Totally Corrective Boosting Algorithms that Maximize the Margin

Manfred K. Warmuth\textsuperscript{1}
Jun Liao\textsuperscript{1}
Gunnar Rätsch\textsuperscript{2}

\textsuperscript{1}University of California, Santa Cruz
\textsuperscript{2}Friedrich Miescher Laboratory, Tübingen, Germany

Last update: March 2, 2007
1 Basics

2 Large margins

3 Games

4 Convergence

5 Illustrative Experiments
Outline

1 Basics
2 Large margins
3 Games
4 Convergence
5 Illustrative Experiments
Protocol of Boosting

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

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

[Freund & Schapire, 1995]
Protocol of Boosting

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

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

[Freund & Schapire, 1995]
Protocol of Boosting

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

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

[Freund & Schapire, 1995]
Boosting: 1st Iteration

First hypothesis:
- **Error rate:** \( \frac{2}{11} \)

\[
\epsilon_t = \sum_{n=1}^{N} d_t^n \mathbb{1}(h_t(x_n) = y_n)
\]

- **Edge:** \( \frac{9}{22} \)

\[
\gamma_t = \sum_{n=1}^{N} d_t^n y_n h_t(x_n) = 1 - 2\epsilon_t
\]
Boosting: 1st Iteration

First hypothesis:
- Error rate: \( \frac{2}{11} \)

\[
\epsilon_t = \sum_{n=1}^{N} d^t_n \mathbb{I}(h_t(x_n) = y_n)
\]

- Edge: \( \frac{9}{22} \)

\[
\gamma_t = \sum_{n=1}^{N} d^t_n y_n h_t(x_n) = 1 - 2\epsilon_t
\]
Update Distribution

Misclassified examples ⇒ Increased weights

After update:
- Error rate:
  \[ \epsilon(h_t, d_{t+1}) = \frac{1}{2} \]
- Edge:
  \[ \gamma(h_t, d_{t+1}) = 0 \]
Update Distribution

Misclassified examples $\Rightarrow$ Increased weights

After update:
- Error rate:
  $\epsilon(h_t, d^{t+1}) = \frac{1}{2}$
- Edge:
  $\gamma(h_t, d^{t+1}) = 0$
Ada-Boost as Entropy Projection

Minimize relative entropy to last distribution subject to constraint

\[
\min_{d} \quad \Delta(d, d^t) \\
\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_t(x_n) = 0 \\
d \in \mathcal{P}^N
\]

where

- \( \Delta(d, d^t) = \sum_{n=1}^{N} d_n \ln \frac{d_n}{d_n^t} \) and
- \( \mathcal{P}^N \) is the \( N \) dimensional probability simplex

[Lafferty, 1999; Kivinen & Warmuth, 1999]
Before 2nd Iteration
Boosting: 2nd Hypothesis

AdaBoost assumption:
Edge $\gamma > \nu$
Update Distribution

Edge $\gamma = 0$

AdaBoost update sets edge of last hypothesis to 0

Number of iterations:

$$\leq \frac{2 \ln N}{\nu^2}$$
Which constraints?

Corrective - Ada-Boost: Single constraint

\[
\begin{align*}
\min_{d \in \mathcal{P}^N} & \quad \Delta(d, d^t) \\
\text{s.t.} & \quad \sum_{n=1}^N d_n y_n h_t(x_n) = 0 \quad \text{for } n = 1, \ldots, t
\end{align*}
\]

Totally corrective: One constraint per past weak hypothesis

\[
\begin{align*}
\min_{d \in \mathcal{P}^N} & \quad \Delta(d, d^1) \\
\text{s.t.} & \quad \sum_{n=1}^N d_n y_n h_q(x_n) = 0 \quad \text{for } q = 1, \ldots, t
\end{align*}
\]

[Lafferty, 1999; Kivinen & Warmuth, 1999]
Which constraints?

Corrective - Ada-Boost: Single constraint

\[
\min_{d \in \mathcal{P}^N} \Delta(d, d^t)
\]
\[
\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_t(x_n) = 0 \quad \text{for } n = 1, \ldots, t
\]

Totally corrective: One constraint per past weak hypothesis

\[
\min_{d \in \mathcal{P}^N} \Delta(d, d^1)
\]
\[
\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_q(x_n) = 0 \quad \text{for } q = 1, \ldots, t
\]

[Lafferty, 1999; Kivinen & Warmuth, 1999]
Boosting: 3rd Hypothesis
Boosting: 4th Hypothesis
## All Hypotheses

<table>
<thead>
<tr>
<th>leicht</th>
<th>nicht rot</th>
<th>schwer</th>
<th>sehr rot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- leicht: leicht
- nicht rot: nicht rot
- schwer: schwer
- sehr rot: sehr rot
Decision: \( f_\alpha(x) = \sum_{t=1}^{T} \alpha_t h_t(x) > 0 \)
Outline

1 Basics
2 Large margins
3 Games
4 Convergence
5 Illustrative Experiments
Large margins in addition to correct classification

**Margin** of the combined hypothesis $f_\alpha$ for example $(x_n, y_n)$

$$
\rho_n(\alpha) = y_n f_\alpha(x_n)
= y_n \sum_{t=1}^{T} \alpha_t h_t(x_n) \quad (\alpha \in \mathcal{P}^T)
$$

Margin of set of examples is minimum over examples

$$
\rho(\alpha) := \min_n \rho_n(\alpha)
$$

[Freund, Schapire, Bartlett & Lee, 1998]
Large Margin and Linear Separation

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

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

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

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

**Margin vs. edge**

**Margin**
- Measure for “confidence” in prediction for a hypothesis weighting
- **Margin of example** $n$ for current hypothesis weighting $\alpha$

$$
\rho_n(\alpha) = y_n f_\alpha(x_n) = y_n \sum_{t=1}^{T} \alpha_t h_t(x_n) \quad \alpha \in P^T
$$

**Edge**
- Measurement of “goodness” of a hypothesis w.r.t. a distribution
- **Edge of a hypothesis** $h$ for a distribution $d$ on the examples

$$
\gamma_h(d) = \sum_{n=1}^{N} d_n y_n h(x_n) \quad d \in P^N
$$

What is the connection? [Breiman, 1999]
## Margin vs. edge

### Margin
- Measure for “confidence” in prediction for a hypothesis weighting
- Margin of example $n$ for current hypothesis weighting $\alpha$

$$\rho_n(\alpha) = y_n f_{\alpha}(x_n) = y_n \sum_{t=1}^{T} \alpha_t h_t(x_n) \quad \alpha \in \mathcal{P}^T$$

### Edge
- Measurement of “goodness” of a hypothesis w.r.t. a distribution
- Edge of a hypothesis $h$ for a distribution $d$ on the examples

$$\gamma_h(d) = \sum_{n=1}^{N} d_n y_n h(x_n) \quad d \in \mathcal{P}^N$$

---

**What is the connection?**
Margin vs. edge

**Margin**
- Measure for “confidence” in prediction for a hypothesis weighting
- Margin of example $n$ for current hypothesis weighting $\alpha$

$$\rho_n(\alpha) = y_n f_\alpha(x_n) = y_n \sum_{t=1}^{T} \alpha_t h_t(x_n) \quad \alpha \in \mathcal{P}^T$$

**Edge**
- Measurement of “goodness” of a hypothesis w.r.t. a distribution
- Edge of a hypothesis $h$ for a distribution $d$ on the examples

$$\gamma_h(d) = \sum_{n=1}^{N} d_n y_n h(x_n) \quad d \in \mathcal{P}^N$$

What is the connection? [Breiman, 1999]
von Neumann’s Minimax-Theorem

Set of examples $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ and hypotheses set $\mathcal{H}^t = \{h_1, \ldots, h_t\}$,

\[
\begin{align*}
\text{minimum edge:} & \quad \gamma^*_t = \min_{d \in \mathcal{P}^N} \max_{h \in \mathcal{H}^t} \gamma_h(d) \\
\text{maximum margin:} & \quad \rho^*_t = \max_{\alpha \in \mathcal{P}^t} \min_n y_n f_\alpha(x_n)
\end{align*}
\]

Duality: $\gamma^*_t = \rho^*_t$

[von Neumann, 1928]
von Neumann’s Minimax-Theorem

Set of examples $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ and hypotheses set $\mathcal{H}^t = \{h_1, \ldots, h_t\}$,

minimum edge: $\gamma^*_t = \min_{d \in \mathcal{P}^N} \max_{q=1}^t \sum_{n=1}^N d_n y_n h_q(x_n)$

maximum margin: $\rho^*_t = \max_{\alpha \in \mathcal{P}^t} \min_{n} \sum_{q=1}^t \alpha_t h_q(x_n)$

Duality: $\gamma^*_t = \rho^*_t$
Outline

1. Basics
2. Large margins
3. Games
4. Convergence
5. Illustrative Experiments
# Two-player Zero Sum Game

**Rock, Paper, Scissors game**

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>$d_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$d_2$</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$d_3$</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

**Row player**

Single row is pure strategy of **row player** and $d$ is mixed strategy

**Column player**

Single column is pure strategy of **column player** and $\alpha$ is mixed strategy

$$\text{payoff} = d^T M \alpha = \sum_{i,j} d_i \alpha_i^{j}$$

[Freund and Schapire, 1997]
Optimum Strategy

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>α₁</td>
<td>α₂</td>
<td>α₃</td>
</tr>
<tr>
<td>P</td>
<td>.33</td>
<td>.33</td>
<td>.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>d₁</td>
<td>.33</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d₂</td>
<td>.33</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>d₃</td>
<td>.33</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

- **Min-max theorem:**

\[
\min_{d} \max_{\alpha} d^T M\alpha = \min_{d} \max_{j} d^T M e_j = \max_{\alpha} \min_{d} d^T M\alpha = \max_{\alpha} \min_{i} e_i M\alpha
\]

\[
= \text{value of the game (0 in example )}
\]

eₐ is pure strategy
Connection to Boosting?

- Rows are the examples
- Columns the weak hypothesis
- $M_{i,j} = h_j(x_i)y_i$
- Row sum: margin of example
- Column sum: edge of weak hypothesis
- Value of game: $\gamma^* = \rho^*$
New column added: boosting

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
<th>α₁</th>
<th>α₂</th>
<th>α₃</th>
<th>α₄</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>.22</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>.33</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>.44</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>.11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Value of game **increases** from 0 to .11
Row added: on-line learning

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>α₁</td>
<td>.33</td>
<td>.44</td>
<td>.22</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>d₁</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>d₂</td>
<td>.22</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>d₃</td>
<td>.44</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>d₄</td>
<td>.33</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>edge</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

Value of game **decreases** from 0 to -.11
Boosting: maximize margin incrementally

<table>
<thead>
<tr>
<th></th>
<th>$\alpha_1^1$</th>
<th>$\alpha_1^2$</th>
<th>$\alpha_2^2$</th>
<th>$\alpha_1^3$</th>
<th>$\alpha_2^3$</th>
<th>$\alpha_3^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1^1$</td>
<td>0</td>
<td>$d_1^2$</td>
<td>0</td>
<td>-1</td>
<td>$d_1^3$</td>
<td>0</td>
</tr>
<tr>
<td>$d_1^2$</td>
<td>1</td>
<td>$d_2^2$</td>
<td>1</td>
<td>0</td>
<td>$d_2^3$</td>
<td>1</td>
</tr>
<tr>
<td>$d_1^3$</td>
<td>-1</td>
<td>$d_3^2$</td>
<td>-1</td>
<td>1</td>
<td>$d_3^3$</td>
<td>-1</td>
</tr>
</tbody>
</table>

iteration 1  iteration 2  iteration 3

- In each iteration solve optimization problem to update $d$
- Column player adds new column - weak hypothesis
- Some assumptions will be needed on the edge of the added hypothesis
Games

Value non-decreasing

$\gamma^*, \rho^*$: edge/margin for all hypotheses
Duality gap

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

$$\gamma(d) \geq \gamma^*_t = \rho^*_t \geq \rho(\alpha)$$
How Large is the Maximal Margin?

Assumptions on Weak learner

For any distribution $d$ on the examples, the weak learner returns a hypothesis $h$ with edge $\gamma_h(d)$ 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 $d$ on the examples, the weak learner returns a hypothesis $h$ with edge $\gamma_h(d)$ at least $g$.

Best case: $g = \rho^* = \gamma^*$

## Implication from Minimax Theorem

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

---

[Breiman, 1999; Bennett et al.; G.R. et al., 2001; Rudin et al., 2004]
Assumptions on Weak learner

For any distribution $d$ on the examples, the weak learner returns a hypothesis $h$ with edge $\gamma_h(d)$ at least $g$.

Best case: $g = \rho^* = \gamma^*$

Implication from Minimax Theorem

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

Idea to iteratively solve LP: LPBoost

Add "best" hypothesis $h = \arg\max \gamma_h(d^t)$ to $\mathcal{H}^t$ and resolve

$$d^{t+1} = \arg\min_{d \in \mathcal{P}^N} \max_{h \in \mathcal{H}^t} \gamma_h(d)$$

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

**Convergence?**

<table>
<thead>
<tr>
<th>LPBoost?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No iteration bounds known</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AdaBoost?</th>
</tr>
</thead>
<tbody>
<tr>
<td>May “oscillate”</td>
</tr>
<tr>
<td>Does not find maximizing $\alpha$ (counter examples)</td>
</tr>
<tr>
<td>But there are some guarantees:</td>
</tr>
<tr>
<td>$\rho(\alpha^t) \geq 0$ after $2 \ln N/g^2$ iterations</td>
</tr>
<tr>
<td>$\rho(\alpha^t) \geq g/2$ in the limit</td>
</tr>
</tbody>
</table>

[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 convergence rates
- **LPBoost** uses a Linear Programming solver
  - Often very fast in practice, but **no converge rates**
- **AdaBoost** requires \( \frac{2 \log(N)}{\nu^2} \) iterations to get \( \rho^t \in [\rho^* - \nu, \rho^*] \)
  - Slow in practice, i.e. not faster than theory predicts
- **TotalBoost** requires \( \frac{2 \log(N)}{\nu^2} \) iterations to get \( \rho^t \in [\rho^* - \nu, \rho^*] \)
  - Fast in practice
  - Combination of benefits

[G.R. & Warmuth, 2004; Warmuth et al., 2006]
## Modify AdaBoost for maximizing margin

- **Arc-GV** asymptotically maximizes the margin
  - quite slow, no convergence rates
- **LPBoost** uses a Linear Programming solver
  - Often very fast in practice, but no convergence rates
- **AdaBoost**\(_\nu\) requires \(\frac{2\log(N)}{\nu^2}\) iterations to get \(\rho^t \in [\rho^* - \nu, \rho^*]\)
  - Slow in practice, i.e. not faster than theory predicts
- **TotalBoost**\(_\nu\) requires \(\frac{2\log(N)}{\nu^2}\) iterations to get \(\rho^t \in [\rho^* - \nu, \rho^*]\)
  - Fast in practice
  - Combination of benefits

---

[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 convergence rates
- **LPBoost** uses a Linear Programming solver
  - Often very fast in practice, but **no converge rates**
- **AdaBoost** requires $\frac{2 \log(N)}{\nu^2}$ iterations to get $\rho^t \in [\rho^* - \nu, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- **TotalBoost** requires $\frac{2 \log(N)}{\nu^2}$ iterations to get $\rho^t \in [\rho^* - \nu, \rho^*]$
  - Fast in practice
  - Combination of benefits

[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 convergence rates
- **LPBoost** uses a Linear Programming solver
  - Often very fast in practice, but **no convergence rates**
- **AdaBoost** \( \nu \) requires \( \frac{2 \log(N)}{\nu^2} \) iterations to get \( \rho_t \in [\rho^* - \nu, \rho^*] \)
  - Slow in practice, i.e. not faster than theory predicts
- **TotalBoost** \( \nu \) requires \( \frac{2 \log(N)}{\nu^2} \) iterations to get \( \rho_t \in [\rho^* - \nu, \rho^*] \)
  - Fast in practice
  - Combination of benefits

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

1 Basics
2 Large margins
3 Games
4 Convergence
5 Illustrative Experiments
Want margin $\geq g - \nu$
Want margin $\geq g - \nu$

- Assumption: $\gamma_t \geq g$
- Estimate of target: $\hat{\gamma}_t = (\min_{q=1}^t \gamma_q) - \nu$
Idea: Projections to $\hat{\gamma}_t$ instead of 0

**Corrective:** Single constraint

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

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

**Totally corrective:** One constraint per past weak hypothesis

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

$$\text{s.t.} \quad \sum_{n=1}^{N} d_n y_n h_q(x_n) \leq \hat{\gamma}_t \quad \text{for } q = 1, \ldots, t$$
**TotalBoost\(\nu\)**

1. **Input:** \(S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle\), desired accuracy \(\nu\)
2. **Initialize:** \(d_n^1 = 1/N\) for all \(n = 1 \ldots N\)
3. **Do for** \(t = 1, \ldots\)
   1. Train classifier on \(\{S, d^t\}\) and obtain hypothesis \(h_t: x \mapsto [-1, 1]\) and let \(u^t_i = y_i h_t(x_i)\)
   2. Calculate the edge \(\gamma_t\) of \(h_t\): \(\gamma_t = d^t \cdot u^t\)
   3. Set \(\hat{\gamma}_t = (\min_{q=1}^t \gamma_q) - \nu\) and solve
      \[
      d^{t+1} = \arg\min_{d \in \mathcal{P}^N \mid d \cdot u^q \leq \hat{\gamma}_t, \text{ for } 1 \leq q \leq t} \Delta(d, d^1) \\
      = C_t
      \]
   4. **If** above infeasible or \(d^{t+1}\) contains a zero
      **then** \(T = t\) and break
4. **Output:** \(f_\alpha(x) = \sum_{t=1}^T \alpha_t h_t(x)\), where the coefficients \(\alpha_t\)
   maximize margin over hypotheses set \(\{h_1, \ldots, h_T\}\).
TotalBoost$_\nu$

1. **Input:** $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$, desired accuracy $\nu$
2. **Initialize:** $d^1_n = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, d^t\}$ and obtain hypothesis $h_t: x \mapsto [-1, 1]$ and let $u^t_i = y_i h_t(x_i)$
   2. Calculate the edge $\gamma_t$ of $h_t$: $\gamma_t = d^t \cdot u^t$
   3. Set $\hat{\gamma}_t = (\min_{q=1}^{t} \gamma_q) - \nu$ and solve
      $$d^{t+1} = \arg\min_{d \in P^N} \Delta(d, d^1)$$
      $$\{d \in P^N | d \cdot u^q \leq \hat{\gamma}_t, \text{ for } 1 \leq q \leq t \} = C_t$$
   4. If above infeasible or $d^{t+1}$ contains a zero
      then $T = t$ and break

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

Optimization Problem

$$d^{t+1} = \arg\min_{d \in C_t} \Delta(d, d^1)$$

with $C_t$: $$\{d \in P^N | d \cdot u^q \leq \hat{\gamma}_t, \text{ for } 1 \leq q \leq t \} = C_t$$
TotalBoost$_{\nu}$

1. **Input:** $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$, desired accuracy $\nu$

2. **Initialize:** $d^1_n = 1/N$ for all $n = 1 \ldots N$

3. **Do for** $t = 1, \ldots$
   1. Train classifier on $\{S, d^t\}$ and obtain hypothesis $h_t : x \mapsto [-1, 1]$ and let $u^t_i = y_i h_t(x_i)$
   2. Calculate the edge $\gamma_t$ of $h_t$: $\gamma_t = d^t \cdot u^t$
   3. Set $\hat{\gamma}_t = (\min_{q=1}^{t} \gamma_q) - \nu$ and solve

   **Optimization Problem**

   $$d^{t+1} = \arg\min_{d \in C_t} \Delta(d, d^1)$$

   with $C_t := \{d \in \mathcal{P}^N | d \cdot u^q \leq \hat{\gamma}_t, \text{ for } 1 \leq q \leq t\}$

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

- High weight given to **hard** examples

\[ d_{n}^{t+1} \sim \exp(-y_{n}f_{\alpha_{t+1}}(x_{n})) \]

- Softmin of margins
- \( \alpha_{t+1} \) are current coefficients of hypotheses
Lower Bounding the Progress

Theorem

For $d^t, d^{t+1} \in \mathcal{P}^N$ and $u^t \in [-1, 1]^N$,

if $\Delta(d^{t+1}, d^t)$ finite and $d^{t+1} \cdot u^t \neq d^t \cdot u^t$ then

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

- $d^t \cdot u^t = \gamma_t$
- $d^{t+1} \cdot u^t \leq \tilde{\gamma}_t \leq \gamma_t - \nu$
- Thus $d^t \cdot u^t - d^{t+1} \cdot u^t \geq \nu$ and $\Delta(d^{t+1}, d^t) > \frac{\nu^2}{2}$
Generalized Pythagorean Theorem

\[ C_t = \{ \mathbf{d} \in \mathcal{P}^N | \mathbf{d} \cdot \mathbf{u}^q \leq \hat{\gamma}_t, 1 \leq q \leq t \}, \quad C_0 = \mathcal{P}^N, \quad C_t \subseteq C_{t-1} \]

\( \mathbf{d}^t \) is projection of \( \mathbf{d}^1 \) onto \( C_{t-1} \) at iteration \( t - 1 \)

\[ \mathbf{d}^t = \arg\min_{\mathbf{d} \in C_{t-1}} \Delta(\mathbf{d}, \mathbf{d}^1) \]

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

[Herbster & Warmuth, 2001]
Convergence

Sketch of Proof

1: \( \Delta(d^2, d^1) - \Delta(d^1, d^1) \geq \Delta(d^2, d^1) > \frac{\nu^2}{2} \)

2: \( \Delta(d^3, d^1) - \Delta(d^2, d^1) \geq \Delta(d^3, d^2) > \frac{\nu^2}{2} \)

\( \cdots \)

\( t: \Delta(d^{t+1}, d^1) - \Delta(d^t, d^1) \geq \Delta(d^{t+1}, d^t) > \frac{\nu^2}{2} \)

\( \cdots \)

\( T - 2: \Delta(d^{T-1}, d^1) - \Delta(d^{T-2}, d^1) \geq \Delta(d^{T-1}, d^{T-2}) > \frac{\nu^2}{2} \)

\( T - 1: \Delta(d^T, d^1) - \Delta(d^{T-1}, d^1) \geq \Delta(d^T, d^{T-1}) > \frac{\nu^2}{2} \)

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

\[
\begin{align*}
1: & \quad \Delta(d^2, d^1) - \Delta(d^1, d^1) \geq \Delta(d^2, d^1) > \frac{\nu^2}{2} \\
2: & \quad \Delta(d^3, d^1) - \Delta(d^2, d^1) \geq \Delta(d^3, d^2) > \frac{\nu^2}{2} \\
& \quad \vdots \\
t: & \quad \Delta(d^{t+1}, d^1) - \Delta(d^t, d^1) \geq \Delta(d^{t+1}, d^t) > \frac{\nu^2}{2} \\
& \quad \vdots \\
T - 2: & \quad \Delta(d^{T-1}, d^1) - \Delta(d^{T-2}, d^1) \geq \Delta(d^{T-1}, d^{T-2}) > \frac{\nu^2}{2} \\
T - 1: & \quad \left\{ \Delta(d^T, d^1) - \Delta(d^{T-1}, d^1) \right\}_{\ln N} \geq \Delta(d^T, d^{T-1}) > \frac{\nu^2}{2}
\end{align*}
\]

Therefore, \(T \leq \left\lceil \frac{2 \ln N}{\nu^2} \right\rceil\)
Overview

[Long & Wu, 2002]
Iteration Bounds for Other Variants

Using the same techniques, we can prove iteration bound for:

- TotalBoost\(_{\nu}\) which optimizes the divergence \(\Delta(d, d^t)\) to the last distribution \(d^t\)
- TotalBoost\(_{\nu}\) which uses the binary relative entropy \(\Delta_2(d, d^1)\) or \(\Delta_2(d, d^t)\) as the divergence
- The variant of AdaBoost\(_{\nu}^*\) which terminates when \(\widehat{\gamma}_t < \gamma_t^*\)
TotalBoost, which optimizes $\Delta(d, d^t)$

$d^{t+1}$ is projection of $d^t$ onto $C_t$ at iteration $t$

$$d^{t+1} = \arg\min_{d \in C_t} \Delta(d, d^t)$$

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

Sketch of Proof

1: \[ \Delta(d^T, d^1) - \Delta(d^T, d^2) \geq \Delta(d^2, d^1) > \frac{\nu^2}{2} \]
2: \[ \Delta(d^T, d^2) - \Delta(d^T, d^3) \geq \Delta(d^3, d^2) > \frac{\nu^2}{2} \]
   \[ \cdots \]
   \[ \Delta(d^T, d^t) - \Delta(d^T, d^{t+1}) \geq \Delta(d^{t+1}, d^t) > \frac{\nu^2}{2} \]
   \[ \cdots \]
   \[ \Delta(d^T, d^{T-1}) - \Delta(d^{T}, d^T) \geq \Delta(d^T, d^{T-1}) > \frac{\nu^2}{2} \]
Cancellation

1: \( \Delta(d^T, d^1) - \Delta(d^T, d^2) \geq \Delta(d^2, d^1) > \frac{\nu^2}{2} \)
2: \( \Delta(d^T, d^2) - \Delta(d^T, d^3) \geq \Delta(d^3, d^2) > \frac{\nu^2}{2} \)

\[ \vdots \]
\( t: \Delta(d^T, d^t) - \Delta(d^T, d^{t+1}) \geq \Delta(d^{t+1}, d^t) > \frac{\nu^2}{2} \)
\[ \vdots \]
\( T - 2: \Delta(d^T, d^{T-2}) - \Delta(d^T, d^{T-1}) \geq \Delta(d^{T-1}, d^{T-2}) > \frac{\nu^2}{2} \)
\( T - 1: \Delta(d^T, d^{T-1}) - \Delta(d^T, d^T) \geq \Delta(d^T, d^{T-1}) > \frac{\nu^2}{2} \)

Therefore, \( T \leq \lceil \frac{2 \ln N}{\nu^2} \rceil \)
TotalBoost\(_\nu\) which optimizes \(\Delta(d, d^t)\)
Outline

1 Basics

2 Large margins

3 Games

4 Convergence

5 Illustrative Experiments

[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
Illustrative Experiments

Cox-1 ($\nu = 0.01$)

- AdaBoost$_\nu$
- LPBoost
- TotalBoost$_\nu$

Results

- Corrective algorithms very slow
- LPBoost & TotalBoost need few iterations
- Initial speed crucially depends on $\nu$
Illustrative Experiments

Cox-1 ($\nu = 0.01$)
- $\text{AdaBoost}_\nu$
- $\text{LPBoost}$
- $\text{TotalBoost}_\nu$

Results
- Corrective algorithms very slow
- $\text{LPBoost}$ & $\text{TotalBoost}_\nu$ need few iterations
- Initial speed crucially depends on $\nu$
Illustrative Experiments

Cox-1 ($\nu = 0.01$)
- AdaBoost$_\nu^*$
- LPBoost
- TotalBoost$_\nu$

Results
- Corrective algorithms very slow
- LPBoost & TotalBoost need few iterations
- Initial speed crucially depends on $\nu$
Illustrative Experiments

Cox-1 ($\nu = 0.01$)
- AdaBoost$_\nu$
- LPBoost
- TotalBoost$_\nu$

Results
- Corrective algorithms very slow
  - LPBoost & TotalBoost$_\nu$ need few iterations
  - Initial speed crucially depends on $\nu$
Illustrative Experiments

Cox-1 ($\nu = 0.01$)
- AdaBoost$_\nu$
- LPBoost
- TotalBoost$_\nu$

Results
- Corrective algorithms very slow
- LPBoost & TotalBoost$_\nu$ need few iterations
- Initial speed crucially depends on $\nu$
Illustrative Experiments

Cox-1 ($\nu = 0.01$)
- AdaBoost$_\nu$
- LPBoost
- TotalBoost$_\nu$

Results
- Corrective algorithms very slow
- LPBoost & TotalBoost$_\nu$ need few iterations
- Initial speed crucially depends on $\nu$
LPBoost May Perform Much Worse Than TotalBoost

- Identified cases where LPBoost converges considerably slower than TotalBoost.
- Dataset is a series of artificial datasets of 1000 examples with varying number of features created as follows:
  - First generated $N_1$ random $\pm 1$-valued features $x_1, \ldots, x_{N_1}$ and set the label of the examples as $y = \text{sign}(x_1 + x_2 + x_3 + x_4 + x_5)$
  - 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$_\nu$ 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
Bound Not True for LPBoost

A counter example:

<table>
<thead>
<tr>
<th>Examples</th>
<th>Hypothesis No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

+: correct prediction  -: incorrect prediction

TotalBoost averages hypothesis 1 and 6 (i.e. 2 iterations) to achieve maximum margin 0
### Illustrative Experiments

**Bound Not True for LPBoost**

<table>
<thead>
<tr>
<th>Hypothesis No.</th>
<th>Distribution of Examples</th>
<th>Iteration No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>+ - - - - -</td>
<td>1/9 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>+ - - - - -</td>
<td>1/9 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>+ - - - - -</td>
<td>1/9 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>+ - - - - -</td>
<td>1/9 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>+ - - - - -</td>
<td>1/9 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>+ - - - - -</td>
<td>1/9 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>- + - - - +</td>
<td>1/9 1 0 0 0</td>
<td></td>
</tr>
<tr>
<td>- - + - - +</td>
<td>1/9 0 1 0 0</td>
<td></td>
</tr>
<tr>
<td>- - - + - +</td>
<td>1/9 0 0 1 0</td>
<td></td>
</tr>
<tr>
<td>- - - - + +</td>
<td>1/9 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>- - - - + +</td>
<td>1/9 0 0 0 1</td>
<td></td>
</tr>
</tbody>
</table>

Selected Hypothesis No.

| 1 2 3 4 5 |

Simplex-based LPBoost uses 5 iterations $$((N+1)/2 \text{ iterations})$$ to achieve margin 0
Regularized LPBoost

- LPBoost makes edge constraints as tight as possible.
- Picking solutions at the corners can lead to slow convergence. Interior points methods avoid corners.
- Regularized LPBoost: pick solution that minimizes relative entropy to initial distribution. It is identical to TotalBoost_\nu, but the latter algorithm uses a higher edge bound.
- Open problem: find an example where all versions of LPBoost need $O(N)$ iterations.
Algorithms for Feature Selection

We test for each algorithm:

- Whether selected hypotheses are redundant
- Size of final hypothesis
Redundancy of Selected Hypotheses

- Test which algorithm selects redundant hypotheses
- Dataset: Dataset 2 was created by expanding Rudin’s dataset. Dataset 2 has 100 blocks of features. Each block contains one original feature and 99 mutations of this feature. Each mutation feature inverts the feature value of one randomly chosen example (with replacement). The mutation features are considered redundant hypotheses.
- A good algorithm would avoid repeatedly selecting hypotheses from the same block.
Experiment on Redundancy of Selected Hypotheses

- Show number of selected blocks v.s. number of selected hypotheses
- Redundancy of selected hypotheses: $\text{AdaBoost} > \text{TotalBoost}_\nu(\nu=0.01) > \text{LPBoost}$
- If $\rho^*$ is known, $\text{TotalBoost}_\nu^g$ selects one hypothesis per block (not shown).
Size of Final Hypothesis

- Show margins v.s. number of used hypotheses (nonzero weight)
- $\text{TotalBoost}_\nu (\nu = 0.01)$ and LPBoost use a small number of hypotheses in final hypothesis
Summary

- AdaBoost can be viewed as entropy projection
- TotalBoost projects based on all previous hypotheses
- Provably maximizes the margin
  - Theory: as fast as AdaBoost*ν
  - Practice: much faster (≈ LPBoost)
- Versatile techniques for proving iteration bounds
- Experiments corroborate our theory
  - Good for feature selection
  - LPBoost may have problems of maximizing the margin
- Future: extension to the soft margin case
Iteration Bound for Variant of AdaBoost

\*ν

M.K.Warmuth et.al. (UCSC)  Totally Corrective Boosting Algorithms that ▼  Last update: March 2, 2007  62 / 62