Boosting Algorithms for Maximizing the Soft Margin

Gunnar Rätsch

Friedrich Miescher Laboratory of the Max Planck Society, Tübingen, Germany

February 11, 2008

4^{th} Workshop on Ensemble Methods

Protocol of Boosting

- Maintain a distribution $\mathbf{d}^t$ on the examples
- At iteration $t = 1, \ldots, T$:
  1. Receive a "weak" hypothesis h_t
  2. Update $\mathbf{d}^t$ to $\mathbf{d}^{t+1}$, put more weights on "hard" examples
- Output a convex combination of the weak hypotheses

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

[Freund & Schapire, 1995]

Protocol of Boosting

- Maintain a distribution $\mathbf{d}^t$ on the examples
- At iteration $t = 1, \ldots, T$:
  1. Receive a "weak" hypothesis h_t
  2. Update $\mathbf{d}^t$ to $\mathbf{d}^{t+1}$, put more weights on "hard" examples
- Output a convex combination of the weak hypotheses

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

[Freund & Schapire, 1995]

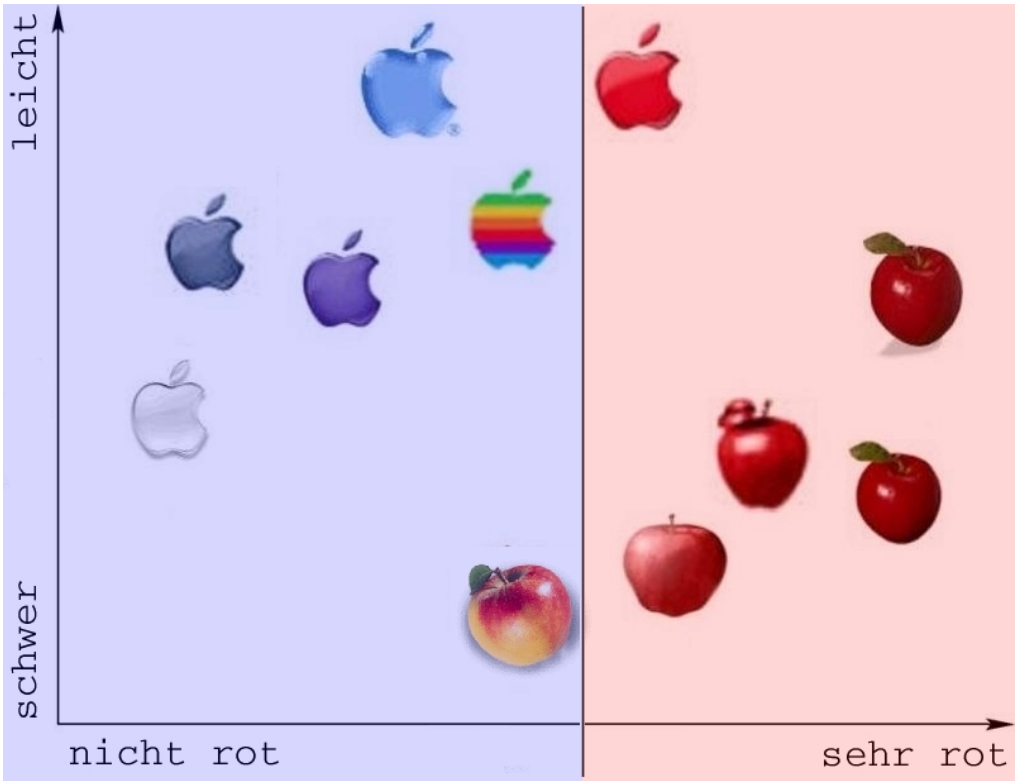
Protocol of Boosting

- Maintain a distribution $\mathbf{d}^t$ on the examples
- At iteration $t = 1, \ldots, T$:
  1. Receive a "weak" hypothesis h_t
  2. Update $\mathbf{d}^t$ to $\mathbf{d}^{t+1}$, put more weights on "hard" examples
- Output a convex combination of the weak hypotheses

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

[Freund & Schapire, 1995]

Boosting: 1st Iteration



First hypothesis:

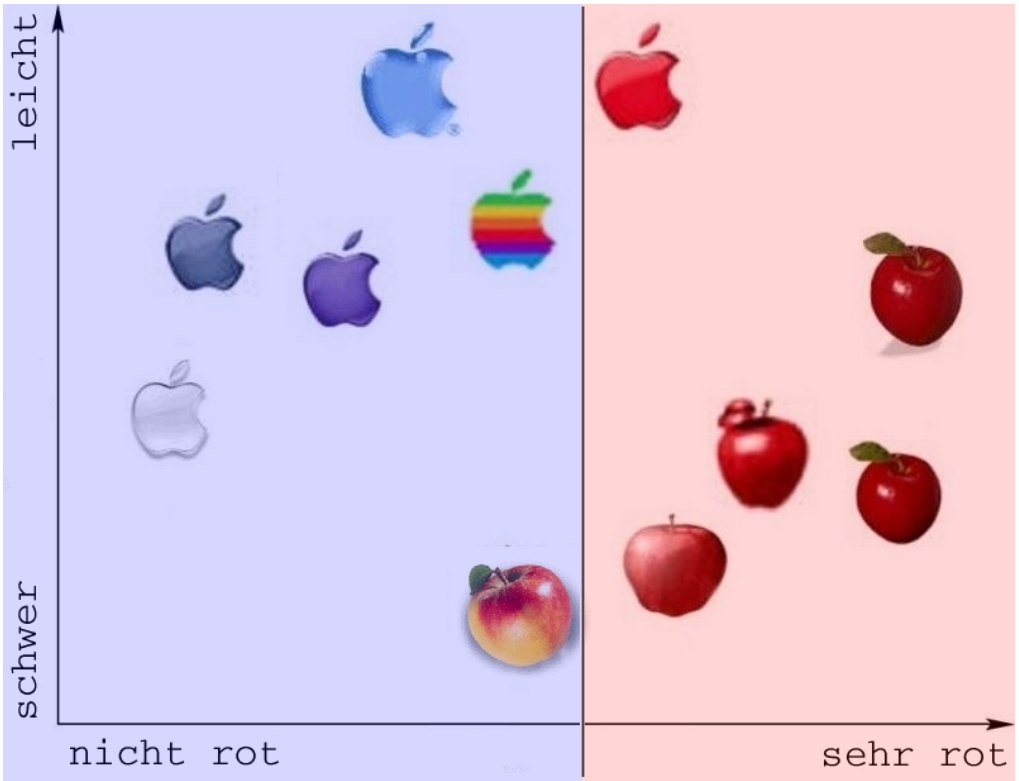
- Error rate: $\frac{2}{11}$

$$\epsilon_t = \sum_{n=1}^{N} d_n^t \mathbf{I}(h_t(\mathbf{x}_n) = y_n)$$

- Edge: $\frac{9}{22}$

$$\gamma_t = \sum_{n=1}^{N} d_n^t y_n h_t(\mathbf{x}_n) = 1 - 2\epsilon_t$$

Boosting: 1st Iteration



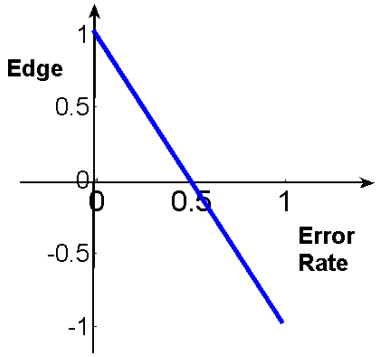
First hypothesis:

- Error rate: $\frac{2}{11}$

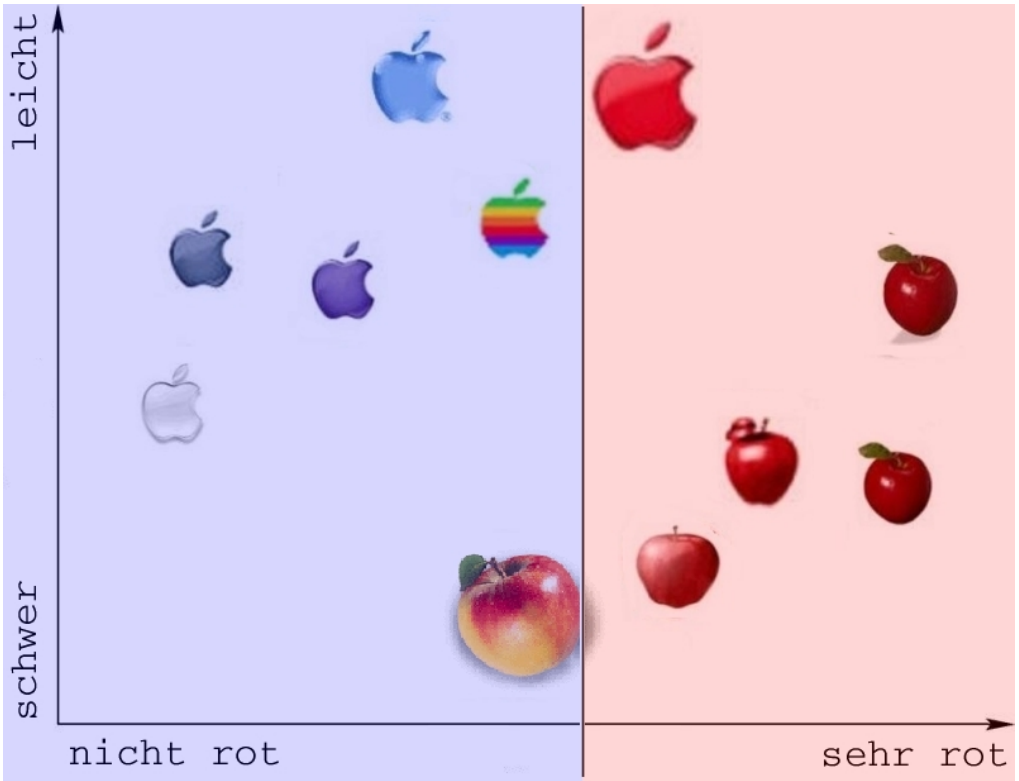
$$\epsilon_t = \sum_{n=1}^{N} d_n^t \mathbf{I}(h_t(\mathbf{x}_n) = y_n)$$

- Edge: $\frac{9}{22}$

$$\gamma_t = \sum_{n=1}^{N} d_n^t y_n h_t(\mathbf{x}_n)$$
$$= 1 - 2\epsilon_t$$



Update Distribution

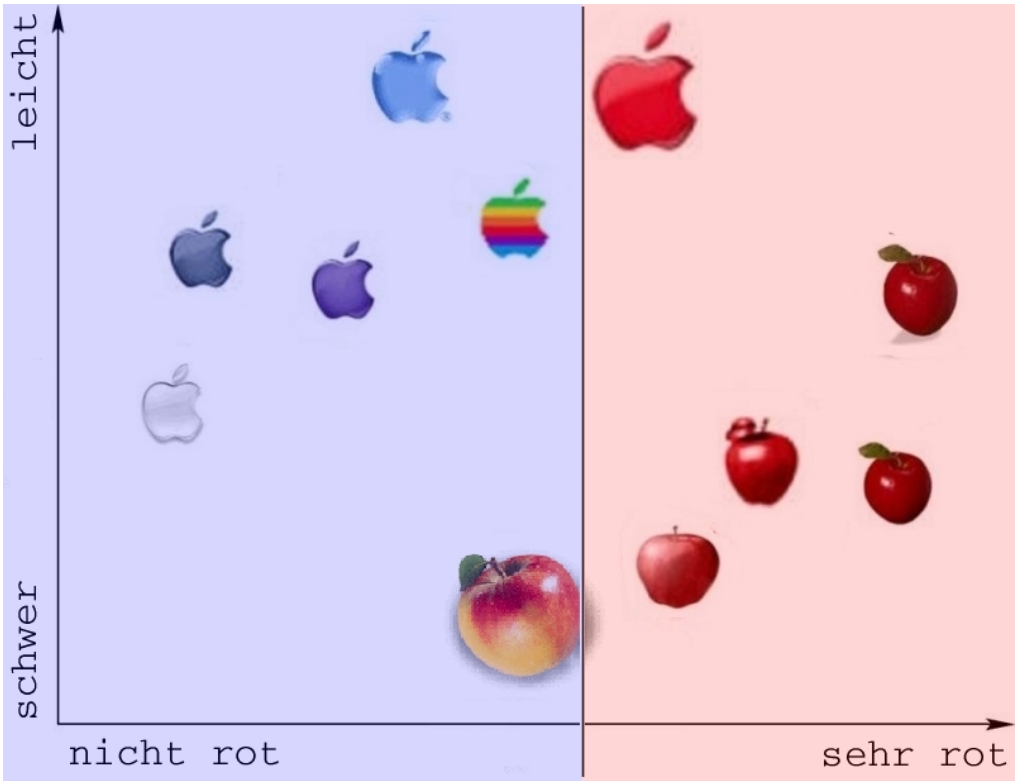


Misclassified examples
$\Rightarrow$ Increased weights

After update:

- Error rate:
  $\epsilon(h_t, \mathbf{d}^{t+1}) = \frac{1}{2}$
- Edge:
  $\gamma(h_t, \mathbf{d}^{t+1}) = 0$

Update Distribution

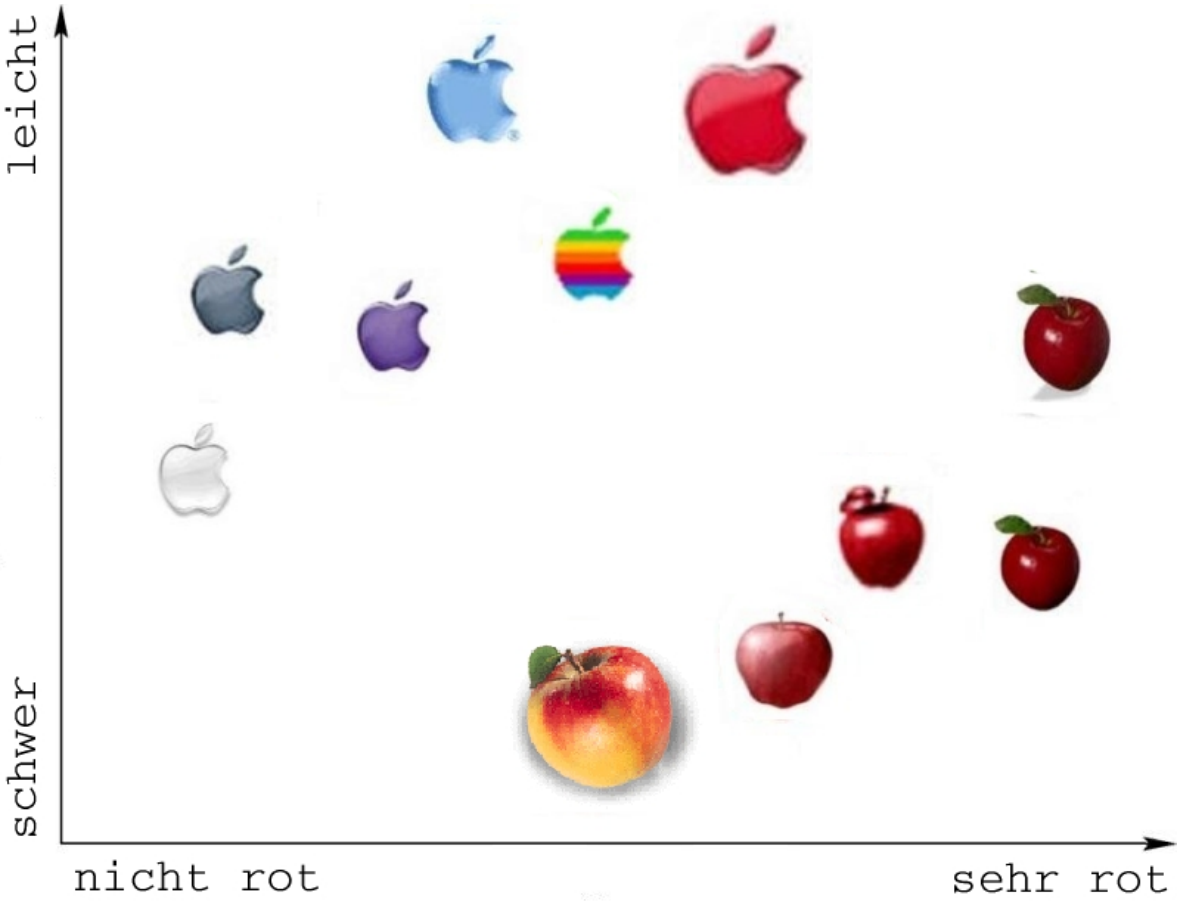


Misclassified examples
$\Rightarrow$ Increased weights

After update:

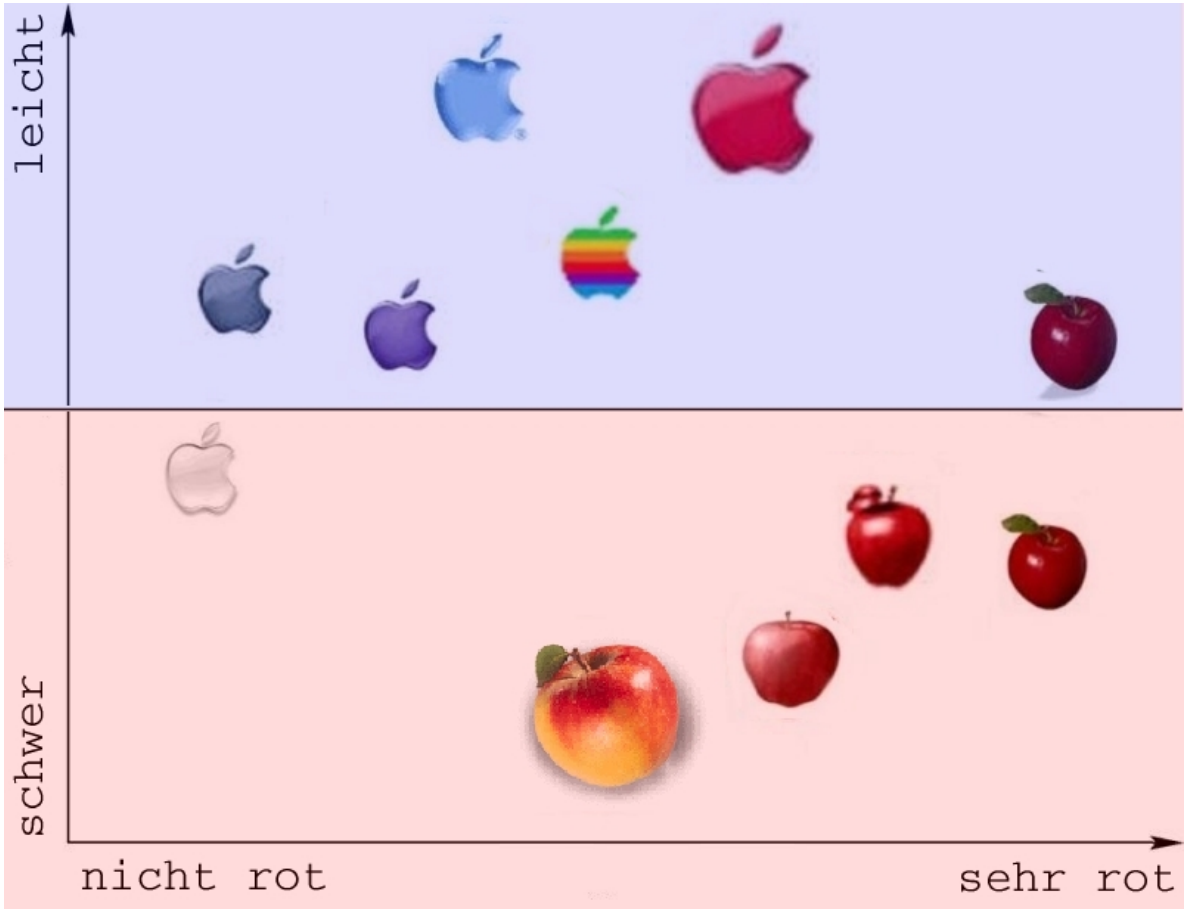
- Error rate:
  $\epsilon(h_t, \mathbf{d}^{t+1}) = \frac{1}{2}$
- Edge:
  $\gamma(h_t, \mathbf{d}^{t+1}) = 0$

Before 2nd Iteration

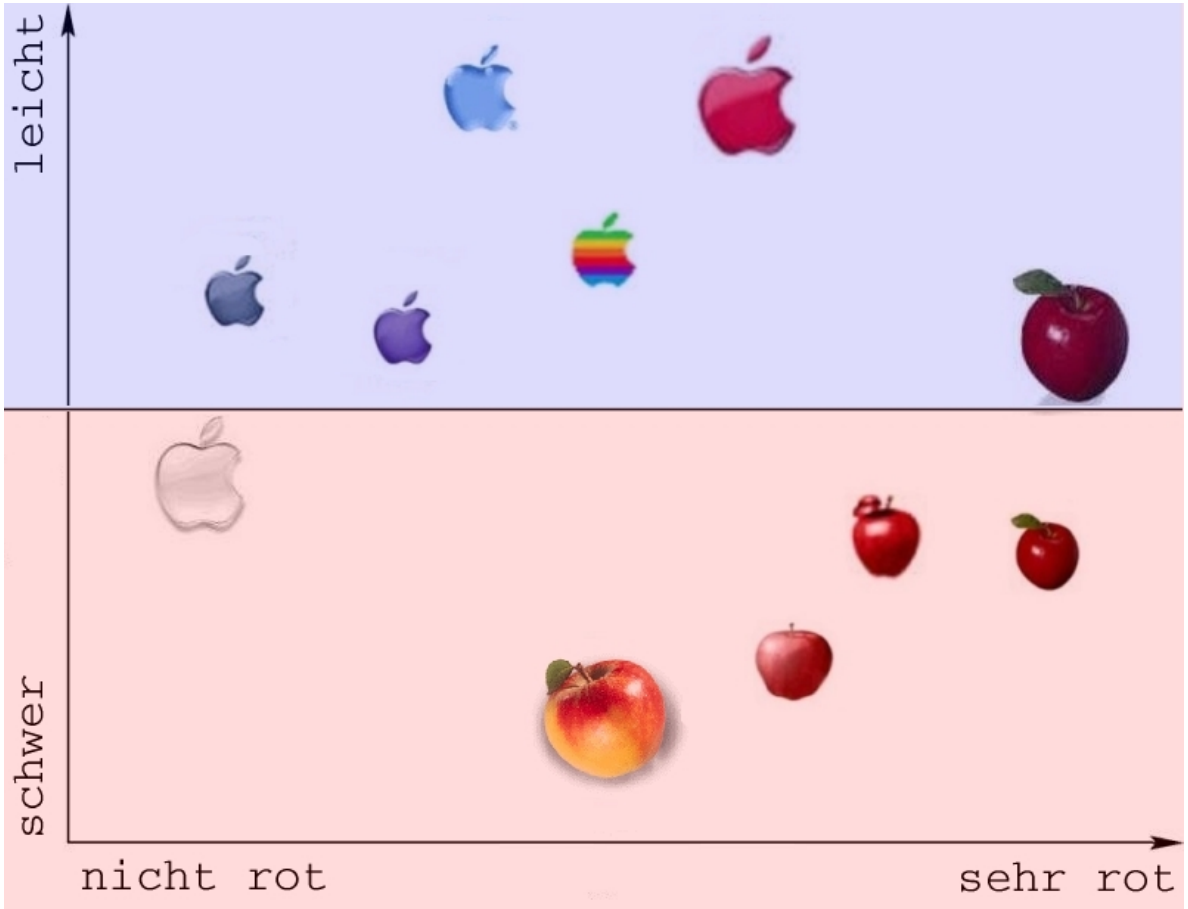


Boosting: 2nd Hypothesis

Edge $\gamma > \delta$



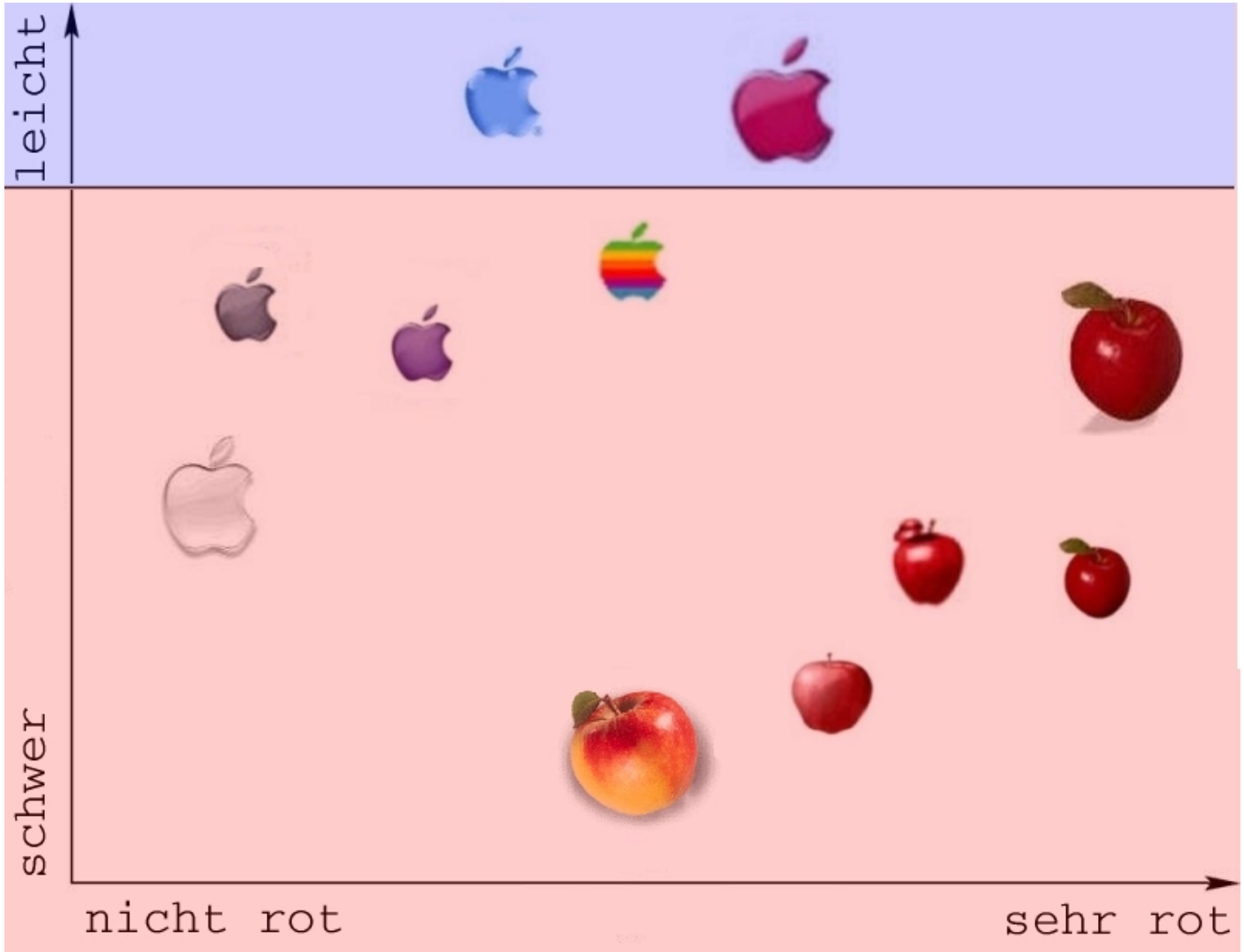
Update Distribution



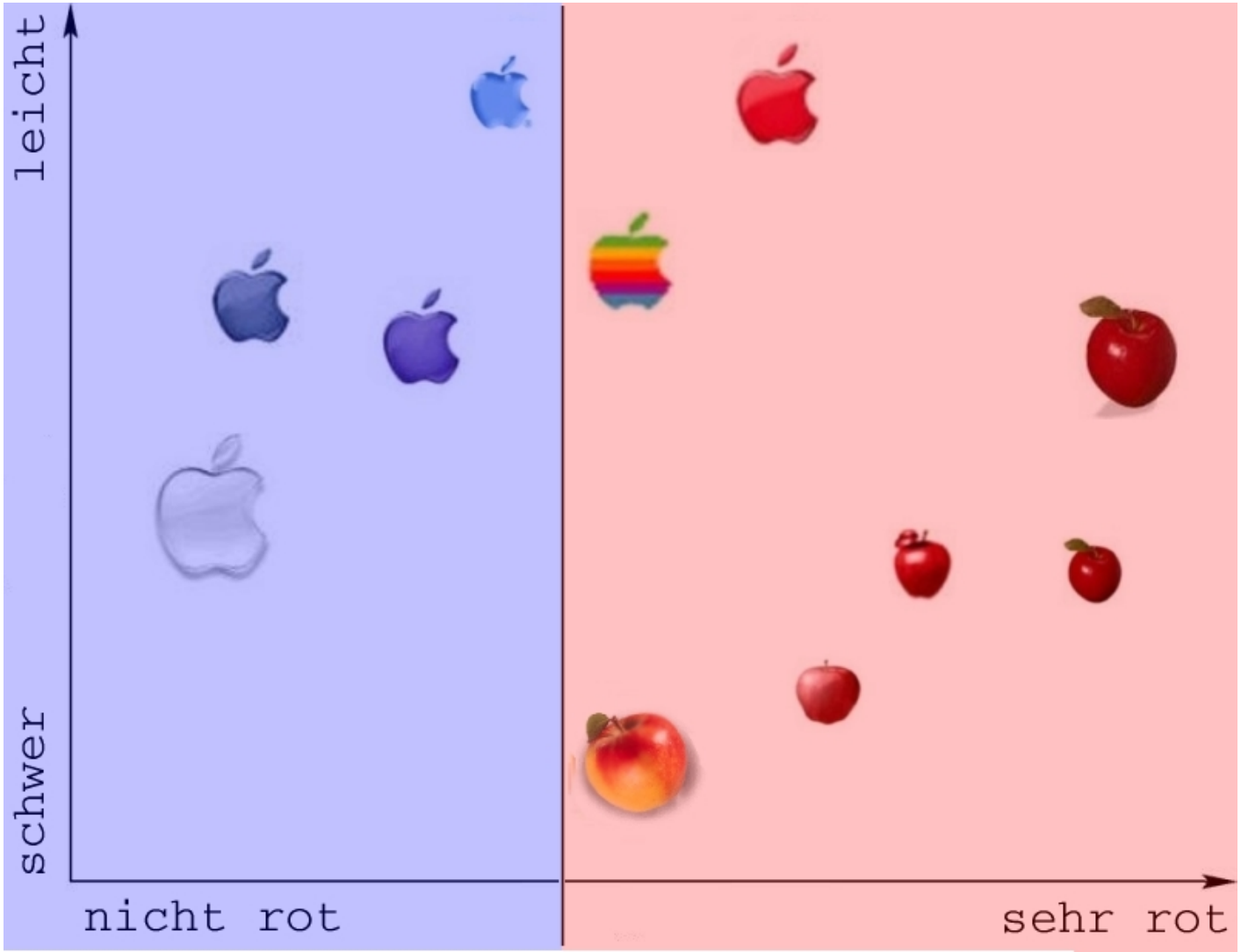
Edge $\gamma = 0$

AdaBoost update sets edge of last hypothesis to 0

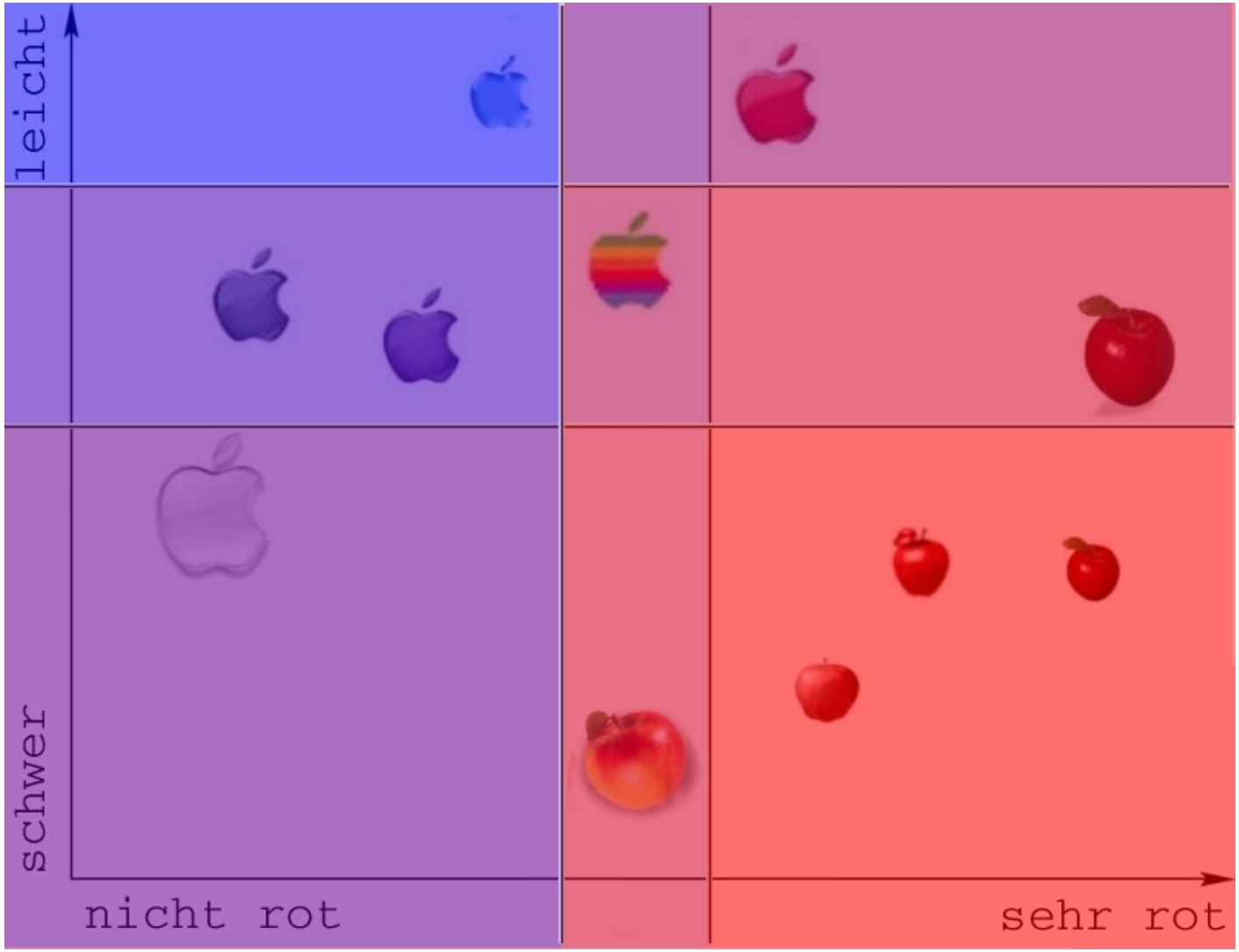
Boosting: 3nd Hypothesis



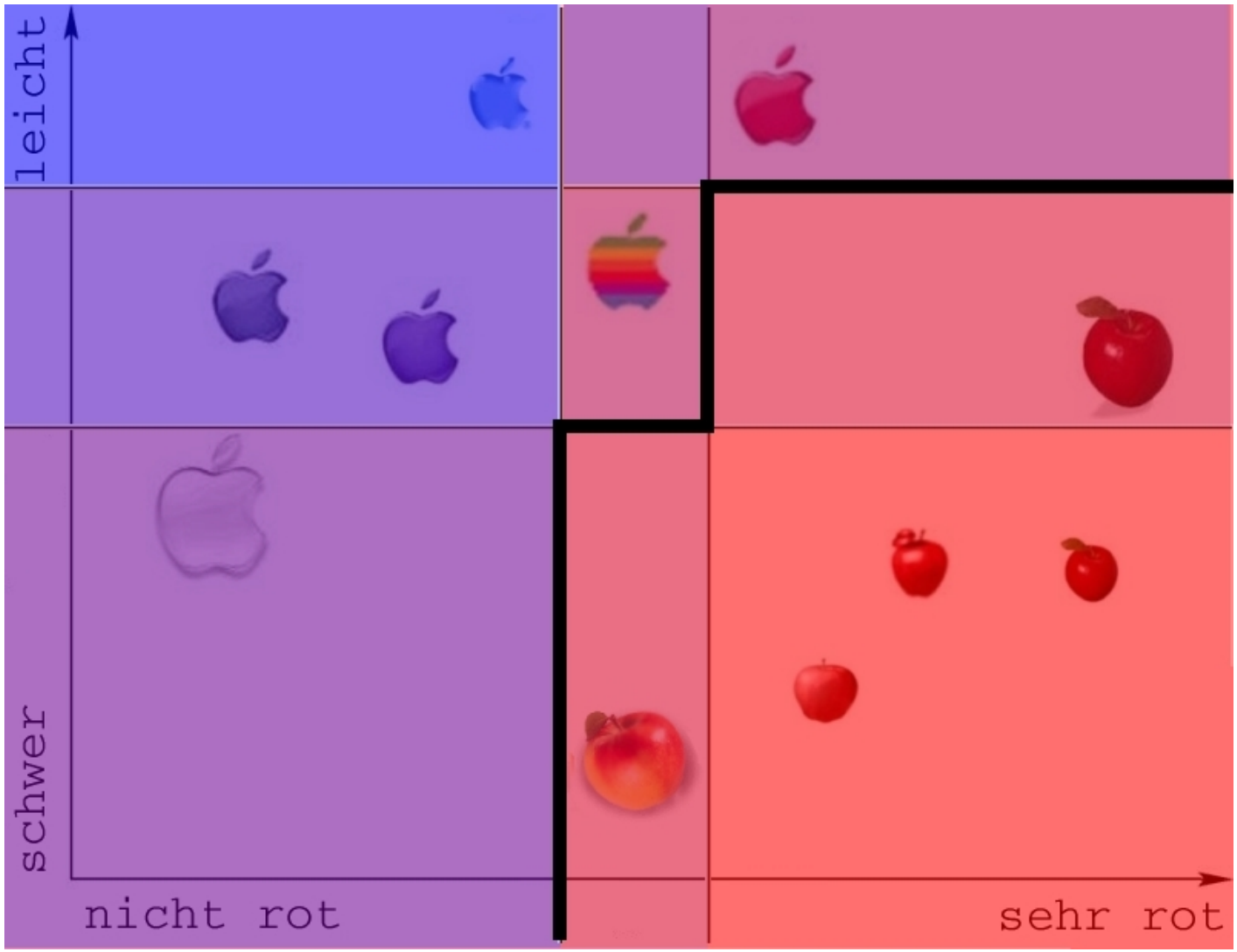
Boosting: 4th Hypothesis



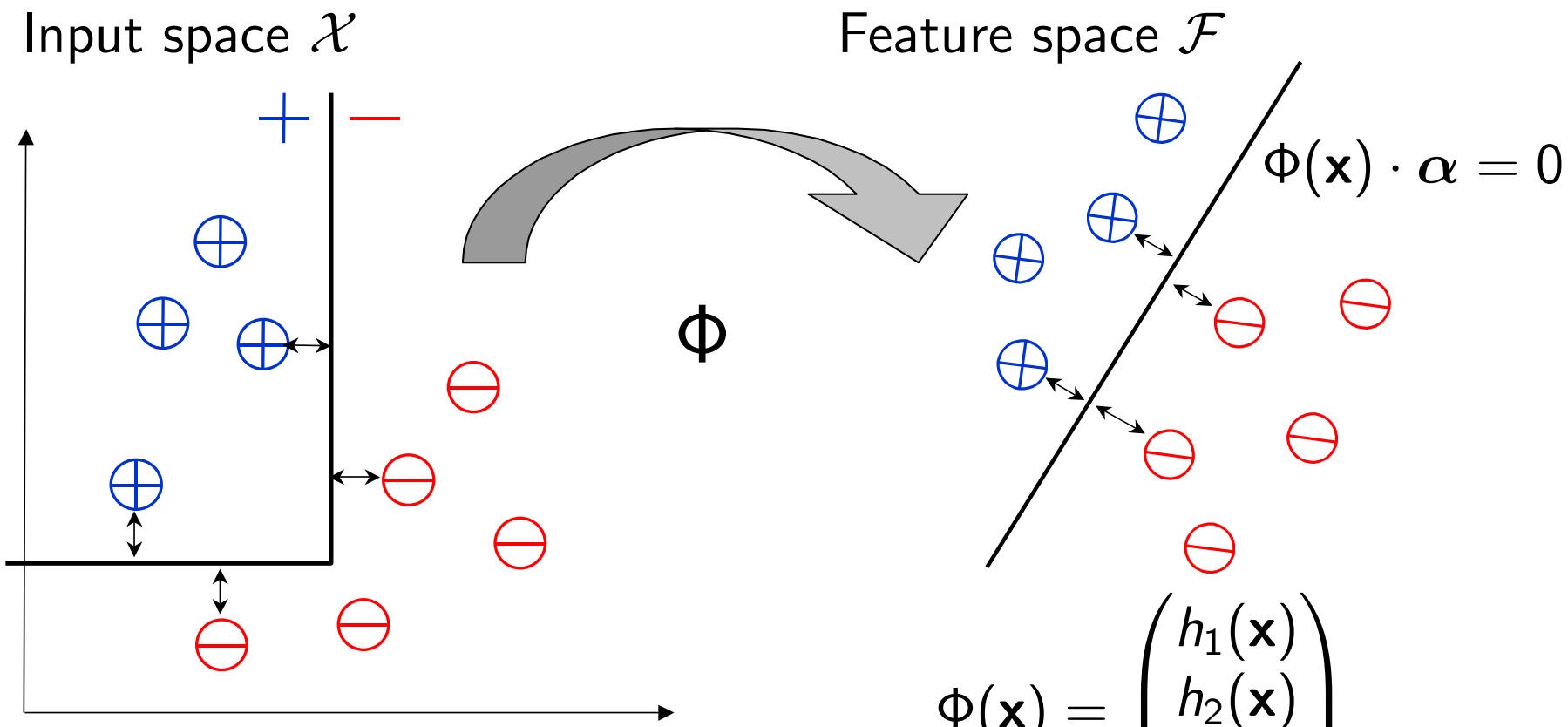
All Hypotheses



Decision: $f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) > 0$?



Large Margin and Linear Separation



$$\Phi(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \vdots \end{pmatrix}$$

$$\mathcal{H} = \{h_1, h_2, \ldots\}$$

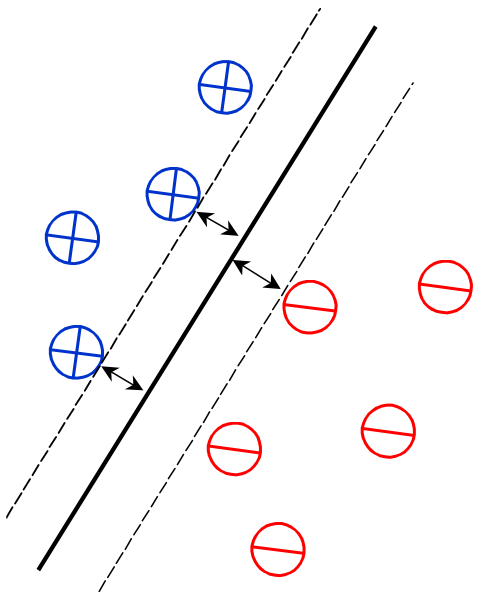
Linear separation in $\mathcal{F}$ is
nonlinear separation in $\mathcal{X}$

[Mangasarian, 1999; G.R., Mika, Schölkopf & Müller, 2002]

Large Margins

- Margin of the combined hypothesis $f_{\boldsymbol{\alpha}}$ for example $(\mathbf{x}_n, y_n)$

$$\begin{aligned}\rho_n(\boldsymbol{\alpha}) &= y(\Phi(\mathbf{x}) \cdot \boldsymbol{\alpha}) \\ &= y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n) \\ &= y_n \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_n) \qquad (\boldsymbol{\alpha} \in \mathcal{P}^T)\end{aligned}$$



Margin of set of examples is minimum over examples

$$\rho(\boldsymbol{\alpha}) := \min_n \rho_n(\boldsymbol{\alpha})$$

[Freund, Schapire, Bartlett & Lee, 1998]

Edge vs. Margin

Edge

- Measurement of "goodness" of a hypothesis w.r.t. a distribution
- Edge of a hypothesis h for a distribution $\mathbf{d}$ on the examples

$$\gamma_h(\mathbf{d}) = \sum_{n=1}^{N} d_n \; y_n h(\mathbf{x}_n) \qquad \mathbf{d} \in \mathcal{P}^N$$

Margin

- Measure for "confidence" in prediction for a hypothesis weighting
- Margin of example n for current hypothesis weighting α

$$\rho_n(\alpha) \;=\; y_n f_\alpha(\mathbf{x}_n) = y_n \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_n) \qquad (\alpha \in \mathcal{P}^T)$$

What is the connection?

[Breiman, 1999]

Edge vs. Margin

Edge

- Measurement of "goodness" of a hypothesis w.r.t. a distribution
- Edge of a hypothesis h for a distribution $\mathbf{d}$ on the examples

$$\gamma_h(\mathbf{d}) = \sum_{n=1}^{N} d_n \; y_n h(\mathbf{x}_n) \qquad \mathbf{d} \in \mathcal{P}^N$$

Margin

- Measure for "confidence" in prediction for a hypothesis weighting
- Margin of example n for current hypothesis weighting $\boldsymbol{\alpha}$

$$\rho_n(\boldsymbol{\alpha}) \;=\; y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n) = y_n \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_n) \qquad (\boldsymbol{\alpha} \in \mathcal{P}^T)$$

What is the connection?

[Breiman, 1999]

Edge vs. Margin

Edge

- Measurement of "goodness" of a hypothesis w.r.t. a distribution
- Edge of a hypothesis h for a distribution $\mathbf{d}$ on the examples

$$\gamma_h(\mathbf{d}) = \sum_{n=1}^{N} d_n \; y_n h(\mathbf{x}_n) \qquad \mathbf{d} \in \mathcal{P}^N$$

Margin

- Measure for "confidence" in prediction for a hypothesis weighting
- Margin of example n for current hypothesis weighting $\boldsymbol{\alpha}$

$$\rho_n(\boldsymbol{\alpha}) \;=\; y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n) = y_n \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_n) \qquad (\boldsymbol{\alpha} \in \mathcal{P}^T)$$

What is the connection?

[Breiman, 1999]

von Neumann's Minimax-Theorem

Set of examples $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
and hypotheses set $\mathcal{H}^t = \{h_1, \ldots, h_t\}$,

$$\text{minimum edge: } \gamma_t^* = \min_{\mathbf{d} \in \mathcal{P}^N} \underbrace{\max_{h \in \mathcal{H}^t} \overbrace{\gamma_h(\mathbf{d})}^{\text{edge of } h}}_{\text{edge of } \mathcal{H}^t}$$

$$\text{maximum margin: } \rho_t^* = \max_{\boldsymbol{\alpha} \in \mathcal{P}^t} \underbrace{\min_n \overbrace{y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)}^{\text{margin of } (\mathbf{x}_n, y_n)}}_{\text{margin of S}}$$

$$\text{Duality: } \gamma_t^* = \rho_t^*$$

[von Neumann, 1928]

Duality gap

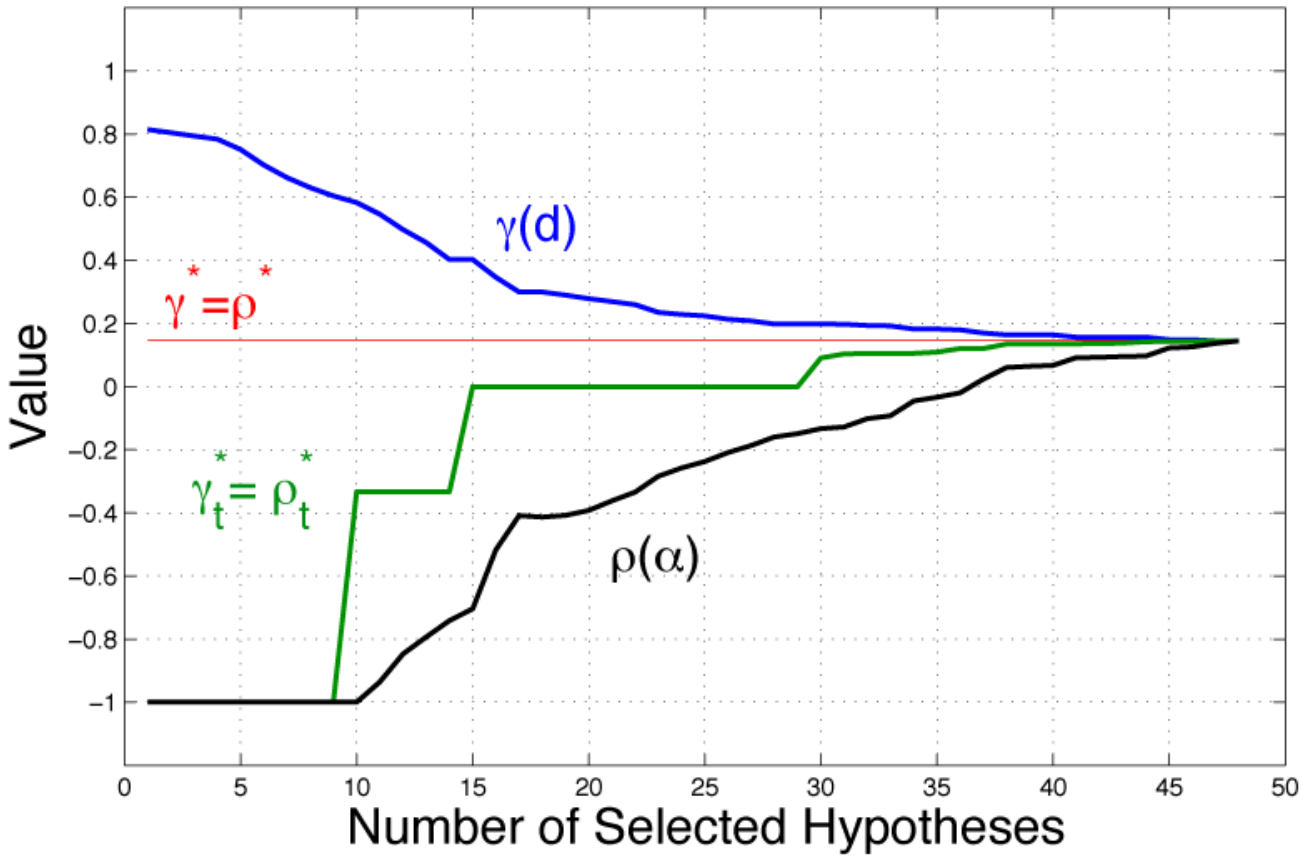
For any non-optimal $\mathbf{d} \in \mathcal{P}^N$ and $\boldsymbol{\alpha} \in \mathcal{P}^t$,

$$\gamma(\mathbf{d}) \geq \gamma_t^* = \rho_t^* \geq \rho(\boldsymbol{\alpha})$$

Duality gap

For any non-optimal $\mathbf{d} \in \mathcal{P}^N$ and $\boldsymbol{\alpha} \in \mathcal{P}^t$,

$$\gamma(\mathbf{d}) \geq \gamma_t^* = \rho_t^* \geq \rho(\boldsymbol{\alpha})$$



How Large is the Maximal Margin?

Assumptions on Weak learner

For any distribution $\mathbf{d}$ on the examples, the weak learner returns a hypothesis h with edge $\gamma_h(\mathbf{d})$ at least g.
Best case: $g = \rho^* = \gamma^*$.

[Breiman, 1999; Bennett et al.; G.R. et al., 2001; Rudin et al., 2004]

How Large is the Maximal Margin?

Assumptions on Weak learner

For any distribution $\mathbf{d}$ on the examples, the weak learner returns a hypothesis h with edge $\gamma_h(\mathbf{d})$ at least g.
Best case: $g = \rho^* = \gamma^*$.

Implication from Minimax Theorem

There exists $\boldsymbol{\alpha} \in \mathcal{P}^N$, such that $\rho(\boldsymbol{\alpha}) \geq g$.

[Breiman, 1999; Bennett et al.; G.R. et al., 2001; Rudin et al., 2004]

How Large is the Maximal Margin?

Assumptions on Weak learner

For any distribution $\mathbf{d}$ on the examples, the weak learner returns a hypothesis h with edge $\gamma_h(\mathbf{d})$ at least g.
Best case: $g = \rho^* = \gamma^*$.

Implication from Minimax Theorem

There exists $\boldsymbol{\alpha} \in \mathcal{P}^N$, such that $\rho(\boldsymbol{\alpha}) \geq g$.

Idea to iteratively solve LP: LPBoost

Add "best" hypothesis $h = \operatorname{argmax} \gamma_h(\mathbf{d}^t)$ to $\mathcal{H}^{t+1}$ and resolve

$$\mathbf{d}^{t+1} = \operatorname*{argmin}_{d \in \mathcal{P}^N} \max_{h \in \mathcal{H}^{t+1}} \gamma_h(\mathbf{d}).$$

[Breiman, 1999; Bennett et al.; G.R. et al., 2001; Rudin et al., 2004]

How Large is the Maximal Margin?

Assumptions on Weak learner

For any distribution $\mathbf{d}$ on the examples, the weak learner returns a hypothesis h with edge $\gamma_h(\mathbf{d})$ at least g.
Best case: $g = \rho^* = \gamma^*$.

What about AdaBoost?

- May "oscillate"
- Does not find maximizing $\boldsymbol{\alpha}$ (counter examples)
- But there some guarantees:
  - $\rho(\boldsymbol{\alpha}^t) \geq 0$ after $2 \ln N / g^2$ iterations
  - $\rho(\boldsymbol{\alpha}^t) \geq g/2$ in the limit

[Breiman, 1999; Bennett et al.; G.R. et al., 2001; Rudin et al., 2004]

How to Maximize the Margin?

Modify AdaBoost for maximizing margin

- Arc-GV asymptotically maximizes the margin
  - quite slow, no converge rates
- LPBoost uses a Linear Programming solver
  - Often very fast in practice, but no converge rates
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice: just as fast as theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Fast in practice
  - Combination of benefits

[Breiman, 1999, G.R. & Warmuth, 2004; Warmuth et al., 2006]

How to Maximize the Margin?

Modify AdaBoost for maximizing margin

- Arc-GV asymptotically maximizes the margin
  - quite slow, no converge rates
- LPBoost uses a Linear Programming solver
  - Often very fast in practice, but no converge rates
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice: just as fast as theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Fast in practice
  - Combination of benefits

[Breiman, 1999, G.R. & Warmuth, 2004; Warmuth et al., 2006]

How to Maximize the Margin?

Modify AdaBoost for maximizing margin

- Arc-GV asymptotically maximizes the margin
  - quite slow, no converge rates
- LPBoost uses a Linear Programming solver
  - Often very fast in practice, but no converge rates
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice: just as fast as theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Fast in practice
  - Combination of benefits

[Breiman, 1999, G.R. & Warmuth, 2004; Warmuth et al., 2006]

How to Maximize the Margin?

Modify AdaBoost for maximizing margin

- Arc-GV asymptotically maximizes the margin
  - quite slow, no converge rates
- LPBoost uses a Linear Programming solver
  - Often very fast in practice, but no converge rates
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice: just as fast as theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Fast in practice
  - Combination of benefits

[Breiman, 1999, G.R. & Warmuth, 2004; Warmuth et al., 2006]

Idea: Projections to $\hat{\gamma}_t$ instead of 0

Separation

$$\min_{\mathbf{d}\in\mathcal{P}^N} \quad \Delta(\mathbf{d}, \mathbf{d}^t) \qquad \text{(AdaBoost)}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_t(\mathbf{x}_n) \leq 0 \qquad \text{for } r = 1, \ldots, t$$

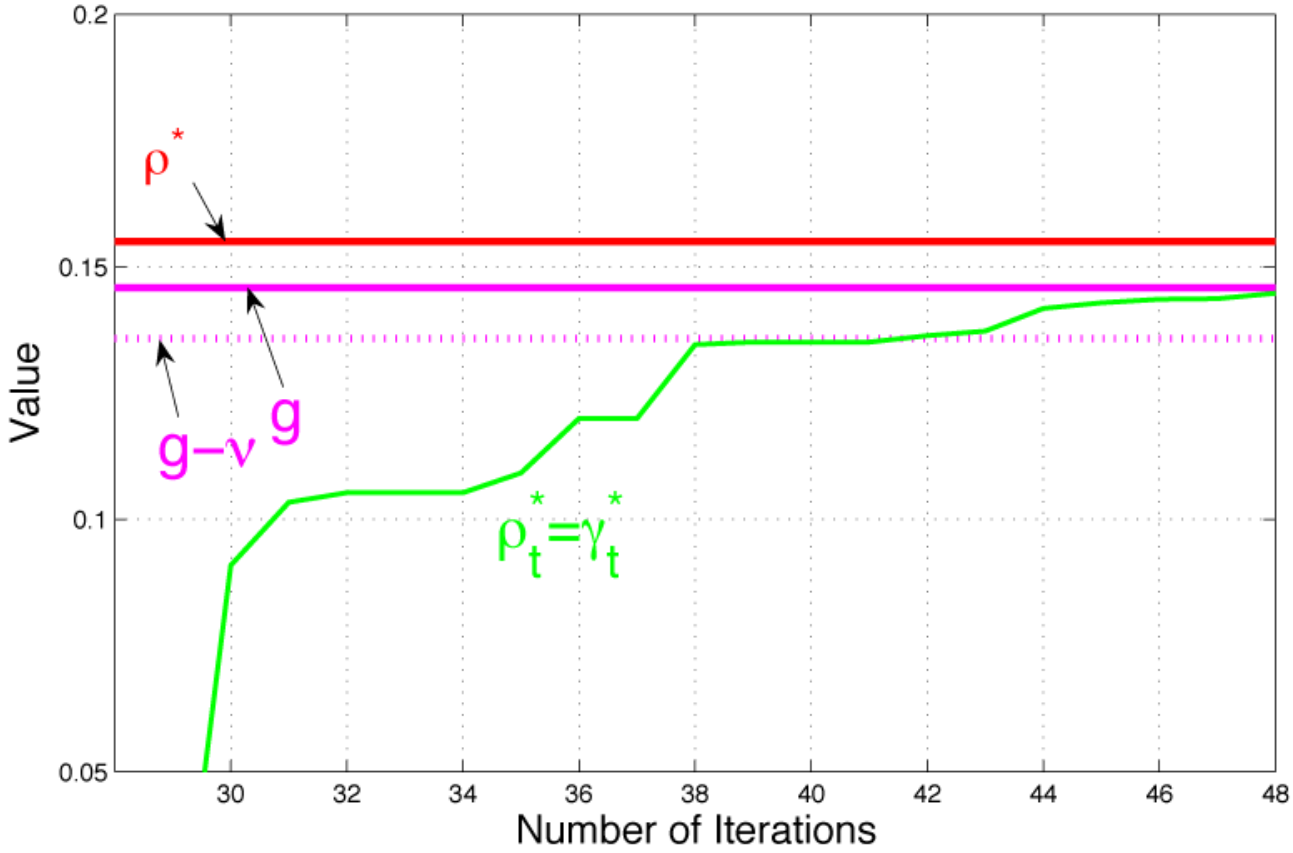
Large Margin Separation

$$\min_{\mathbf{d}\in\mathcal{P}^N} \quad \Delta(\mathbf{d}, \mathbf{d}^t) \qquad \text{(AdaBoost}^*\text{)}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_t(\mathbf{x}_n) \leq \hat{\gamma}_t \qquad \text{for } r = 1, \ldots, t$$

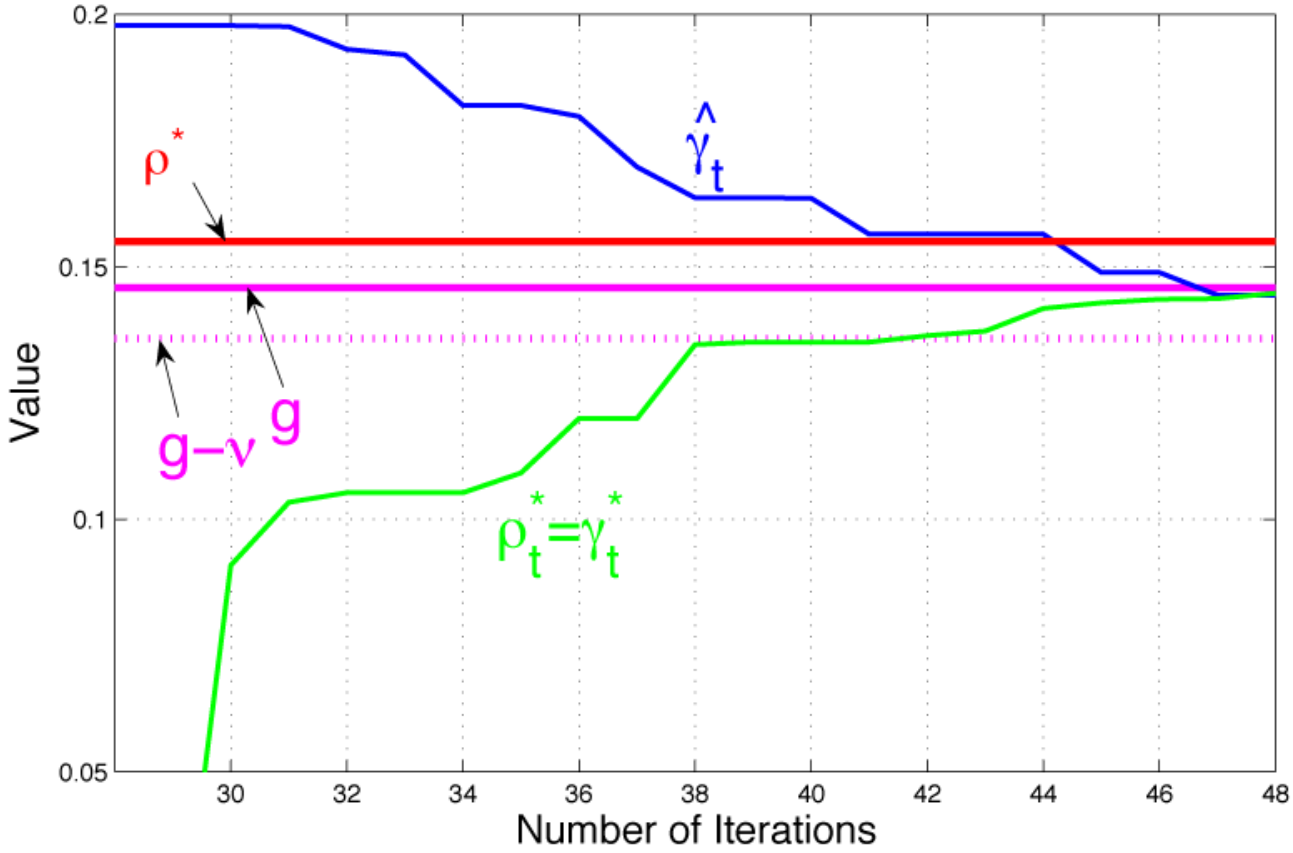
[G.R. & Warmuth, 2004]

Want margin $\geq g - \delta$



- Assumption: $\gamma_t \geq g$
- Estimate of target: $\hat{\gamma}_t = (\min_{q=1,\ldots,t} \gamma_q) - \delta$

Want margin $\geq g - \delta$



- Assumption: $\gamma_t \geq g$
- Estimate of target: $\hat{\gamma}_t = (\min_{q=1,\ldots,t} \gamma_q) - \delta$

Corrective vs. Totally Corrective Updates

Corrective: Single constraint

$$\min_{\mathbf{d}\in\mathcal{P}^N} \quad \Delta(\mathbf{d}, \mathbf{d}^t) \qquad \text{(AdaBoost}^*\text{)}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_t(\mathbf{x}_n) \leq \hat{\gamma}_t$$

Totally corrective: One constraint per past weak hypothesis

$$\min_{\mathbf{d}\in\mathcal{P}^N} \quad \Delta(\mathbf{d}, \mathbf{d}^1) \qquad \text{(TotalBoost)}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_r(\mathbf{x}_n) \leq \hat{\gamma}_t \qquad \text{for } r = 1, \ldots, t$$

[Kivinen & Warmuth, 1999; Warmuth et al., 2006]

TotalBoost$_\delta$

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, desired accuracy δ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$ and let $u_i^t = y_i h_t(\mathbf{x}_i)$
   2. Calculate the edge γ_t of h_t: $\gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   3. Set $\widehat{\gamma}_t = \left(\min_{q=1,\ldots,t} \gamma_q \right) - \delta$ and solve
      $$\mathbf{d}^{t+1} = \operatorname*{argmin}_{\{\mathbf{d} \in \mathcal{P}^N \mid \mathbf{d} \cdot \mathbf{u}^q \leq \widehat{\gamma}_t, \text{ for } 1 \leq q \leq t\} = \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$$
   4. **If** above infeasible or $\mathbf{d}^{t+1}$ contains a zero then $T = t$ and break
4. **Output:** $f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

TotalBoost$_\delta$

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, desired accuracy δ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$ and let $u_i^t = y_i h_t(\mathbf{x}_i)$
   2. Calculate the edge γ_t of h_t: $\gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   3. Set $\widehat{\gamma}_t = \left(\min_{q=1,\ldots,t} \gamma_q \right) - \delta$ and solve
      $$\mathbf{d}^{t+1} = \operatorname*{argmin}_{\{\mathbf{d} \in \mathcal{P}^N \mid \mathbf{d} \cdot \mathbf{u}^q \leq \widehat{\gamma}_t, \text{ for } 1 \leq q \leq t\} = \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$$
   4. **If** above infeasible or $\mathbf{d}^{t+1}$ contains a zero then $T = t$ and break
4. **Output:** $f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

TotalBoost$_\delta$

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, desired accuracy δ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$ and let $u_i^t = y_i h_t(\mathbf{x}_i)$
   2. Calculate the edge γ_t of h_t: $\gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   3. Set $\widehat{\gamma}_t = (\min_{q=1,\ldots,t} \gamma_q) - \delta$ and solve
   4.

Optimization Problem

$$\mathbf{d}^{t+1} = \operatorname*{argmin}_{\mathbf{d} \in \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$$

with $$\mathcal{C}_t := \{\mathbf{d} \in \mathcal{P}^N \mid \mathbf{d} \cdot \mathbf{u}^q \leq \widehat{\gamma}_t, \text{ for } 1 \leq q \leq t\}$$

4. **Output:** $f_\alpha(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

Iteration Bound $\lceil \frac{2 \ln N}{\delta^2} \rceil$

Theorem

Assume the base learner returns hypotheses with edge greater than g, then AdaBoost* and TotalBoost terminate after at most $\frac{2 \ln N}{\delta^2}$ iteration with margin at least $g - \delta$.

Lemma

For $\mathbf{d}^t, \mathbf{d}^{t+1} \in \mathcal{P}^N$ and $\mathbf{u} \in [-1, 1]^N$,
if $\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t)$ finite and $\mathbf{d}^{t+1} \cdot \mathbf{u} \neq \mathbf{d}^t \cdot \mathbf{u}$ then

$$\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t) > \frac{(\mathbf{d}^{t+1} \cdot \mathbf{u} - \mathbf{d}^t \cdot \mathbf{u})^2}{2}$$

[Warmuth, Liao & G.R., 2006]

Iteration Bound $\lceil \frac{2 \ln N}{\delta^2} \rceil$

Theorem

Assume the base learner returns hypotheses with edge greater than g, then AdaBoost* and TotalBoost terminate after at most $\frac{2 \ln N}{\delta^2}$ iteration with margin at least $g - \delta$.

Lemma

For $\mathbf{d}^t, \mathbf{d}^{t+1} \in \mathcal{P}^N$ and $\mathbf{u} \in [-1, 1]^N$,
if $\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t)$ finite and $\mathbf{d}^{t+1} \cdot \mathbf{u} \neq \mathbf{d}^t \cdot \mathbf{u}$ then

$$\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t) > \frac{(\mathbf{d}^{t+1} \cdot \mathbf{u} - \mathbf{d}^t \cdot \mathbf{u})^2}{2}$$

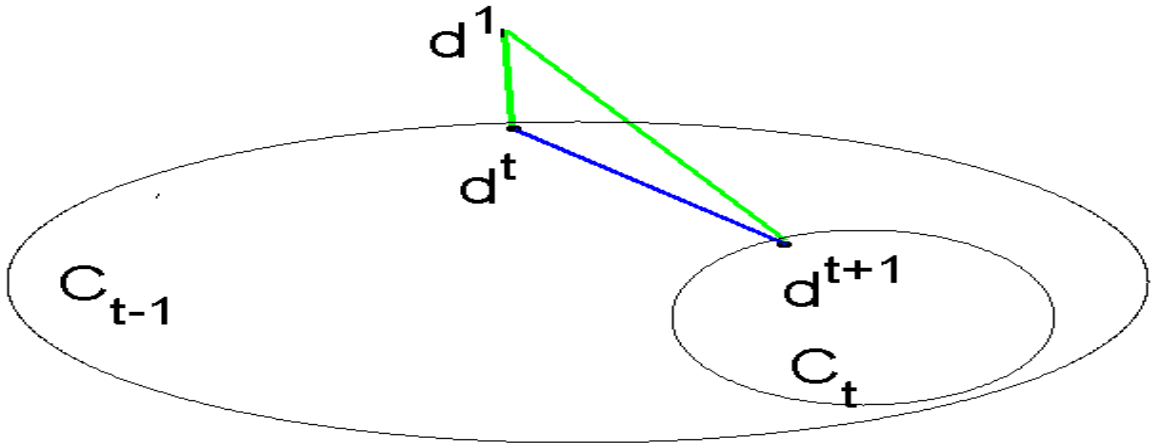
[Warmuth, Liao & G.R., 2006]

Generalized Pythagorean Theorem

$\mathcal{C}_t = \{\mathbf{d} \in \mathcal{P}^N | \mathbf{d} \cdot \mathbf{u}^q \leq \widehat{\gamma}_t, 1 \leq q \leq t\}, \mathcal{C}_0 = \mathcal{P}^N, \mathcal{C}_t \subseteq \mathcal{C}_{t-1}$
$\mathbf{d}^t$ is projection of $\mathbf{d}^1$ onto $\mathcal{C}_{t-1}$ at iteration $t-1$

$$\mathbf{d}^t = \operatorname*{argmin}_{\mathbf{d} \in \mathcal{C}_{t-1}} \Delta(\mathbf{d}, \mathbf{d}^1)$$

$$\Delta(\mathbf{d}^{t+1}, \mathbf{d}^1) \geq \Delta(\mathbf{d}^t, \mathbf{d}^1) + \Delta(\mathbf{d}^{t+1}, \mathbf{d}^t)$$



[Herbster & Warmuth, 2001]

Sketch of Proof

$$
\begin{array}{rccccc}
1: & \Delta(\mathbf{d}^2,\mathbf{d}^1) - \Delta(\mathbf{d}^1,\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^2,\mathbf{d}^1) & > & \frac{\delta^2}{2} \\
2: & \Delta(\mathbf{d}^3,\mathbf{d}^2) - \Delta(\mathbf{d}^2,\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^3,\mathbf{d}^2) & > & \frac{\delta^2}{2} \\
 & \dots & & \dots & & \\
t: & \Delta(\mathbf{d}^{t+1},\mathbf{d}^1) - \Delta(\mathbf{d}^t,\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^{t+1},\mathbf{d}^t) & > & \frac{\delta^2}{2} \\
 & \dots & & \dots & & \\
T-1: & \Delta(\mathbf{d}^T,\mathbf{d}^1) - \Delta(\mathbf{d}^{T-1},\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^T,\mathbf{d}^{T-1}) & > & \frac{\delta^2}{2}
\end{array}
$$

Therefore, $T \leq \lceil \frac{2 \ln N}{\delta^2} \rceil$

[Warmuth, Liao & G.R., 2006]

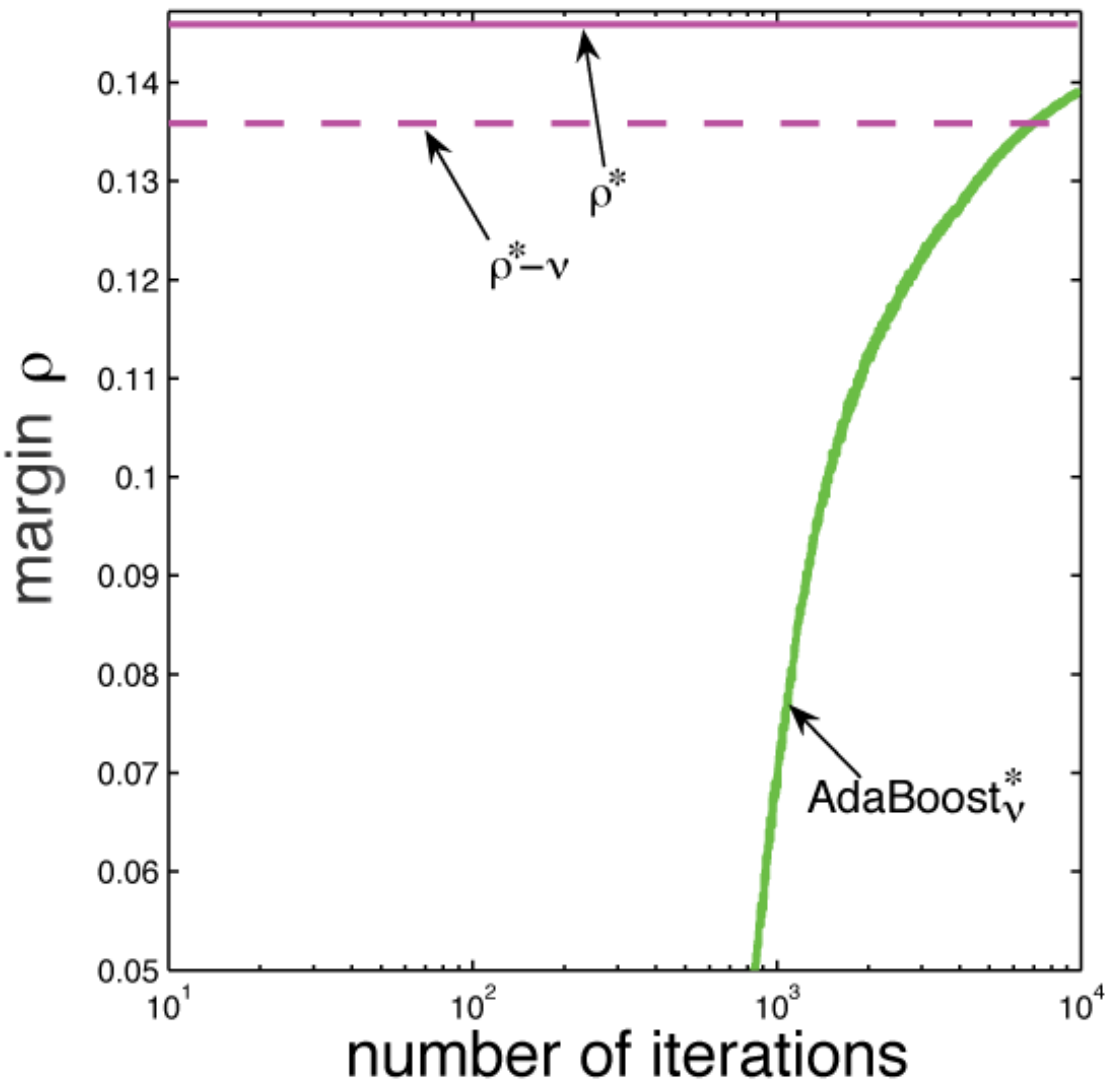
Illustrative Experiments

Cox-1 dataset from Telik Inc.

- Relatively small drug-design data set
  - 125 binary labeled examples
  - 3888 binary features
- Compare convergence of margin versus number of iterations

[Warmuth, Liao & G.R., 2006]

Illustrative Experiments

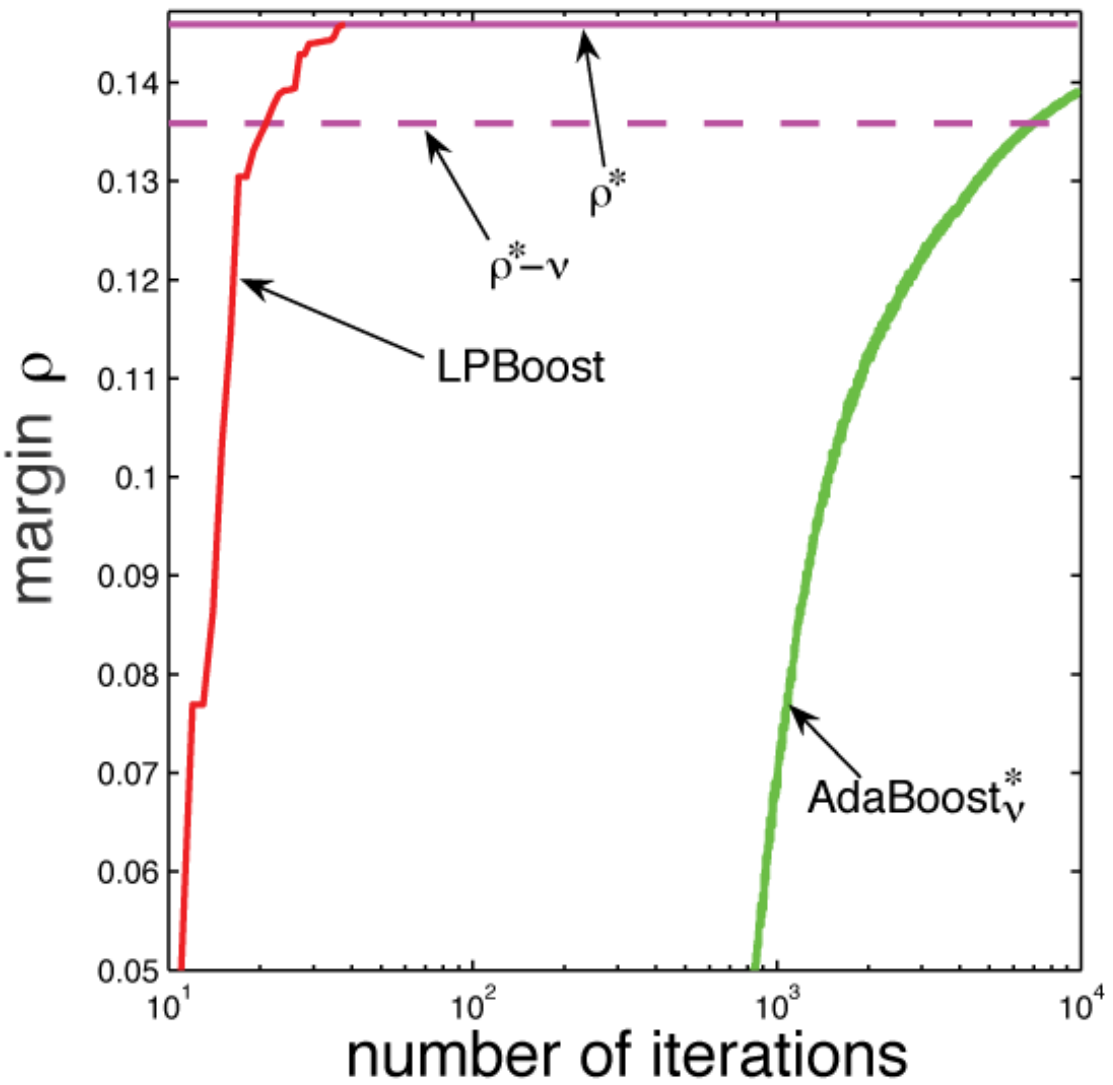


Cox-1 ($\delta = 0.01$)

- AdaBoost$^*_\delta$
- LPBoost
- TotalBoost$_\delta$

Results

Illustrative Experiments



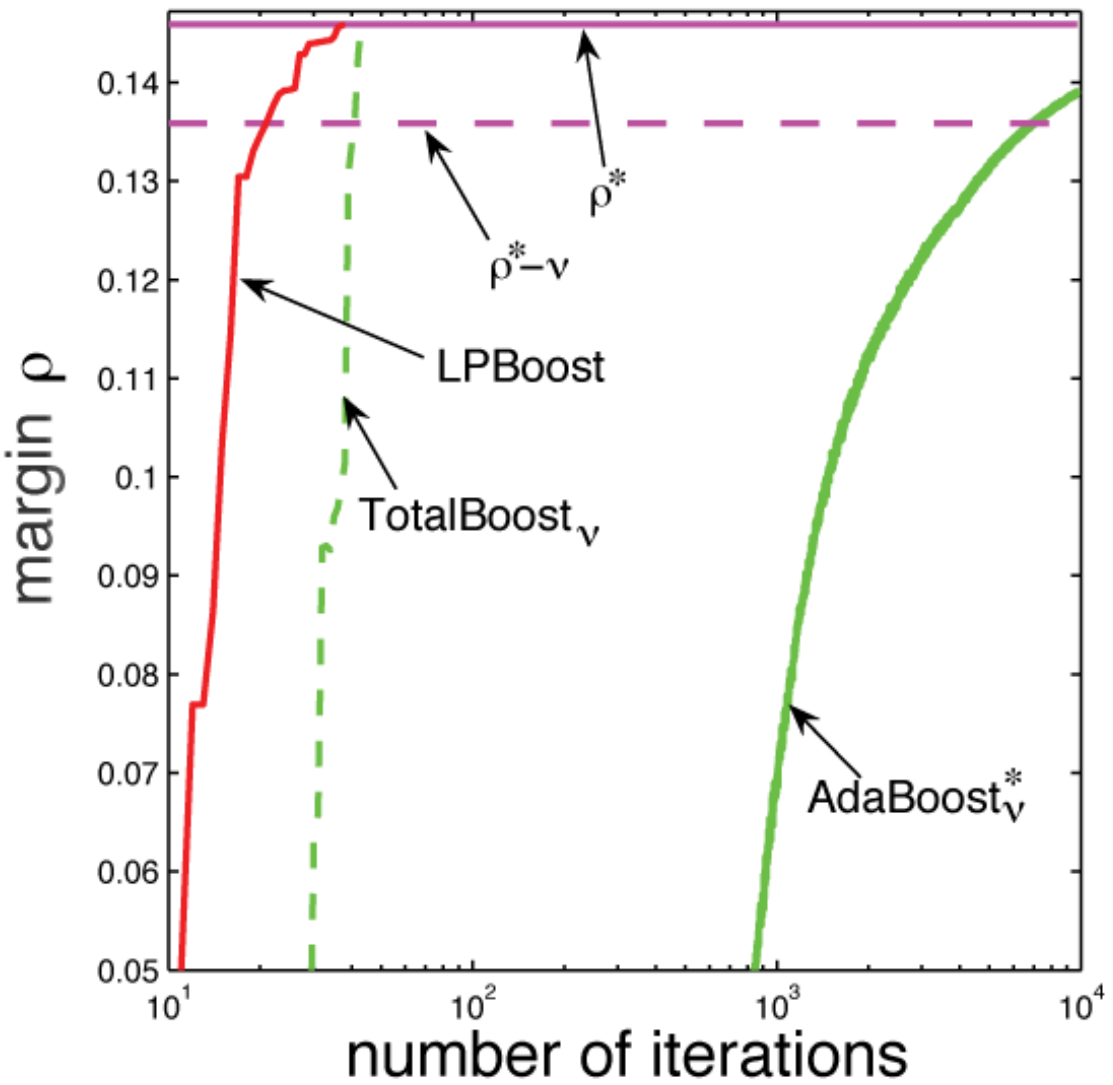
Cox-1 ($\delta = 0.01$)

- AdaBoost$^*_\delta$
- LPBoost
- TotalBoost$_\delta$

Results

- Corrective algorithms very slow
- LPBoost: $\mathcal{O}(N)$
- TotalBoost: several few iterations
- Totally unregularized: zero step size

Illustrative Experiments



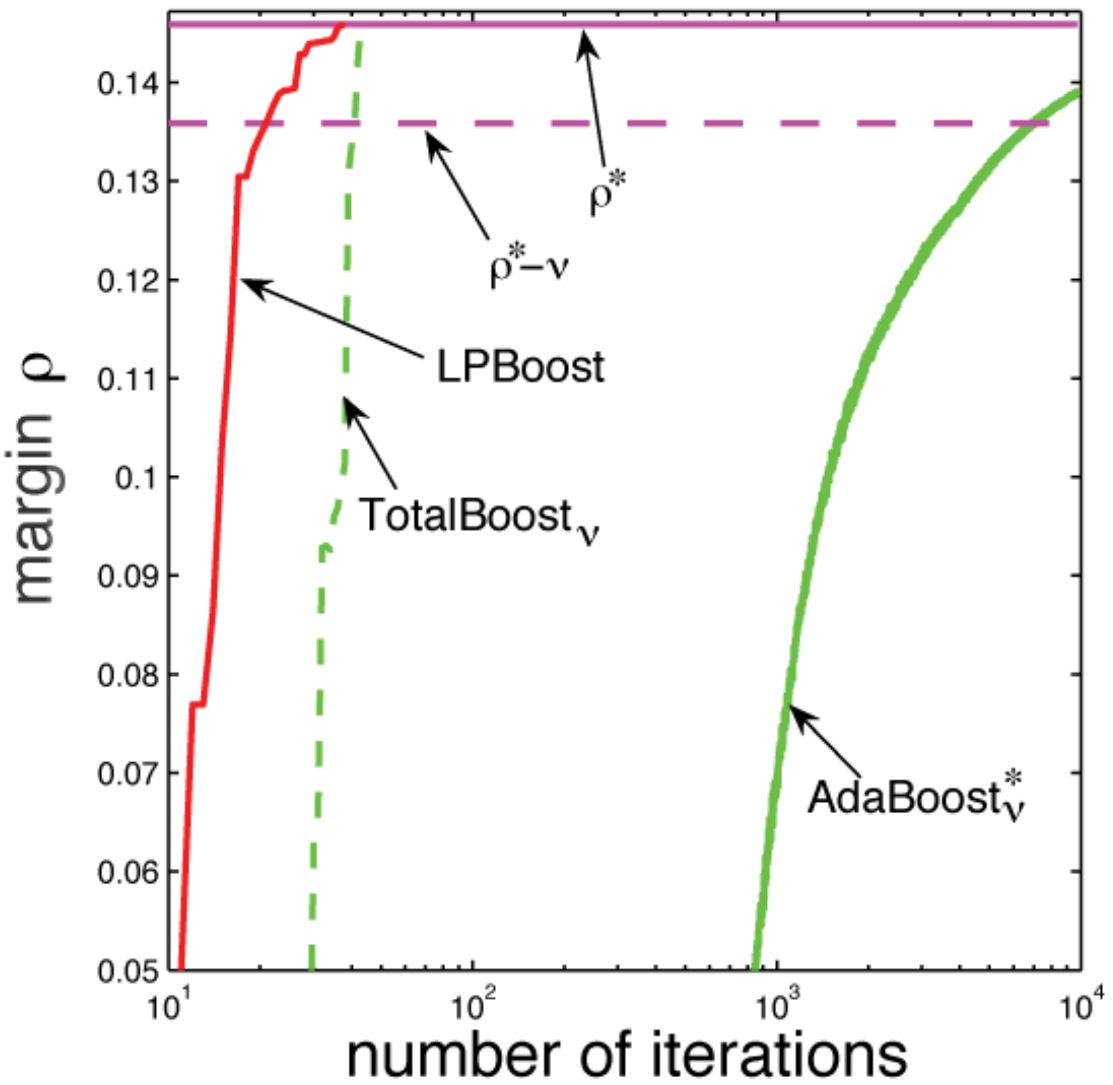
Cox-1 ($\delta = 0.01$)

- AdaBoost$^*_\delta$
- LPBoost
- TotalBoost$_\delta$

Results

- Corrective algorithms very slow
- LPBoost & TotalBoost$_\nu$ need few iterations
- AdaBoost$^*_\nu$ converges fast

Illustrative Experiments



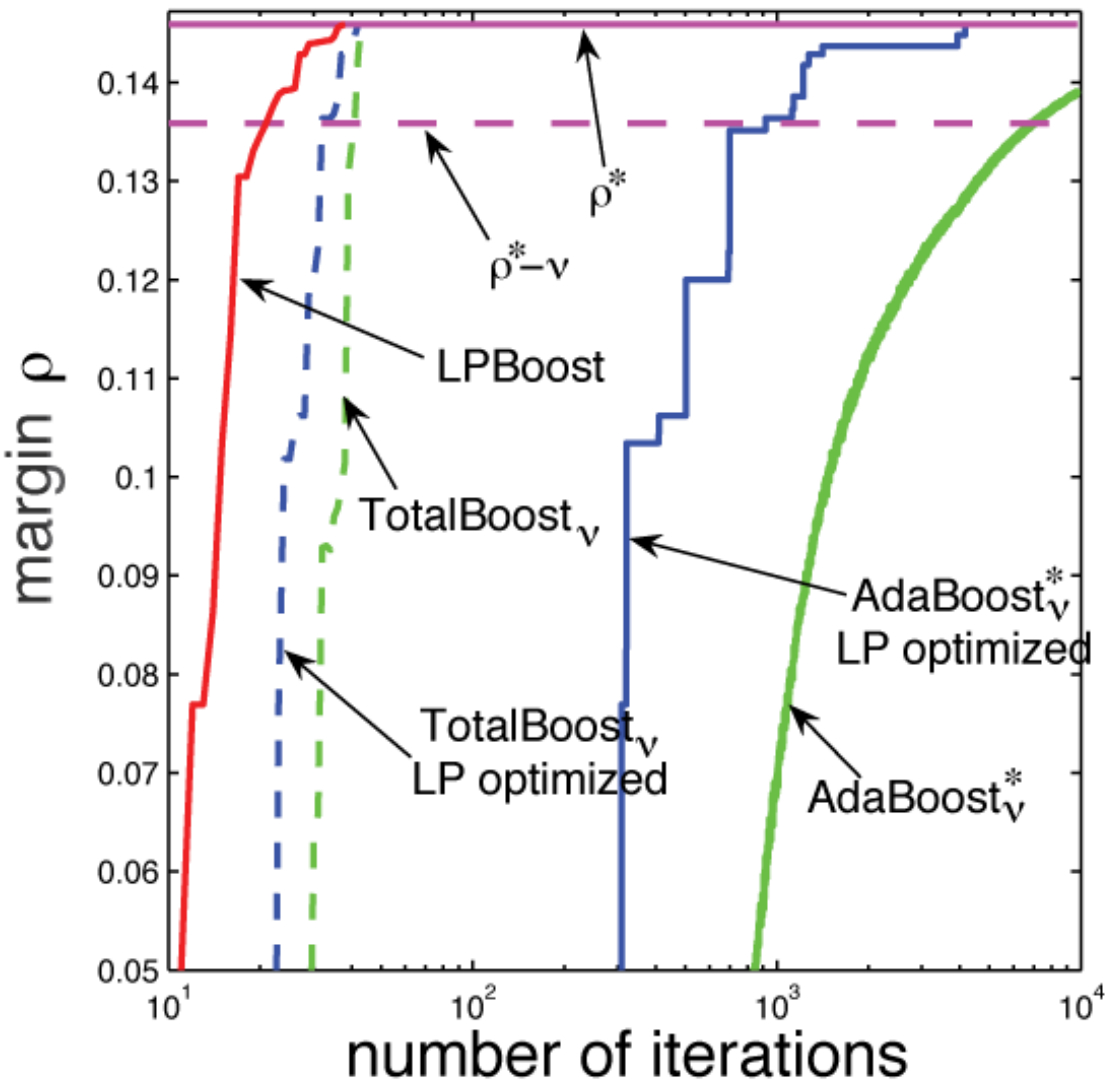
Cox-1 ($\delta = 0.01$)

- AdaBoost$^*_\delta$
- LPBoost
- TotalBoost$_\delta$

Results

- Corrective algorithms very slow
- LPBoost & TotalBoost$_\delta$ need few iterations
- Initial speed crucially depends on δ

Illustrative Experiments



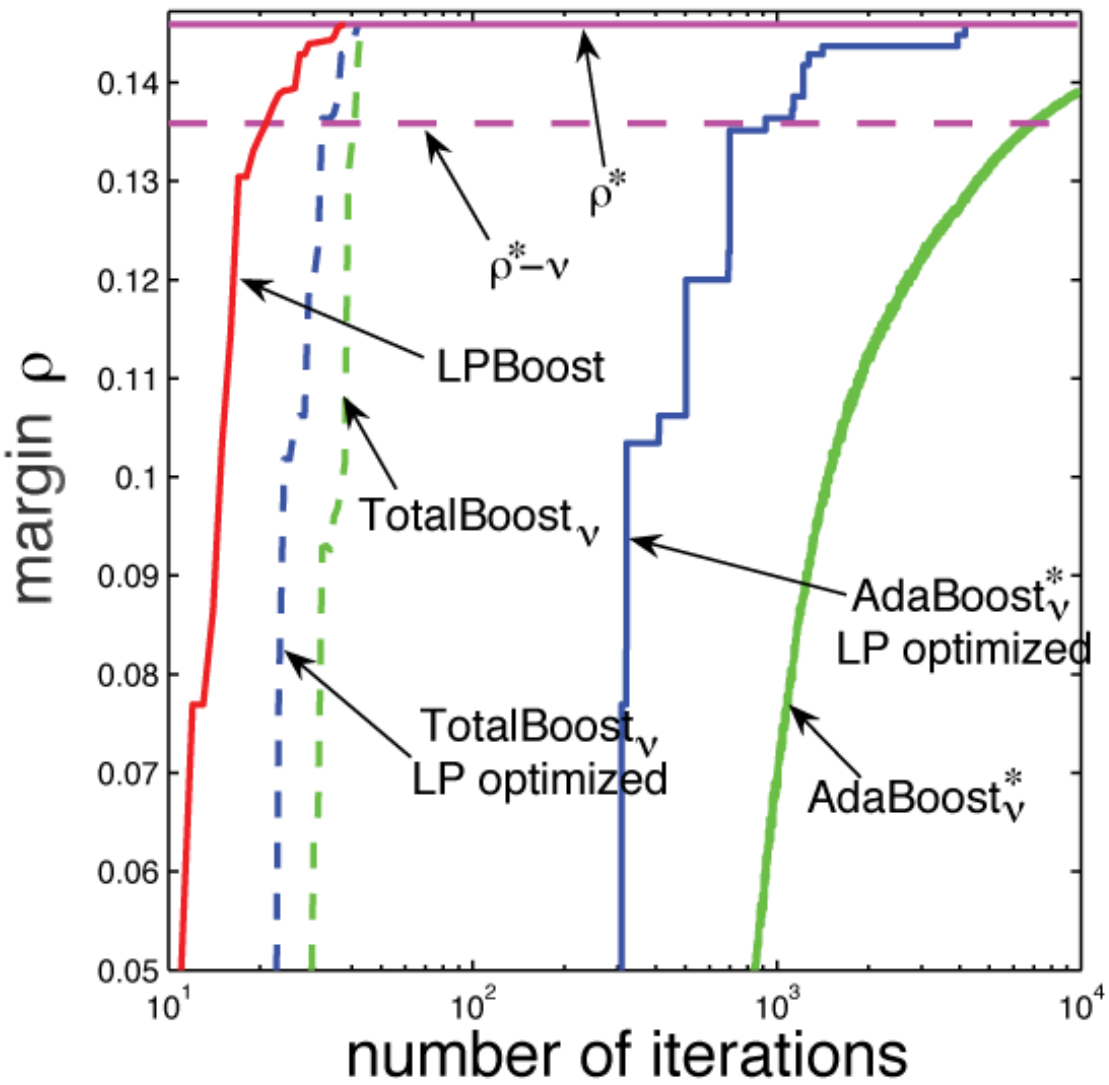
Cox-1 ($\delta = 0.01$)

- AdaBoost$^*_\delta$
- LPBoost
- TotalBoost$_\delta$

Results

- Corrective algorithms very slow
- LPBoost & TotalBoost$_\delta$ need few iterations
- Initial speed crucially depends on δ

Illustrative Experiments



Cox-1 ($\delta = 0.01$)

- AdaBoost$^*_\delta$
- LPBoost
- TotalBoost$_\delta$

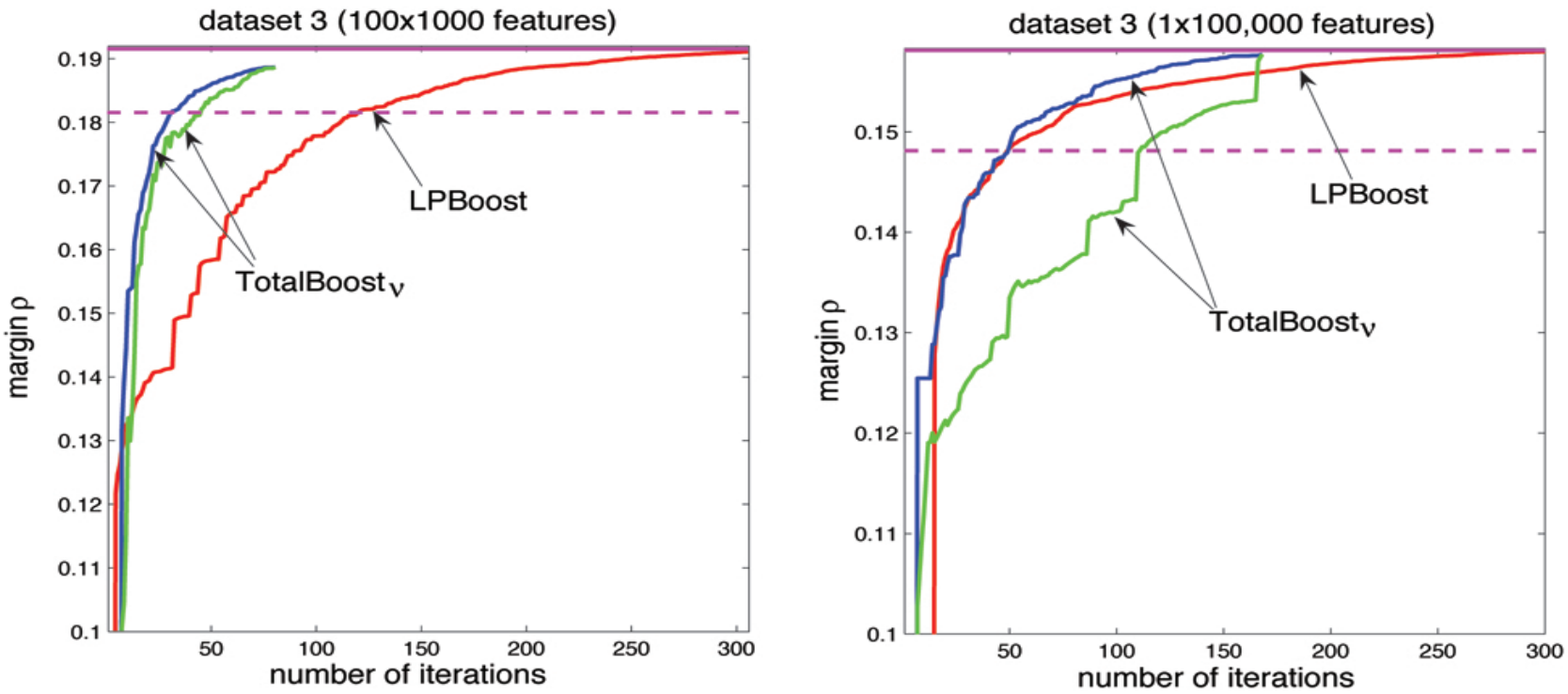
Results

- Corrective algorithms very slow
- LPBoost & TotalBoost$_\delta$ need few iterations
- Initial speed crucially depends on δ

LPBoost May Perform Much Worse Than TotalBoost

- Identified cases where LPBoost converges considerably slower than TotalBoost$_\delta$
- Dataset is a series of artificial datasets of 1000 examples with varying number of features created as follows:
  - First generated N_1 random ± 1-valued features $x_1, \ldots, x_{N_1}$ and set the label of the examples as $y = \mathrm{sign}(x_1 + x_2 + x_3 + x_4 + x_5)$
  - Then duplicated each features N_2 times, perturbed the features by Gaussian noise with $\sigma = 0.1$, and clipped the feature values so that they lie in the interval [-1,1]
  - Considered different N_1, N_2, the total number of features is $N_2 \times N_1$

LPBoost performs worse for high dimensional data with many redundant features



LPBoost vs. TotalBoost$_\delta$ on two 100,000 dimensional datasets: [*left*] many redundant features ($N_1 = 1,000, N_2 = 100$) and [*right*] independent features ($N_1 = 100,000, N_2 = 1$). Show margin vs. number of iterations

Do These Algorithms Work Better in Practice?

Do they generalize better?

- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result ...
   - ... but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

Do These Algorithms Work Better in Practice?

Do they generalize better?

- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result ...
   - ... but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results
   - New: Can prove fast convergence with soft margin (NIPS'07)

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

Do These Algorithms Work Better in Practice?

Do they generalize better?

- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result ...
   - ... but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results
   - **New:** Can prove fast convergence with soft margin (NIPS'07)

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

Do These Algorithms Work Better in Practice?

Do they generalize better?

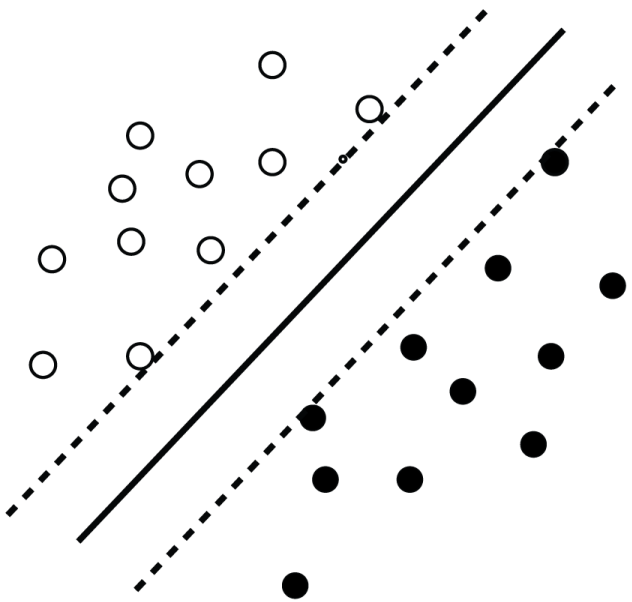
- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result ...
   - ... but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results
   - **New:** Can prove fast convergence with soft margin (NIPS'07)

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

SoftBoost = TotalBoost with ν-Trick



Hard-margin separation

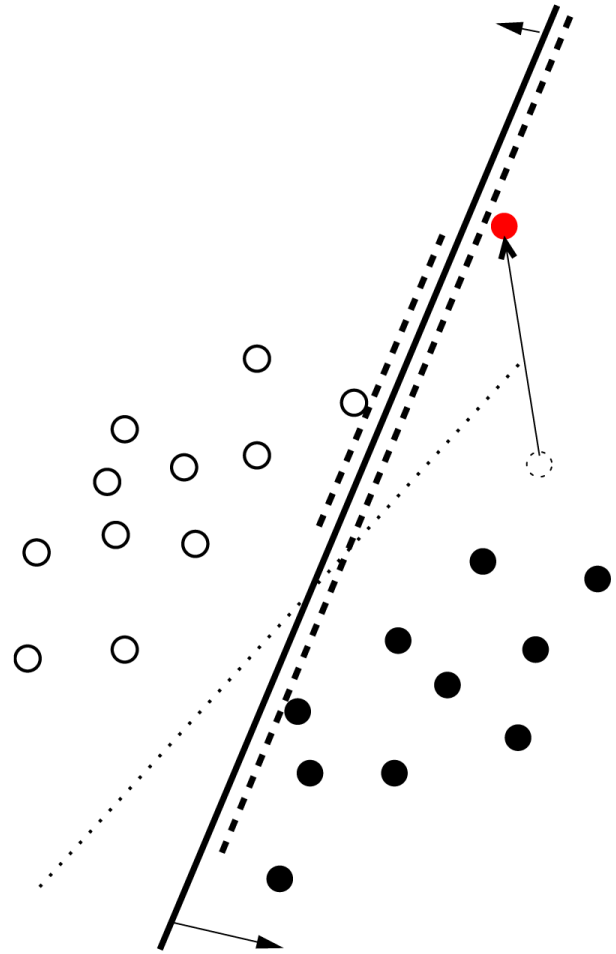
$$\max_{\rho,\boldsymbol{\xi},\boldsymbol{\alpha}} \quad \rho - \frac{1}{\nu}\sum_{n=1}^{N}\xi_n$$

$$\text{s.t.} \quad \rho \geq 0, \boldsymbol{\xi} \in \mathbb{R}_+, \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{H}|}$$

$$\text{with} \quad y_n \sum_{j=1}^{|\mathcal{H}|} \alpha_j h_j(\mathbf{x}_n) \geq \rho - \xi_n$$

$$\text{for } n = 1, \ldots, N.$$

SoftBoost = TotalBoost with ν-Trick



Hard-margin separation with outlier

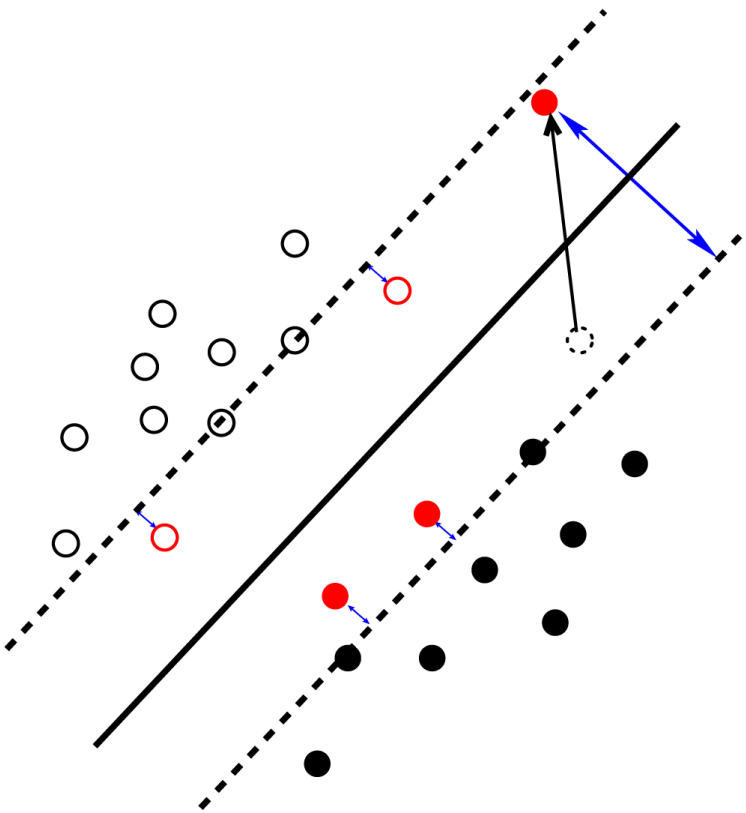
$$\max_{\rho,\boldsymbol{\xi},\boldsymbol{\alpha}} \quad \rho - \frac{1}{\nu}\sum_{n=1}^{N}\xi_n$$

$$\text{s.t.} \quad \rho \geq 0, \boldsymbol{\xi} \in \mathbb{R}_+, \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{H}|}$$

$$\text{with} \quad y_n \sum_{j=1}^{|\mathcal{H}|} \alpha_j h_j(\mathbf{x}_n) \geq \rho - \xi_n$$

$$\text{for } n = 1, \ldots, N.$$

SoftBoost = TotalBoost with ν-Trick



$$\max_{\rho,\boldsymbol{\xi},\boldsymbol{\alpha}} \quad \rho - \frac{1}{\nu}\sum_{n=1}^{N}\xi_n$$

$$\text{s.t.} \quad \rho \geq 0, \boldsymbol{\xi} \in \mathbb{R}_+, \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{H}|}$$

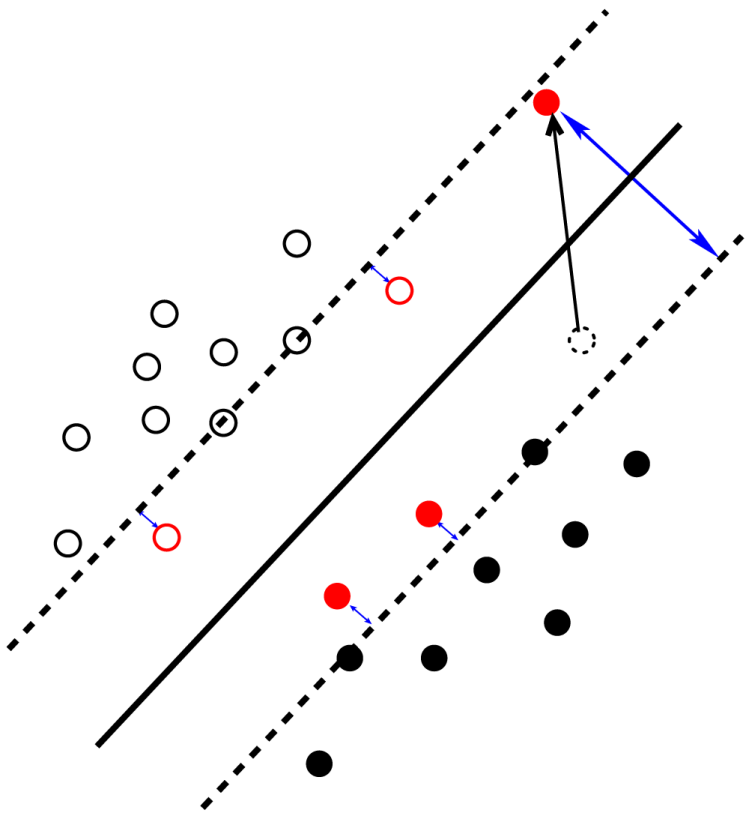
$$\text{with} \quad y_n \sum_{j=1}^{|\mathcal{H}|} \alpha_j h_j(\mathbf{x}_n) \geq \rho - \xi_n$$

$$\text{for } n = 1, \ldots, N.$$

Soft-margin separation with outlier

Chose margin such that ν examples are within the margin area

SoftBoost = TotalBoost with ν-Trick



$$\begin{aligned} \max_{\rho,\boldsymbol{\xi},\boldsymbol{\alpha}} \quad & \rho - \frac{1}{\nu}\sum_{n=1}^{N}\xi_n \\ \text{s.t.} \quad & \rho \geq 0, \boldsymbol{\xi} \in \mathbb{R}_+, \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{H}|} \\ \text{with} \quad & y_n \sum_{j=1}^{|\mathcal{H}|} \alpha_j h_j(\mathbf{x}_n) \geq \rho - \xi_n \\ & \text{for } n = 1, \ldots, N. \end{aligned}$$

Soft-margin separation with outlier

Chose margin such that ν examples are within the margin area

Softboost$_{\delta,\nu}$

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, desired accuracy δ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$ and let $u_i^t = y_i h_t(\mathbf{x}_i)$
   2. Calculate the edge γ_t of h_t: $\gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   3. Set $\widehat{\gamma}_t = (\min_{q=1,\ldots,t} \gamma_q) - \delta$ and solve
      $$\mathbf{d}^{t+1} = \operatorname*{argmin}_{\{\mathbf{d} \in \mathcal{P}^N \mid \mathbf{d} \leq 1/\nu; \mathbf{d}\cdot\mathbf{u}^q \leq \widehat{\gamma}_t, \text{ for } 1 \leq q \leq t\} = \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$$
   4. **If** above infeasible or $\mathbf{d}^{t+1}$ contains a zero then $T = t$ and break
4. **Output:** $f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

Softboost$_{\delta,\nu}$

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, desired accuracy δ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$ and let $u_i^t = y_i h_t(\mathbf{x}_i)$
   2. Calculate the edge γ_t of h_t: $\gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   3. Set $\widehat{\gamma}_t = (\min_{q=1,\ldots,t} \gamma_q) - \delta$ and solve
      $$\mathbf{d}^{t+1} = \operatorname*{argmin}_{\{\mathbf{d} \in \mathcal{P}^N \mid \mathbf{d} \leq 1/\nu; \mathbf{d} \cdot \mathbf{u}^q \leq \widehat{\gamma}_t, \text{ for } 1 \leq q \leq t\} = \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$$
   4. **If** above infeasible or $\mathbf{d}^{t+1}$ contains a zero then $T = t$ and break
4. **Output:** $f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

Softboost$_{\delta,\nu}$

1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, desired accuracy δ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto [-1, 1]$ and let $u_i^t = y_i h_t(\mathbf{x}_i)$
   2. Calculate the edge γ_t of h_t: $\gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   3. Set $\widehat{\gamma}_t = (\min_{q=1,\ldots,t} \gamma_q) - \delta$ and solve

Optimization Problem

$$\mathbf{d}^{t+1} = \operatorname*{argmin}_{\mathbf{d} \in \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$$

with $$\mathcal{C}_t := \{\mathbf{d} \in \mathcal{P}^N \mid \mathbf{d} \leq 1/\nu,\ \mathbf{d} \cdot \mathbf{u}^q \leq \widehat{\gamma}_t,\ \text{for } 1 \leq q \leq t\}$$

4. **Output:** $f_\alpha(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

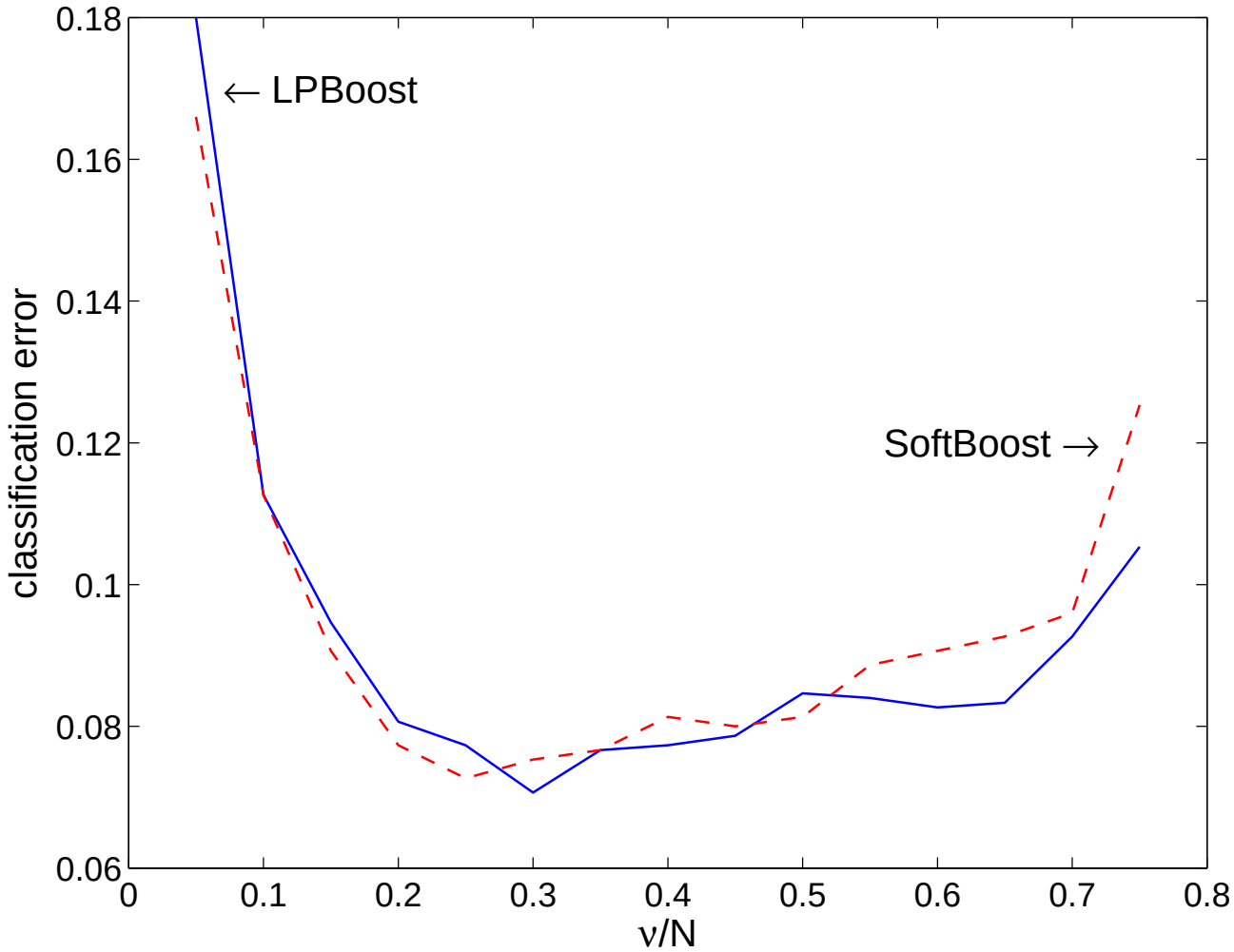
Iteration Bound

Theorem

Assume the base learner returns hypotheses with edge greater than g, then SoftBoost terminates after at most $\frac{2\ln(N/\nu)}{\delta^2}$ iteration with margin at least $g - \delta$.

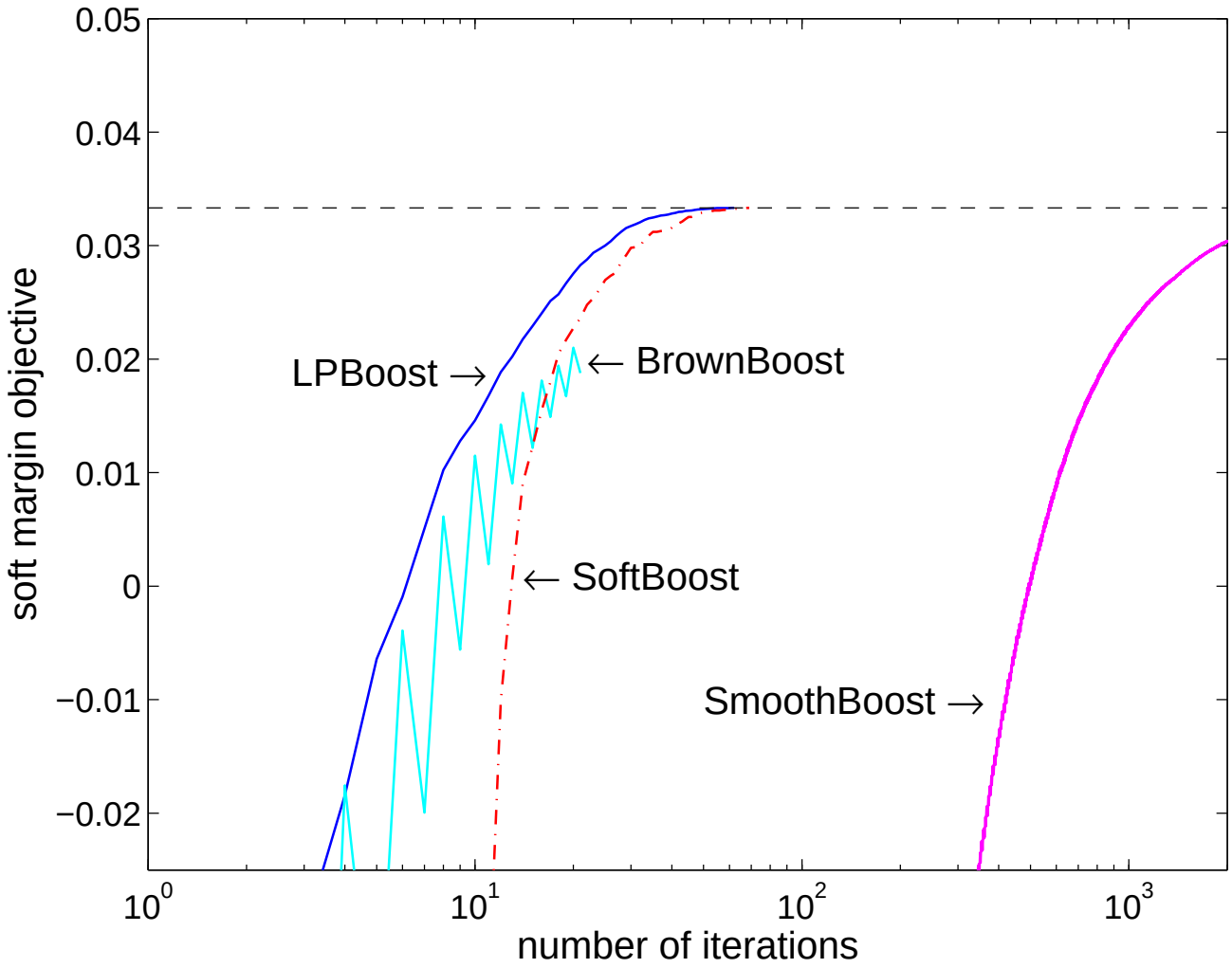
[Warmuth, Glocer & G.R., 2007]

LPBoost vs. SoftBoost



Generalization performance of SoftBoost (solid) and LPBoost (dotted) on a synthetic data set with 10% label-noise for different δ.

Convergence Speed of Different Algorithms



Soft margin objective vs. the number of iterations for LPBoost, SoftBoost, BrownBoost and SmoothBoost.

Generalization Errors on IDA Benchmarks

- RBF networks as base learner
- 5-fold cross-validation on 100 splits
- Average error $\pm$ standard deviation
- LPBoost very similar to SoftBoost (not shown)

	AdaBoost	BrownBoost	AdaBoost reg	SoftBoost
Banana	13.3 $\pm$ 0.7	12.9 $\pm$ 0.7	11.3 $\pm$ 0.6	**11.1 $\pm$ 0.5**
B.Cancer	32.1 $\pm$ 3.8	30.2 $\pm$ 3.9	**27.3 $\pm$ 4.3**	28.0 $\pm$ 4.5
Diabetes	27.9 $\pm$ 1.5	27.2 $\pm$ 1.6	24.5 $\pm$ 1.7	**24.4 $\pm$ 1.7**
German	26.9 $\pm$ 1.9	24.8 $\pm$ 1.9	25.0 $\pm$ 2.2	**24.7 $\pm$ 2.1**
Heart	20.1 $\pm$ 2.7	20.0 $\pm$ 2.8	**17.6 $\pm$ 3.0**	18.2 $\pm$ 2.7
Ringnorm	1.9 $\pm$ 0.3*	1.9 $\pm$ 0.2	**1.7 $\pm$ 0.2**	1.8 $\pm$ 0.2
F.Solar	36.1 $\pm$ 1.5	36.1 $\pm$ 1.4	**34.4 $\pm$ 1.7**	35.5 $\pm$ 1.4
Thyroid	**4.4 $\pm$ 1.9***	4.6 $\pm$ 2.1	4.9 $\pm$ 2.0	4.9 $\pm$ 1.9
Titanic	22.8 $\pm$ 1.0	22.8 $\pm$ 0.8	**22.7 $\pm$ 1.0**	23.0 $\pm$ 0.8
Waveform	10.5 $\pm$ 0.4	10.4 $\pm$ 0.4	10.4 $\pm$ 0.7	**9.8 $\pm$ 0.5**

Summarizing Boosting

- AdaBoost can be viewed as entropy projection
- TotalBoost projects based on all previous hypotheses
- Provably maximizes the margin
  - Theory: as fast as AdaBoost$^*_\delta$
  - Practice: much faster ($\approx$ LPBoost)
- Experiments corroborate our theory
  - Few iterations (good for feature selection)
  - LPBoost may have problems of maximizing the margin
- Soft margin extension with provable convergence
- First time to combine
  - Theoretical convergence guarantee
  - Empirical fast convergence
  - Good generalization performance on noisy data

The End

- Slides with references will be available at `http://www.fml.mpg.de/raetsch/lectures/`
- Collaborators for this work:
  Kristin Bennett, Ayhan Demiriz, Karen Glocer, Jun Liao, Sebastian Mika, Klaus-Robert Müller, Takashi Onoda, Bernhard Schölkopf, Sören Sonnenburg, Manfred Warmuth
- **Thank You!**


C. Cortes and V.N. Vapnik.
Support vector networks.
Machine Learning, 20:273–297, 1995.


A. Demiriz, K.P. Bennett, and J. Shawe-Taylor.
Linear programming boosting via column generation.
Machine Learning, 46:225–254, 2002.


Y. Freund and R.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.


R. Hettich and K.O. Kortanek.
Semi-infinite programming: Theory, methods and applications.
SIAM Review, 3:380–429, September 1993.


R. Meir and G. Rätsch.
An introduction to boosting and leveraging.

In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNCS, pages 119–184. Springer, 2003.


G. Rätsch.
Robust Boosting via Convex Optimization.
PhD thesis, University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany, October 2001.


G. Rätsch, A. Demiriz, and K. Bennett.
Sparse regression ensembles in infinite and finite hypothesis spaces.
Machine Learning, 48(1-3):193–221, 2002.


G. Rätsch and M.K. Warmuth.
Efficient margin maximization with boosting.
Journal of Machine Learning Research, 6:2131–2152, 2005.


M.K. Warmuth, J. Liao, and G. Rätsch.
Totally corrective boosting algorithms that maximize the margin.

In W. Cohen and A. Moore, editors, *Proc. ICML'06*, pages 1001–1008. ACM Press, 2006.