Foundations of Machine Learning
Boosting

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

(Kearns and Valiant, 1994)

Definition: concept class $C$ is weakly PAC-learnable if there exists a (weak) learning algorithm $L$ and $\gamma > 0$ such that:

- for all $\delta > 0$, for all $c \in C$ and all distributions $D$,

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

- for samples $S$ of size $m = poly(1/\delta)$ for a fixed polynomial.
Finding simple relatively accurate base classifiers often not hard weak learner.

Main ideas:

- use weak learner to create a strong learner.
- combine base classifiers returned by weak learner (ensemble method).

But, how should the base classifiers be combined?
AdaBoost

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

**AdaBoost** \((S = ((x_1, y_1), \ldots, (x_m, y_m)))\)

1. \textbf{for} \( i \leftarrow 1 \) \textbf{to} \( m \) \textbf{do}
2. \hspace{1em} \( D_1(i) \leftarrow \frac{1}{m} \)
3. \textbf{for} \( t \leftarrow 1 \) \textbf{to} \( T \) \textbf{do}
4. \hspace{1em} \( h_t \leftarrow \text{base classifier in } H \text{ with small error } \epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i] \)
5. \hspace{1em} \( \alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t} \)
6. \hspace{1em} \( Z_t \leftarrow 2[\epsilon_t (1 - \epsilon_t)]^{\frac{1}{2}} \quad \triangleright \text{normalization factor} \)
7. \textbf{for} \( i \leftarrow 1 \) \textbf{to} \( m \) \textbf{do}
8. \hspace{1em} \( D_{t+1}(i) \leftarrow \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \)
9. \hspace{1em} \( f_t \leftarrow \sum_{s=1}^{t} \alpha_s h_s \)
10. \textbf{return} \( h = \text{sgn}(f_T) \)
Notes

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

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

■ Weight assigned to base classifier $h_t$: $\alpha_t$ directly depends on the accuracy of $h_t$ at round $t$. 
Illustration

$t = 1$

$t = 2$
\[ t = 3 \]
\[ \alpha_1 + \alpha_2 + \alpha_3 = \]

- A mathematical expression involving \( \alpha_1 \), \( \alpha_2 \), and \( \alpha_3 \) is shown, with each term represented by a colored shape. The shapes are divided into regions with red and blue dots, indicating some form of partition or distribution. This visual representation seems to illustrate the concept of combining these quantities, possibly in the context of machine learning or a related field.
Bound on Empirical Error

(Freund and Schapire, 1997)

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

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

- If further for all \( t \in [1, T] \), \( \gamma \leq \left( \frac{1}{2} - \epsilon_t \right) \), then
  \[
  \hat{R}(h) \leq \exp(-2\gamma^2 T).
  \]
- \( \gamma \) does not need to be known in advance: adaptive boosting.
• **Proof:** Since, as we saw, \( D_{t+1}(i) = \frac{e^{-y_if_t(x_i)}}{m \prod_{s=1}^{t} Z_s} \),

\[
\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{y_if(x_i) \leq 0} \leq \frac{1}{m} \sum_{i=1}^{m} \exp(-y_if(x_i))
\]

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

• Now, since \( Z_t \) is a normalization factor,

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

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

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

\[
= (1 - \epsilon_t) \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = 2 \sqrt{\epsilon_t (1 - \epsilon_t)}.
\]
• Thus,

\[
\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t (1 - \epsilon_t)} = \prod_{t=1}^{T} \sqrt{1 - 4 \left( \frac{1}{2} - \epsilon_t \right)^2} \\
\leq \prod_{t=1}^{T} \exp \left[ -2 \left( \frac{1}{2} - \epsilon_t \right)^2 \right] = \exp \left[ -2 \sum_{t=1}^{T} \left( \frac{1}{2} - \epsilon_t \right)^2 \right].
\]

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

- **Objective Function**: convex and differentiable.

\[
F(\bar{\alpha}) = \frac{1}{m} \sum_{i=1}^{m} e^{-y_i f(x_i)} = \frac{1}{m} \sum_{i=1}^{m} e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_j h_j(x_i)}. 
\]
• **Direction**: unit vector $e_k$ with best directional derivative:

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

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

$$F'(\bar{\alpha}_{t-1}, e_k) = -\frac{1}{m} \sum_{i=1}^{m} y_i h_k(x_i) e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i)}$$

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

$$= -\left[ \sum_{i=1}^{m} \bar{D}_t(i) 1_{y_i h_k(x_i) = +1} - \sum_{i=1}^{m} \bar{D}_t(i) 1_{y_i h_k(x_i) = -1} \right] \bar{Z}_t \frac{1}{m}$$

$$= -\left[ (1 - \bar{\epsilon}_{t,k}) - \bar{\epsilon}_{t,k} \right] \bar{Z}_t \frac{1}{m} = \frac{2\bar{\epsilon}_{t,k} - 1}{m} \bar{Z}_t.$$ 

Thus, direction corresponding to base classifier with smallest error.
Step size: $\eta$ chosen to minimize $F(\alpha_{t-1} + \eta e_k)$;

$$\frac{dF(\alpha_{t-1} + \eta e_k)}{d\eta} = 0 \iff -\sum_{i=1}^{m} y_i h_k(x_i) e^{-y_i} \sum_{j=1}^{N} \alpha_{t-1,j} h_j(x_i) e^{-\eta y_i h_k(x_i)} = 0$$

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

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

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

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

Thus, step size matches base classifier weight of AdaBoost.
Alternative Loss Functions

- **boosting loss**: $x \mapsto e^{-x}$
- **square loss**: $x \mapsto (1 - x)^2 \mathbb{1}_{x \leq 1}$
- **logistic loss**: $x \mapsto \log_2 (1 + e^{-x})$
- **hinge loss**: $x \mapsto \max(1 - x, 0)$
- **zero-one loss**: $x \mapsto \mathbb{1}_{x < 0}$
Standard Use in Practice

- **Base learners**: decision trees, quite often just decision stumps (trees of depth one).

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

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

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

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

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

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

**Empirical Observations**

C4.5 decision trees (Schapire et al., 1998).
Rademacher Complexity of Convex Hulls

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

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

Then, for any sample $S$, $\hat{\mathcal{R}}_S(\text{conv}(H)) = \hat{\mathcal{R}}_S(H)$.

**Proof**: $\hat{\mathcal{R}}_S(\text{conv}(H)) = \frac{1}{m} \mathbb{E}_{\sigma} \left[ \sup_{h_k \in H, \mu \geq 0, ||\mu||_1 \leq 1} \sum_{i=1}^{m} \sigma_i \sum_{k=1}^{p} \mu_k h_k(x_i) \right]$

$= \frac{1}{m} \mathbb{E}_{\sigma} \left[ \sup_{h_k \in H} \sup_{\mu \geq 0, ||\mu||_1 \leq 1} \sum_{k=1}^{p} \mu_k \left( \sum_{i=1}^{m} \sigma_i h_k(x_i) \right) \right]$

$= \frac{1}{m} \mathbb{E}_{\sigma} \left[ \sup_{h_k \in H} \max_{k \in [1,p]} \left( \sum_{i=1}^{m} \sigma_i h_k(x_i) \right) \right]$

$= \frac{1}{m} \mathbb{E}_{\sigma} \left[ \sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i) \right] = \hat{\mathcal{R}}_S(H)$. 
Margin Bound - Ensemble Methods

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

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

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

Proof: Direct consequence of margin bound of Lecture 4 and $\mathcal{K}_S(\text{conv}(H)) = \mathcal{K}_S(H)$. (Koltchinskii and Panchenko, 2002)
Margin Bound - Ensemble Methods

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

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

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

Proof: Follows directly previous corollary and VC dimension bound on Rademacher complexity (see lecture 3).
All of these bounds can be generalized to hold uniformly for all $\rho \in (0, 1)$, at the cost of an additional term $\sqrt{\frac{\log \log_2 \frac{2}{\rho}}{m}}$ and other minor constant factor changes (Koltchinskii and Panchenko, 2002).

For AdaBoost, the bound applies to the functions

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

Note that $T$ does not appear in the bound.
**Margin Distribution**

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

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

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

  $$\frac{1}{m} \sum_{i=1}^{m} 1_{y_i f(x_i) - \|\alpha\|_1 \rho \leq 0} \leq \frac{1}{m} \sum_{i=1}^{m} \exp(-y_i f(x_i) + \|\alpha\|_1 \rho)$$

  $$= \frac{1}{m} \sum_{i=1}^{m} e^{\|\alpha\|_1 \rho} \left[ m \prod_{t=1}^{T} Z_t \right] D_{T+1}(i)$$

  $$= e^{\|\alpha\|_1 \rho} \prod_{t=1}^{T} Z_t = 2^T \prod_{t=1}^{T} \left[ \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \right]^\rho \sqrt{\epsilon_t (1 - \epsilon_t)}.$$
Notes

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

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

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

For the bound to be convergent: \( \rho \gg O(1/\sqrt{m}) \), thus \( \gamma \gg O(1/\sqrt{m}) \) is roughly the condition on the edge value.
L1-Geometric Margin

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

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

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

$$
\rho_f = \min_{i \in [1,m]} \rho_f(x_i) = \min_{i \in [1,m]} \frac{\|\alpha \cdot h(x_i)\|_1}{\|\alpha\|_1}.
$$
# SVM vs AdaBoost

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>AdaBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>features or base hypotheses</strong></td>
<td>( \Phi(x) = \begin{bmatrix} \Phi_1(x) \ \vdots \ \Phi_N(x) \end{bmatrix} )</td>
<td>( h(x) = \begin{bmatrix} h_1(x) \ \vdots \ h_N(x) \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>predictor</strong></td>
<td>( x \mapsto w \cdot \Phi(x) )</td>
<td>( x \mapsto \alpha \cdot h(x) )</td>
</tr>
<tr>
<td><strong>geom. margin</strong></td>
<td>( \frac{</td>
<td>w \cdot \Phi(x)</td>
</tr>
<tr>
<td><strong>conf. margin</strong></td>
<td>( y(w \cdot \Phi(x)) )</td>
<td>( y(\alpha \cdot h(x)) )</td>
</tr>
<tr>
<td><strong>regularization</strong></td>
<td>( |w|_2 )</td>
<td>( |\alpha|_1 (\text{L1-AB}) )</td>
</tr>
</tbody>
</table>
Maximum-Margin Solutions

Norm $\| \cdot \|_2$.

Norm $\| \cdot \|_\infty$.
But, Does AdaBoost Maximize the Margin?

- **No:** AdaBoost may converge to a margin that is significantly below the maximum margin (Rudin et al., 2004) (e.g., 1/3 instead of 3/8)!

- **Lower bound:** AdaBoost can achieve asymptotically a margin that is at least $\frac{\rho_{\text{max}}}{2}$ if the data is separable and some conditions on the base learners hold (Rätsch and Warmuth, 2002).

- Several boosting-type margin-maximization algorithms: but, performance in practice not clear or not reported.
AdaBoost’s Weak Learning Condition

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

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

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

  $$\gamma(t) \geq \gamma.$$
Zero-Sum Games

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

Example:

<table>
<thead>
<tr>
<th></th>
<th>rock</th>
<th>paper</th>
<th>scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>rock</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>paper</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>scissors</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Mixed Strategies

**Definition:** player row selects a distribution \( p \) over the rows, player column a distribution \( q \) over columns. The expected payoff for row is

\[
E_{i \sim p, j \sim q} [M_{ij}] = \sum_{i=1}^{m} \sum_{j=1}^{n} p_i M_{ij} q_j = p^\top M q.
\]

**von Neumann’s minimax theorem:**

\[
\max_p \min_q p^\top M q = \min_q \max_p p^\top M q.
\]

**equivalent form:**

\[
\max_p \min_{j \in [1,n]} p^\top M e_j = \min_q \max_{i \in [1,m]} e_i^\top M q.
\]
John von Neumann (1903 - 1957)
AdaBoost and Game Theory

- **Game:**
  - Player A: selects point $x_i$, $i \in [1, m]$.
  - Player B: selects base hypothesis $h_t$, $t \in [1, T]$.
  - Payoff matrix $M \in \{-1, +1\}^{m \times T}$: $M_{it} = y_i h_t(x_i)$.

- **von Neumann's theorem:** assume finite $H$.

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

- **Weak learning condition** $\implies$ non-zero margin.
  - thus, possible to search for non-zero margin.
  - AdaBoost = (suboptimal) search for corresponding $\alpha$; achieves at least half of the maximum margin.

- **Weak learning = strong condition:**
  - the condition implies linear separability with margin $2\gamma^* > 0$. 
Maximizing the margin:

\[
\rho = \max_{\alpha} \min_{i \in [1,m]} y_i \frac{(\alpha \cdot x_i)}{||\alpha||_1}.
\]

This is equivalent to the following convex optimization LP problem:

\[
\begin{align*}
\max_{\alpha} & \quad \rho \\
\text{subject to} & \quad y_i (\alpha \cdot x_i) \geq \rho \\
& \quad ||\alpha||_1 = 1.
\end{align*}
\]

Note that:

\[
\frac{|\alpha \cdot x|}{||\alpha||_1} = ||x - H||_\infty, \text{ with } H = \{x : \alpha \cdot x = 0\}.
\]
Advantages of AdaBoost

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

- AdaBoost assigns larger weights to harder examples.

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

- **Parameters:**
  - need to determine $T$, the number of rounds of boosting: stopping criterion.
  - need to determine base learners: risk of overfitting or low margins.

- **Noise:** severely damages the accuracy of Adaboost (Dietterich, 2000).
Other Boosting Algorithms

- **arc-gv** (Breiman, 1996): designed to maximize the margin, but outperformed by AdaBoost in experiments (Reyzin and Schapire, 2006).

- **L1-regularized AdaBoost** (Raetsch et al., 2001): outperforms AdaBoost in experiments (Cortes et al., 2014).

- **DeepBoost** (Cortes et al., 2014): more favorable learning guarantees, outperforms both AdaBoost and L1-regularized AdaBoost in experiments.
References


References


