{q}.$$

- von Neumann's minimax theorem:

$$\max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p}^\top \mathbf{M} \mathbf{q} = \min_{\mathbf{q}} \max_{\mathbf{p}} \mathbf{p}^\top \mathbf{M} \mathbf{q}.$$

  - equivalent form:

$$\max_{\mathbf{p}} \min_{j \in [1, n]} \mathbf{p}^\top \mathbf{M} \mathbf{e}_j = \min_{\mathbf{q}} \max_{i \in [1, m]} \mathbf{e}_i^\top \mathbf{M} \mathbf{q}.$$

John von Neumann (1903 - 1957)

AdaBoost and Game Theory

- Game:
  - Player A: selects point x_i, $i \in [1, m]$.
  - Player B: selects base hypothesis h_t, $t \in [1, T]$.
  - Payoff matrix $\mathbf{M} \in \{-1, +1\}^{m \times T}$: $\mathbf{M}_{it} = y_i h_t(x_i)$.
- von Neumann's theorem: assume finite H.

$$2\gamma^* = \min_D \max_{h \in H} \sum_{i=1}^{m} D(i) y_i h(x_i) = \max_{\boldsymbol{\alpha}} \min_{i \in [1,m]} y_i \sum_{t=1}^{T} \frac{\alpha_t h_t(x_i)}{\|\boldsymbol{\alpha}\|_1} = \rho^*.$$

Consequences

- Weak learning condition $\implies$ non-zero margin.
  - thus, possible to search for non-zero margin.
  - AdaBoost $=$ (suboptimal) search for corresponding α; achieves at least half of the maximum margin.
- Weak learning $=$ strong condition:
  - the condition implies linear separability with margin $2\gamma^* > 0$.

Linear Programming Problem

- Maximizing the margin:

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

- This is equivalent to the following convex optimization LP problem:

$$\max_{\boldsymbol{\alpha}} \rho$$
$$\text{subject to}: y_i(\boldsymbol{\alpha} \cdot \mathbf{x}_i) \geq \rho$$
$$\|\boldsymbol{\alpha}\|_1 = 1.$$

- Note that:

$$\frac{|\boldsymbol{\alpha} \cdot \mathbf{x}|}{\|\boldsymbol{\alpha}\|_1} = \|\mathbf{x} - H\|_\infty, \text{ with } H = \{\mathbf{x} \colon \boldsymbol{\alpha} \cdot \mathbf{x} = 0\}.$$

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.

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).

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).

Other Boosting Algorithms

- arc-gv (Breiman, 1996): designed to maximize the margin, but outperformed by AdaBoost in experiments (Reyzin and Schapire, 2006).
- L1-regularized AdaBoost (Raetsch et al., 2001): outperfoms AdaBoost in experiments (Cortes et al., 2014).
- DeepBoost (Cortes et al., 2014): more favorable learning guarantees, outperforms both AdaBoost and L1-regularized AdaBoost in experiments.

References

- Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. In *ICML*, pages 262-270, 2014.

- Leo Breiman. Bagging predictors. *Machine Learning*, 24(2): 123-140, 1996.

- 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.

References

- 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.

- 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.

- Reyzin, Lev and Schapire, Robert E. How boosting the margin can also boost classifier complexity. In *ICML*, pages 753-760, 2006.

- 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 and Yoav Freund. *Boosting, Foundations and Algorithms*. The MIT Press, 2012.

- 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.