MLSS 2011 Boosting Tutorial
Survey of Boosting from an Optimization Perspective

Manfred K. Warmuth - UCSC
many parts prepared jointly with S.V.N. Vishwanathan - Purdue
greatly expanded from an ICML 2009 tutorial

Updated November 29, 2011
CMPS 242 - Fall 11
1. Introduction to Boosting

2. Squared Euclidean versus relative entropy regularization

3. Boosting as margin maximization with no regularization
   - Game theory interpretation of Boosting

4. LPBoost $\rightarrow$ Entropy Regularized LPBoost
   - Overview of Boosting algorithms
Outline

1. Introduction to Boosting

2. Squared Euclidean versus relative entropy regularization

3. Boosting as margin maximization with no regularization
   - Game theory interpretation of Boosting

4. LPBoost → Entropy Regularized LPBoost
   - Overview of Boosting algorithms
Setup for Boosting

- Fixed set of $\pm 1$ labeled examples
- Easy to find “weak learners” that have error $\leq \frac{1}{2} - \epsilon$
- Can these weak learners be “boosted” to arbitrary high accuracy? - question posed by Kearns in his thesis [K89]
- First recursive Boosting algorithm in Schapire’s thesis [S90]
- AdaBoost: Boosting with 5 lines of code [FS97]
- Here: Fancier Boosting algorithms - big overview
- Black box setup:
  - Assume an oracle provides weak learners for weighted examples
  - The boosting algorithm “aggregates” the weak learners
Protocol of Boosting

- Maintain distribution on $N \pm 1$ labeled examples
- At iteration $t = 1, \ldots, T$:
  - Receive “weak” hypothesis $h^t$ from oracle
  - Update $d^{t-1}$ to $d^t$: put more weights on “hard” examples
- Output convex combination of the $T$ weak hypotheses

Goals:
- Final hypothesis highly accurate
- Small number of iterations
- No overfitting
Classifying spam

Weak/base hypotheses

- “viagra” in text
- Mailed from certain sites
- Any fancy spam classifier that can steal from friend or foe
  - base hypothesis don’t have to be “weak”
  - combine anything and get an improvement
  - use large variety of base learners
    - simple decision trees
    - SVMs
    - nearest neighbor
Other example problems

- Classify pictures as to whether they contain a dog?
- Does picture contain a human face? [Viola]

Most common base learners

- “Good” feature from a large set of binary features that are known to work well for your problem
- Decision stumps on a large set of real features

\[ h_\theta^i = (\text{feature}_i \geq \theta) \]

Boosting is simple versatile technique for building strong classifiers

- provided you have oracle for providing good base learner
- coordinate descent
- choice is greedy - i.e. oracle iteratively provides a good feature
Our running example: Classifying apples

- examples: 11 apples
- +1 if artificial
- -1 if natural
- goal: accurate classification
- 2 real features: redness & lightness
Setup for Boosting

- $+1/-1$ examples
- Weight $d_n \approx$ size
- Separable
Weak hypotheses

- weak hypotheses: decision stumps on two features
- one can’t do it
- goal: find convex combination of weak hypotheses that classifies all
Boosting: 1st iteration

First hypothesis:
- error: $\frac{1}{11}$
- edge: $\frac{9}{11}$

low error = high edge

edge = $1 - 2 \text{ error}$
Accuracy on example / edge

\[
\begin{array}{c|c}
\text{ynhn(xn)} & \text{un} \\
\hline
\text{perfect} & +1 \\
\text{wrong} & -1 \\
\text{neutral} & 0 \\
\end{array}
\]

\( u_n \) is accuracy of \( h_n \) on example \((x_n, y_n)\)

\text{edge} is accuracy on all examples weighted by distribution

\[
\sum_n d_n u_n = d \cdot u
\]
### Weak hypothesis - column vector of accuracies

<table>
<thead>
<tr>
<th>examples $x_n$</th>
<th>labels $y_n$</th>
<th>1st stump $h^1(x_n)$</th>
<th>accuracies $u^1_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>
Update after 1st

Misclassified examples
- increased weights

After update
- edge of hypothesis decreased
Before 2nd iteration

Hard examples

- high weight
Boosting: 2nd hypothesis

Pick hypotheses with high (weighted) edge
Update after 2nd

After update
- edges of all past hypotheses should be small
3rd hypothesis
Update after 3rd
4th hypothesis
Update after 4th

![Diagram of feature 1 vs feature 2 with points distributed across the axes.]

Warmuth (UCSC) MLSS 2011, Purdue
Final convex combination of all hypotheses

Decision: $\sum_{t=1}^{T} w^t h^t(x) \geq 0$?

Positive total weight - Negative total weight
Protocol of Boosting

Maintain distribution on $N \pm 1$ labeled examples

At iteration $t = 1, \ldots, T$:
- Receive “weak” hypothesis $h^t$ from oracle of high edge
- Update $d^{t-1}$ to $d^t$: put more weights on “hard” examples

Output convex combination of the weak hypotheses
$$\sum_{t=1}^{T} w^t h^t(x)$$

Two sets of weights:
- distribution on $d$ on examples
- distribution on $w$ on hypotheses
### Recall data representation

The accuracy of an example can be represented as:

$$y_n h^t(x_n) := u^t_n$$

<table>
<thead>
<tr>
<th>Accuracy of example</th>
<th>$y_n h^t(x_n)$ := $u^t_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>perfect</td>
<td>+1</td>
</tr>
<tr>
<td>opposite</td>
<td>-1</td>
</tr>
<tr>
<td>neutral</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>examples $x_n$</th>
<th>labels $y_n$</th>
<th>$h^t(x_n)$</th>
<th>$u^t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

Warmuth (UCSC) MLSS 2011, Purdue
AdaBoost update

\[ d_n^t \sim d_n^{t-1} \exp(-w^t u_n^t) \]

where

- \( d_n^{t-1} \) are old weights
- \( u_n^t \in \pm 1 \) are accuracies
- \( w^t \) is “learning rate” / coefficient of hypothesis \( h^t \)

\[ w^t = \frac{1}{2} \ln \frac{1 + d_n^{t-1} \cdot u_n^t}{1 - d_n^{t-1} \cdot u_n^t} \]

Oracle gives weak hypothesis such that \( \underbrace{d_n^{t-1} \cdot u_n^t}_{\text{edge}} \geq \gamma \geq 0 \)
AdaBoost

+ 
  - Trivial to code, provided you have oracle
  - Fast update
  - Has iteration bound
    consistent hypothesis in $\leq \frac{\ln n}{\gamma^2}$ iterations
  - Good initial Boosting algorithm

- 
  - Too many iterations
  - Same hypothesis chosen multiple times
  - Cycles on inseparable case
Questions

- How to motivate Boosting updates?
- What is underlying optimization problem?
- What is the regularization?
- How do we get bound on number of iterations?
Two applications of Boosting

- Build strong classifier from weak classifier
- Use Boosting as filter
  - Determine the examples that remain hard to classify after combining a set of base classifier
  - Those examples are likely to be “noisy”
  - Present expensing human agent only with hard examples
Outline

1. Introduction to Boosting

2. Squared Euclidean versus relative entropy regularization

3. Boosting as margin maximization with no regularization
   - Game theory interpretation of Boosting

4. LPBoost $\rightarrow$ Entropy Regularized LPBoost
   - Overview of Boosting algorithms
Motivations of the updates?

- Motivate additive and multiplicative updates
- Use linear regression as example problem
- Motivate all updates as minimizing

$$\text{regularization} + \eta \text{ loss/objective}$$

- Sometimes additional constraints
- Sometimes no regularization
Online linear regression

For $t = 1, 2, \ldots$

- Get instance $x_t \in \mathbb{R}^n$
- Predict $\hat{y}_t = w_t \cdot x_t$
- Get label $y_t \in \mathbb{R}$
- Incur square loss $(y_t - \hat{y}_t)^2$
- Update $w_t \rightarrow w_{t+1}$
Two main update families - linear regression

- **Additive**

\[ w_{t+1} = w_t - \eta \left( w_t \cdot x_t - y_t \right) x_t \]

Gradient of Square Loss

- Motivated by squared Euclidean distance
- Weights can go negative
- Gradient Descent \((GD)\)

- **Multiplicative**

\[ w_{t+1,i} = \frac{w_{t,i} e^{-\eta(w_t \cdot x_t - y_t)} x_{t,i}}{Z_t} \]

- Motivated by relative entropy
- Updated weight vector stays on probability simplex
- Exponentiated Gradient \((EG)\) \[KW97\]

[MLSS 2011, Purdue]
Additive Updates

**Goal**

Find $w_{t+1}$ close to old $w_t$ that has small loss on last example OR Minimize tradeoff between closeness and loss

$$w_{t+1} = \arg\min_w U(w)$$

$$U(w) = \left\| w - w_t \right\|_2^2 + \eta (w \cdot x_t - y_t)^2$$

$\eta > 0$ is the *learning rate*
Additive updates

\[
\frac{\partial U(w)}{\partial w_i} \bigg|_{w_i = w_{t+1}, i} = 2(w_{t+1,i} - w_{t,i}) + 2\eta(w \cdot x_t - y_t)x_t, i = 0
\]

Therefore,

**implicit:** \[w_{t+1} = w_t - \eta(w_{t+1} \cdot x_t - y_t)x_t\]

**explicit:** \[= w_t - \eta(w_t \cdot x_t - y_t)x_t\]
Multiplicative updates

\[ \mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} U(\mathbf{w}) \quad \text{where} \quad U(\mathbf{w}) = \sum_i w_i \ln \frac{w_i}{w_{t,i}} + \eta (\mathbf{w} \cdot \mathbf{x}_t - y_t)^2 \]

Define Lagrangian

\[ L(\mathbf{w}) = \sum w_i \ln \frac{w_i}{w_{t,i}} + \eta (\mathbf{w} \cdot \mathbf{x}_t - y_t)^2 + \lambda (\sum_i w_i - 1) \]

where \( \lambda \) Lagrange coeff.
Multiplicative updates

\[
\frac{\partial L(w)}{\partial w_i} = \ln \frac{w_i}{w_{t,i}} + 1 + \eta (w \cdot x_t - y_t) x_{t,i} + \lambda = 0
\]

\[
\ln \frac{w_{t+1,i}}{w_{t,i}} = -\eta (w_{t+1} \cdot x_t - y_t) x_{t,i} - \lambda - 1
\]

\[
w_{t+1,i} = w_{t,i} e^{-\eta (w_{t+1} \cdot x_t - y_t) x_{t,i}} e^{-\lambda - 1}
\]

Enforce normalization constraint by setting \(e^{-\lambda - 1}\) to \(1/Z_t\)

**implicit:**
\[
w_{t+1,i} = \frac{w_{t,i} e^{-\eta w_{t+1} \cdot x_t - y_t} x_{t,i}}{Z_t}
\]

**explicit:**
\[
w_{t+1,i} = \frac{w_{t,i} e^{-\eta w_{t} \cdot x_t - y_t} x_{t,i}}{Z_t'}
\]

Warmuth (UCSC) MLSS 2011, Purdue
Minimizing tradeoff

- Batch: tradeoff between regularization and total loss
- Regularization determines how parameter space is searched

Two main regularization
- squared Euclidean distance
- relative entropy
## Two families of updates

<table>
<thead>
<tr>
<th>$| \cdot |_2^2$</th>
<th>relative entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widrow Hoff update</td>
<td>EG update</td>
</tr>
<tr>
<td>Perceptron</td>
<td>Winnow</td>
</tr>
<tr>
<td>Backprop</td>
<td>Weighted Majority algorithm</td>
</tr>
<tr>
<td>SVMs</td>
<td>Baysian update</td>
</tr>
<tr>
<td>Newton’s method</td>
<td>Boosting</td>
</tr>
</tbody>
</table>

### Different properties

<table>
<thead>
<tr>
<th>ignores small weights</th>
<th>smoothed 1-norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>log of a Gaussian</td>
<td>information theoretic</td>
</tr>
<tr>
<td>rotation invariant</td>
<td>not rotation invariant</td>
</tr>
</tbody>
</table>
Send symbol $X$ on channel

<table>
<thead>
<tr>
<th>$X$</th>
<th>$P(X = x_i)$</th>
<th>$- \log P(x_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$\frac{1}{2}$</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$\frac{1}{4}$</td>
<td>2</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$\frac{1}{8}$</td>
<td>3</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$\frac{1}{8}$</td>
<td>3</td>
</tr>
</tbody>
</table>

$- \log P(x_i)$ is measure of surprise in bits

- $- \log 1 = 0$  no surprise
- $- \log 0 = \infty$ infinite surprise
- $- \log \frac{1}{2^i} = i$ $i$ bits of surprise
Entropy equals expected surprise

\[ H(X) = \sum_i P(x_i) \log \frac{1}{p(x_i)} \]

\[ = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} \]

\[ = 1 \frac{3}{4} \text{ bits} \]

\[-p \log p - (1 - p) \log (1 - p) \]

3-dim entropy unif. in dim. \( n \)

\[ \log(n) \]
Code

Assigns symbols a bitstring (codeword)
- any sequence of codewords must be uniquely decodable

Expected codelength

$$L(C) = \sum_{i} p(x_i) \ell_C(x_i)$$

length of codeword for $x_i$

Optimal code $C^*$
- $L(C^*)$ is minimum
  - Thm: $H(X) \leq L(C^*) \leq H(X) + 1$
  - Thm: Huffman codes are optimal
  - More info: first five chapters of Cover & Thomas
Relative entropy between distributions $p$ and $q$

$$\Delta(p, q) = \sum_i p_i \log \frac{p_i}{q_i}$$

$$= \sum_i p_i \log \frac{1}{q_i} - \sum_i p_i \log \frac{1}{p_i}$$

where all expectations are wrt $p$
Relative entropy to the uniform distribution

\[ \Delta(p, \frac{1}{n}) = \sum_i p_i \log \frac{p_i}{1/n} \]
\[ = \sum_i p_i \log p_i + p_i \log n \]
\[ = \log n - H(p) \]
\[ \geq 0 \]

0 at center of simplex

In general

\[ \Delta(p, q) \geq 0, \]

where equality holds iff \( p = q \)
Which argument should be the variable?

- Not too steep at boundary
- Steep at boundary

Motivates EG and Boosting
Squared Euclidean versus relative entropy regularization

Two relative entropies in 3D

Contour plot of relative entropy with first argument as variable

Contour plot of relative entropy with second argument as variable

Both are barriers for simplex
Squared Euclidean versus relative entropy regularization

Use of relative entropy (w. first argument as var.)

- As regularizer in motivation of update
- As measure of progress in analysis

Squared Euclidean distance “ignores” the simplex
Questions

- What optimization problems motivate Boosting?
- Why is relative entropy used as a regularization?
- What is the loss/objective for Boosting?
- What if no regularization is used?
Outline

1. Introduction to Boosting
2. Squared Euclidean versus relative entropy regularization
3. Boosting as margin maximization with no regularization
   - Game theory interpretation of Boosting
4. LPBoost → Entropy Regularized LPBoost
   - Overview of Boosting algorithms
What is the objective of Boosting?

- Give objective into edges and margins
- Minimize objective without regularization
  - Game theoretic interpretation of Boosting
  - Column generation interpretation
  - Minimizing objective alone: LPBoost
### Recall data representation

The accuracy of an example is defined as:

$$ y_n h^t(x_n) := u^t_n $$

<table>
<thead>
<tr>
<th>Accuracy of example</th>
<th>$y_n h^t(x_n) := u^t_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>perfect</td>
<td>+1</td>
</tr>
<tr>
<td>opposite</td>
<td>-1</td>
</tr>
<tr>
<td>neutral</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>examples $x_n$</th>
<th>labels $y_n$</th>
<th>$h^t(x_n)$</th>
<th>$u^t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

Warmuth (UCSC)  
MLSS 2011, Purdue  
50 / 87
Edge vs. margin

Edge of a hypothesis $h^t$ for a distribution $d$ on the examples

$$
\sum_{n=1}^{N} \sum_{t} \left( u^t_n \cdot d_n \right) \quad d \in \mathcal{P}^N
$$

weighted accuracy of hypothesis

Margin of example $n$ for current hypothesis weighting $w$

$$
\sum_{t=1}^{T} \sum_{t} \left( u^t_n \cdot w_t \right) \quad w \in \mathcal{P}^T
$$

weighted accuracy of example
**Edge vs. margin**

**Edge of a hypothesis** $h^t$ for a distribution $d$ on the examples

$$\sum_{n=1}^{N} d_n \sum_{t=1}^{T} u_n^t d_n \quad d \in \mathcal{P}^N$$

**Margin of example** $n$ for current hypothesis weighting $w$

$$\sum_{t=1}^{T} w_t u_n^t w \quad w \in \mathcal{P}^T$$
Objective in the $d$ domain

- Edges of past hypotheses should be small after update
  - More weight on hard (low accuracy examples) decreases weighted accuracy $= \text{edge}$
- Minimize maximum edge of past hypotheses
Objective in the $w$ domain

- Choose convex combination of weak hypotheses that **maximizes the minimum margin** of the examples.

Which margin in $w$ domain?
- SVM 2-norm
- Boosting 1-norm
Connection between objectives?

$$\min_{d \in S^N} \max_{q=1,2,...,t-1} \ u^q \cdot d = \max_{w \in S^{t-1}} \ min_{n=1,2,...,N} \ \sum_{q=1}^{t-1} u^q_n w^q$$

Van Neumann’s Minimax Theorem
Boosting as zero-sum-game [FS97]

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>Column</td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>$w_3$</td>
</tr>
</tbody>
</table>

Gain matrix

Row player minimizes
Column player maximizes payoff

$$\text{payoff} = d^T U w = \sum_{i,j} d_i U_{i,j} w_j$$

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

Single column is pure strategy of column player and $w$ is mixed strategy
Optimum strategy

\[
\begin{array}{ccc}
R & P & S \\
\begin{array}{ccc}
\{ w_1, w_2, w_3 \} & .33 & .33 & .33 \\
\end{array}
\end{array}
\]

\[
\begin{array}{ccc}
R & d_1 & .33 & 0 & 1 & -1 \\
P & d_2 & .33 & -1 & 0 & 1 \\
S & d_3 & .33 & 1 & -1 & 0 \\
\end{array}
\]

- Minimax theorem:

\[
\min_d \max_w d^T U w = \min_d \max_j d^T U e_j \\
= \max_w \min_d d^T U w = \max_w \min_i e_i^T U w \\
= \text{value of the game (} 0 \text{ in example)}
\]

\( e_j \) is pure strategy.
Connection to Boosting?

Payoff matrix $\mathbf{U}$

- Rows are the examples
- Columns $\mathbf{u}^q$ encode weak hypothesis $h^q$
- Row sum: margin of example
- Column sum: edge of weak hypothesis
- Value of game:

$$\min \max \text{ edge} = \max \min \text{ margin}$$

Van Neumann’s Minimax Theorem
### Edges/margins

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>w₁</td>
<td>.33</td>
<td>.33</td>
<td>.33</td>
</tr>
</tbody>
</table>

**R**  | **P**  | **S**  |
---|---|---|
**d₁** | .33 | 0 | 1 | -1 | 0 |
**d₂** | .33 | -1 | 0 | 1 | 0 |
**d₃** | .33 | 1 | -1 | 0 | 0 |

Edge: 0

Max value of game: 0

Warmuth (UCSC)
New column added: Boosting

Helps maximizing column player

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
<th>w_4</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>.44</td>
<td>0</td>
<td>.22</td>
<td>.33</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>d_1</td>
<td>.22</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d_2</td>
<td>.33</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>d_3</td>
<td>.44</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

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

Helps minimizing row player

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>$w_3$</td>
</tr>
<tr>
<td></td>
<td>.33</td>
<td>.44</td>
<td>.22</td>
</tr>
</tbody>
</table>

| d_1 | 0    | 0    | 1    | -1    |
| d_2 | .22  | -1   | 0    | 1     |
| d_3 | .44  | 1    | -1   | 0     |
| d_4 | .33  | -1   | 1    | -1    |

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

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>iteration 1</td>
<td>$d_1^1$</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>$d_2^1$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$d_3^1$</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>iteration 2</td>
<td>$d_1^2$</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>$d_2^2$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$d_3^2$</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>iteration 3</td>
<td>$d_1^3$</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>$d_2^3$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$d_3^3$</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

- In each iteration solve optimization problem to update $d$
- Column player = oracle provides new hypothesis
- Boosting is column generation method in $d$ domain and coordinate ascent in $w$ domain
Boosting = greedy method for increasing margin

Converges to optimum margin w.r.t. all hypotheses

Want small number of iterations
Power of the oracle

**Strong oracle:**
- Return hypothesis of maximum edge

**Goal:**
- For given $\epsilon$, produce convex combination of weak hypotheses with soft margin $\geq$ value $- \epsilon$

**Weak oracle:**
- For some guarantee $g$, return hypothesis of edge $\geq g$

**Goal:**
- For given $\epsilon$, produce convex combination of weak hypotheses with soft margin $\geq g - \epsilon$
Visualizing the margin

Not the margin corresponding to Boosting
1-norm versus 2-norm margin

Boosting with $w$ in diamond rather than simplex:

$$\min_{d \in S^N} \max_{q=1,2,...,t} |u^q \cdot d| = \max_{w \in D^t} \min_{n=1,2,...,N} U_{n^*} w$$

$$= \max_{w \in R^t} \min_{n=1,2,...,N} U_{n^*} \frac{w}{||w||_1}$$

2-norm margins as in SVMs:

$$\min_{d \in S^n} \max_{w \in R^t} d^T U \frac{w}{||w||_2} = \max_{w \in R^t} \min_{d \in S^n} d^T U \frac{w}{||w||_2}$$

$$= \max_{w \in R^t} \min_{n=1,2,...,N} U_{n^*} \frac{w}{||w||_2}$$
Outline

1. Introduction to Boosting
2. Squared Euclidean versus relative entropy regularization
3. Boosting as margin maximization with no regularization
   - Game theory interpretation of Boosting
4. LPBoost → Entropy Regularized LPBoost
   - Overview of Boosting algorithms
Choose distribution that minimizes the maximum edge of current hypotheses by solving w. LP

\[
\min \sum_{n} d_n = 1 \frac{\max_{q=1,2,\ldots,t} u^q \cdot d}{P_{t}^{L}}
\]

All weight is put on examples with minimum margin.
Entropy Regularized LPBoost

\[
\min_{\sum_n d_n = 1} \left( \frac{1}{\eta} \Delta(d, d^0) + \max_{q=1,2,…,t} u^q \cdot d \right)
\]

- \(d_n = \exp^{-\eta \text{ margin of example } n} \frac{1}{Z} \)
  - "soft min"

- Form of weights first in \(\nu\)-Arc algorithm [RSS+00]
- Regularization in \(d\) domain makes problem strongly convex
- Gradient of dual Lipschitz continuous in \(w\) [e.g. HL93, RW97]
Effect of entropy regularization

Different distributions on the examples

**LPBoost:** lots of zeros / brittle

**ERLPBoost:** smoother
Effect of the regularization in the $d$ domain

$$\min_{\sum_n d_n = 1} \left( \frac{1}{\eta} \Delta(d, d^0) + \max_{q=1,2,\ldots,t} u^q \cdot d \right)$$

Uncapped case, $\eta = \infty$ becomes LP objective
From ERLPBoost to SVMs

Two steps removed
- Replace 1-norm margin by 2-norm margin
- Replace relative entropy regularization by $\| \cdot \|^2$

What is the simplest path?
- minimax duality - Lagrange duality - Fenchel duality

Many things are easier for quadratic regularization:
- Slack variables
- Bias term

Warmuth (UCSC)
Minimax thm - inseparable case

Slack variables in $w$ domain = capping in $d$ domain

$$\min_{d \in S^N, d \leq \frac{1}{\nu} 1} \max_{q=1,2,...,t} u^q \cdot d$$

$$= \max_{w \in S^t, \psi \geq 0} \min_{n=1,2,...,N} \left( \sum_{q=1}^{t} u^q_n w^q + \psi_n \right) - \frac{1}{\nu} \sum_{n=1}^{N} \psi_n$$

soft margin of example $n$
Visualizing the soft margin
Adding slack variables to the algorithms

LPBoost

\[
\min \sum_{n} d_n = 1, d \leq \frac{1}{\nu} \max_{q=1,2,...,t} u^q \cdot d
\]

All weight put on examples with minimum soft margin ERLPBoost

\[
\min \sum_{n} d_n = 1, d \leq \frac{1}{\nu} \left( \frac{1}{\eta} \Delta(d, d^0) + \max_{q=1,2,...,t} u^q \cdot d \right)
\]

\[
d_n = \exp^{-\eta \text{ soft margin of example } n} \frac{1}{Z} \quad \text{"soft min"}
\]
AdaBoost - the most common motivation \[\text{[FS97]}\]

\[
d^t_n := \frac{d^{t-1}_n \exp(-w^t u^t_n)}{\sum_{n'} d^{t-1}_{n'} \exp(-w^t u^t_{n'})},
\]

where \(w^t\) s.t. \(-\ln \sum_n d^{t-1}_n \exp(-w u^t_n)\) is minimized

Choose \(w\) s.t.
\[
\frac{\partial - \ln \sum_{n'} d^{t-1}_{n'} \exp(-w u^t_{n'})}{\partial w} = u^t \cdot d^t(w) = 0
\]

- Easy to implement
- Adjusts distribution so that last hypothesis has edge zero
- When \(h^t_n \in \{-1, +1\}\), then \(w^t = \frac{1}{2} \ln \frac{1+d^{t-1}_n u^t}{1-d^{t-1}_n u^t}\)
  when \(h^t_n \in [-1, +1]\), then line search
- Gets within half of the optimal hard margin but only in the limit \[\text{[RSD07]}\]
Corrective versus totally corrective

<table>
<thead>
<tr>
<th>Corrective</th>
<th>Totally Corrective</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>LPBoost</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>TotalBoost</td>
</tr>
<tr>
<td>AdaBoost*</td>
<td>SoftBoost</td>
</tr>
<tr>
<td>SS, Colt08</td>
<td>ERLPBoost</td>
</tr>
</tbody>
</table>
From AdaBoost to ERLPBoost

**AdaBoost**

Primal:

\[
\begin{align*}
\min_d & \quad \Delta(d, d^{t-1}) \\
\text{s.t.} & \quad d \cdot u^t = 0, \|d\|_1 = 1
\end{align*}
\]

Dual:

\[
\begin{align*}
\max_w & \quad -\ln \sum_n d_n^{t-1} \exp(-u_n^t w) \\
\text{s.t.} & \quad w \geq 0
\end{align*}
\]

Achieves half of optimum hard margin in the limit

(as interpreted in [KW99,La99])

---

**AdaBoost**

Primal:

\[
\begin{align*}
\min_d & \quad \Delta(d, d^{t-1}) \\
\text{s.t.} & \quad d \cdot u^t \leq \gamma_t, \|d\|_1 = 1
\end{align*}
\]

Dual:

\[
\begin{align*}
\max_w & \quad -\ln \sum_n d_n^{t-1} \exp(-u_n^t w) - \gamma_t \|w\|_1 \\
\text{s.t.} & \quad w \geq 0
\end{align*}
\]

where edge bound \(\gamma_t\) is adjusted downward by a heuristic

Good iteration bound for reaching optimum hard margin

[RW05]
**SoftBoost**

**Primal:**
\[
\begin{align*}
\min_{d} & \quad \Delta(d, d^0) \\
\text{s.t.} & \quad \|d\|_1 = 1, \quad d \leq \frac{1}{\nu} 1 \\
& \quad d \cdot u^q \leq \gamma_t, \quad 1 \leq q \leq t
\end{align*}
\]

**Dual:**
\[
\begin{align*}
\min_{w, \psi} & \quad -\ln \sum_n d_n^0 \exp(-\eta \sum_{q=1}^t u_n^q w^q - \eta \psi_n) \\
& \quad -\frac{1}{\nu} \|\psi\|_1 - \gamma_t \|w\|_1 \\
\text{s.t.} & \quad w \geq 0, \quad \psi \geq 0
\end{align*}
\]

where edge bound $\gamma_t$ is adjusted downward by a heuristic.

Good iteration bound for reaching soft margin

**ERLPBoost**

**Primal:**
\[
\begin{align*}
\min_{d, \gamma} & \quad \gamma + \frac{1}{\eta} \Delta(d, d^0) \\
\text{s.t.} & \quad \|d\|_1 = 1, \quad d \leq \frac{1}{\nu} 1 \\
& \quad d \cdot u^q \leq \gamma, \quad 1 \leq q \leq t
\end{align*}
\]

**Dual:**
\[
\begin{align*}
\min_{w, \psi} & \quad -\frac{1}{\eta} \ln \sum_n d_n^0 \exp(-\eta \sum_{q=1}^t u_n^q w^q - \eta \psi_n) \\
& \quad -\frac{1}{\nu} \|\psi\|_1 \\
\text{s.t.} & \quad w \geq 0, \quad \|w\|_1 = 1, \quad \psi \geq 0
\end{align*}
\]

where for the iteration bound $\eta$ is fixed to $\max\left(\frac{2}{\epsilon} \ln \frac{N}{\nu}, \frac{1}{2}\right)$.

Good iteration bound for reaching soft margin
Corrective ERLPBoost

Primal:

\[
\begin{align*}
\min_d & \quad (\sum_{q=1}^{t} w^q u^q) \cdot d + \frac{1}{\eta} \Delta(d, d^0) \\
\text{s.t.} & \quad \|d\|_1 = 1, \quad d \leq \frac{1}{\nu} \mathbf{1} 
\end{align*}
\]

Dual:

\[
\begin{align*}
\min_{\psi} & \quad -\frac{1}{\eta} \ln \sum_{n} d^0_n \exp(-\eta \sum_{q=1}^{t} u^q_n w^q - \eta \psi_n) - \frac{1}{\nu} \|\psi\|_1 \\
\text{s.t.} & \quad \psi \geq 0 
\end{align*}
\]

where for the iteration bound \( \eta \) is fixed to \( \max\left(\frac{2}{\epsilon} \ln \frac{N}{\nu}, \frac{1}{2}\right) \)

Good iteration bound for reaching soft margin
## Iteration bounds

<table>
<thead>
<tr>
<th>Corrective</th>
<th>Totally Corrective</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>LPBoost</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>TotalBoost</td>
</tr>
<tr>
<td>AdaBoost*</td>
<td>SoftBoost</td>
</tr>
<tr>
<td>SS, Colt08</td>
<td>ERLLPBoost</td>
</tr>
</tbody>
</table>

- **Strong oracle:** returns hypothesis with maximum edge
- **Weak oracle:** returns hypothesis with edge $\geq g$

- In $O\left(\frac{\log N}{\epsilon^2}\right)$ iterations
  - within $\epsilon$ of maximum soft margin for strong oracle
  - or within $\epsilon$ of $g$ for weak oracle
- **Ditto for hard margin case**
- **When $g > 0$, in** $O\left(\frac{\log N}{g^2}\right)$ iterations
  - consistency with weak oracle
LPBoost may require $\Omega(N)$ iterations

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>.125</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_2$</td>
<td>.125</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_3$</td>
<td>.125</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_4$</td>
<td>.125</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_5$</td>
<td>.125</td>
<td>-.98</td>
<td>+1</td>
<td>-.93</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_6$</td>
<td>.125</td>
<td>-.97</td>
<td>-.96</td>
<td>+1</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_7$</td>
<td>.125</td>
<td>-.97</td>
<td>-.95</td>
<td>-.94</td>
<td>+1</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_8$</td>
<td>.125</td>
<td>-.97</td>
<td>-.95</td>
<td>-.93</td>
<td>-.92</td>
<td>+.99</td>
</tr>
<tr>
<td>edge</td>
<td>.0137</td>
<td>-.7075</td>
<td>-.6900</td>
<td>-.6725</td>
<td>.0000</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LPBoost may require $\Omega(N)$ iterations

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_4$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_5$</td>
<td>1</td>
<td>-.98</td>
<td>+1</td>
<td>-.93</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_6$</td>
<td>0</td>
<td>-.97</td>
<td>-.96</td>
<td>+1</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_7$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.94</td>
<td>+1</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_8$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.93</td>
<td>-.92</td>
<td>+.99</td>
</tr>
<tr>
<td>edge</td>
<td>-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>value</td>
<td>-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
LPBoost may require $\Omega(N)$ iterations

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_4$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_5$</td>
<td>0</td>
<td>-.98</td>
<td>+1</td>
<td>-.93</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_6$</td>
<td>1</td>
<td>-.97</td>
<td>-.96</td>
<td>+1</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_7$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.94</td>
<td>+1</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_8$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.93</td>
<td>-.92</td>
<td>+.99</td>
</tr>
</tbody>
</table>

| edge  | -.97  | -.96  | 1     | -.91  | .99   |
| value | -1    | -.98  | -.96  |       |       |
LPBoost may require $\Omega(N)$ iterations

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_4$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_5$</td>
<td>0</td>
<td>-.98</td>
<td>+1</td>
<td>-.93</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_6$</td>
<td>0</td>
<td>-.97</td>
<td>-.96</td>
<td>+1</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_7$</td>
<td>1</td>
<td>-.97</td>
<td>-.95</td>
<td>-.94</td>
<td>+1</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_8$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.93</td>
<td>-.92</td>
<td>+.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>edge</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.97</td>
<td>-.95</td>
</tr>
</tbody>
</table>

| value | -1 | -.98 | -.96 | -.94 |
**LPBoost may require $\Omega(N)$ iterations**

<table>
<thead>
<tr>
<th>$d$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-.91</td>
</tr>
<tr>
<td>$d_1$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_4$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_5$</td>
<td>0</td>
<td>-.98</td>
<td>+1</td>
<td>-.93</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_6$</td>
<td>0</td>
<td>-.97</td>
<td>-.96</td>
<td>+1</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_7$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.94</td>
<td>+1</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_8$</td>
<td>1</td>
<td>-.97</td>
<td>-.95</td>
<td>-.93</td>
<td>-.92</td>
<td>+.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>edge</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
LPBoost may require $\Omega(N)$ iterations

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>.497</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_4$</td>
<td>0</td>
<td>+1</td>
<td>-.95</td>
<td>-.93</td>
<td>-.91</td>
<td>-.99</td>
</tr>
<tr>
<td>$d_5$</td>
<td>0</td>
<td>-.98</td>
<td>+1</td>
<td>-.93</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_6$</td>
<td>.490</td>
<td>-.97</td>
<td>-.96</td>
<td>+1</td>
<td>-.91</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_7$</td>
<td>0</td>
<td>-.97</td>
<td>-.95</td>
<td>-.94</td>
<td>+1</td>
<td>+.99</td>
</tr>
<tr>
<td>$d_8$</td>
<td>.013</td>
<td>-.97</td>
<td>-.95</td>
<td>-.93</td>
<td>-.92</td>
<td>+.99</td>
</tr>
</tbody>
</table>

No ties!
Synopsis

- LPBoost often unstable
- For safety, add relative entropy regularization
- Corrective algs
  - Sometimes easy to code
  - Fast per iteration
- Totally corrective algs
  - Smaller number of iterations
  - Faster overall time when $\epsilon$ small
- Weak versus strong oracle makes a big difference in practice