Advanced Machine Learning
Deep Ensemble Methods
Outline

- Model selection.
- Deep boosting.
  - theory.
  - algorithm.
  - experiments.
Estimation and Approximation

- **General equality:** for any $h \in H$,

$$R(h) - R^* = [R(h) - R(h^*)] + [R(h^*) - R^*].$$

- **Approximation:** not a random variable, only depends on $H$.

- **Estimation:** only term we can hope to bound; for ERM, bounded by two times gen. bound:

$$R(h_{ERM}) - R(h^*) = R(h_{ERM}) - \hat{R}(h_{ERM}) + \hat{R}(h_{ERM}) - R(h^*)$$
$$\leq R(h_{ERM}) - \hat{R}(h_{ERM}) + \hat{R}(h^*) - R(h^*)$$
$$\leq 2 \sup_{h \in H} |R(h) - \hat{R}(h)|.$$
Model Selection

Problem: how to select hypothesis set $H$?

- $H$ too complex, no gen. bound, overfitting.
- $H$ too simple, gen. bound, but underfitting.

$\rightarrow$ balance between estimation and approx. errors.
Structural Risk Minimization

(Vapnik and Chervonenkis, 1974; Vapnik, 1995)

- **SRM:** \( H = \bigcup_{k=1}^{\infty} H_k \) with \( H_1 \subset H_2 \subset \cdots \subset H_k \subset \cdots \)

- solution: \( f^* = \arg \min_{h \in H_k, k \geq 1} \widehat{R}_S(h) + \text{pen}(k, m). \)
SRM Guarantee

**Definitions:**

- $H_k(h)$ simplest hypothesis set containing $h$.
- $f^*$ the hypothesis returned by SRM:
  \[
  f^* = \arg\min_{h \in H_k, k \geq 1} \hat{R}_S(h) + R_m(H_k) + \sqrt{\frac{\log k}{m}} = F_k(h).
  \]

**Theorem:** for any $\delta > 0$, with probability at least $1 - \delta$,

\[
R(f^*) \leq R(h^*) + 2\mathfrak{R}_m(H_k(h^*)) + \sqrt{\frac{\log k(h^*)}{m}} + \sqrt{\frac{2 \log \frac{3}{\delta}}{m}}.
\]
Proof

General bound for all $h \in H$:

$$
\Pr \left[ \sup_{h \in H} R(h) - F_k(h) > \epsilon \right]
= \Pr \left[ \sup_{k \geq 1} \sup_{h \in H_k} R(h) - F_k(h) > \epsilon \right]
\leq \sum_{k=1}^{\infty} \Pr \left[ \sup_{h \in H_k} R(h) - F_k(h) > \epsilon \right]
= \sum_{k=1}^{\infty} \Pr \left[ \sup_{h \in H_k} R(h) - \hat{R}_S(h) - \mathcal{R}_m(H_k) > \epsilon + \sqrt{\frac{\log k}{m}} \right]
\leq \sum_{k=1}^{\infty} \exp \left( -2m \left[ \epsilon + \sqrt{\frac{\log k}{m}} \right]^2 \right)
\leq \sum_{k=1}^{\infty} e^{-2m\epsilon^2} e^{-2 \log k}
= e^{-2m\epsilon^2} \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} e^{-2m\epsilon^2} \leq 2e^{-2m\epsilon^2}.
$$
Proof

Using the union bound and the bound just derived gives:

\[
\Pr \left[ R(f^*) - R(h^*) - 2\mathfrak{R}_m(H_k(h^*)) - \sqrt{\frac{\log k(h^*)}{m}} > \epsilon \right]
\]

\[
\leq \Pr \left[ R(f^*) - F_k(f^*) > \frac{\epsilon}{2} \right] + \Pr \left[ F_k(f^*)(f^*) - R(h^*) - 2\mathfrak{R}_m(H_k(h^*)) - \sqrt{\frac{\log k(h^*)}{m}} > \frac{\epsilon}{2} \right]
\]

\[
\leq 2e^{-\frac{m\epsilon^2}{2}} + \Pr \left[ F_k(h^*)(h^*) - R(h^*) - 2\mathfrak{R}_m(H_k(h^*)) - \sqrt{\frac{\log k(h^*)}{m}} > \frac{\epsilon}{2} \right]
\]

\[
= 2e^{-\frac{m\epsilon^2}{2}} + \Pr \left[ \hat{R}_S(h^*) - R(h^*) - \mathfrak{R}_m(H_k(h^*)) > \frac{\epsilon}{2} \right]
\]

\[
= 2e^{-\frac{m\epsilon^2}{2}} + e^{-\frac{m\epsilon^2}{2}} = 3e^{-\frac{m\epsilon^2}{2}}.
\]
Remarks

- **SRM bound:**
  - similar to learning bound when $k(h^*)$ is known!
  - can be extended if approximation error assumed to be small or zero.
  - if $H$ contains the Bayes classifier, only finitely many hypothesis sets need to be considered in practice.
  - restriction: $H$ decomposed as countable union of families with converging Rademacher complexity.

- **Issues:** (1) SRM typically computationally intractable; (2) how should we choose $H_k$s?
Voted Risk Minimization

- **Ideas:**
  - no selection of specific $H_k$.
  - instead, use all $H_k$s: $h = \sum_{k=1}^{p} \alpha_k h_k$, $h_k \in H_k, \alpha \in \Delta$.
  - hypothesis-dependent penalty:
    $$\sum_{k=1}^{p} \alpha_k \mathcal{R}_m(H_k).$$

→ Deep ensembles.
Outline

- Model selection.
- Deep boosting.
  - theory.
  - algorithm.
  - experiments.
Deep Boosting Essence
Ensemble Methods in ML

Combining several base classifiers to create a more accurate one.

- Bagging (Breiman 1996).
- AdaBoost (Freund and Schapire 1997).
- Stacking (Smyth and Wolpert 1999).
- Bayesian averaging (MacKay 1996).
- Other averaging schemes e.g., (Freund et al. 2004).

Often very effective in practice.

Benefit of favorable learning guarantees.
Convex Combinations

- Base classifier set $H$.
  - boosting stumps.
  - decision trees with limited depth or number of leaves.
- Ensemble combinations: convex hull of base classifier set.

$$\text{conv}(H) = \left\{ \sum_{t=1}^{T} \alpha_t h_t : \alpha_t \geq 0; \sum_{t=1}^{T} \alpha_t \leq 1; \forall t, h_t \in H \right\}.$$
Ensembles - Margin Bound

(Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002)

**Theorem**: Let $H$ be a family of real-valued functions. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f = \sum_{t=1}^{T} \alpha_t h_t \in \text{conv}(H)$:

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

where $\hat{R}_{S, \rho}(f) = \frac{1}{m} \sum_{i=1}^{m} 1_{y_i f(x_i) \leq \rho}$. 
Questions

- Can we use a much richer or deeper base classifier set?
  - richer families needed for difficult tasks in speech and image processing.
  - but generalization bound indicates risk of overfitting.
AdaBoost

(Freund and Schapire, 1997)

- **Description:** coordinate descent applied to

\[
F'(\alpha) = \sum_{i=1}^{m} e^{-y_i f(x_i)} = \sum_{i=1}^{m} \exp \left( -y_i \sum_{t=1}^{T} \alpha_t h_t(x_i) \right).
\]

- **Guarantees:** ensemble margin bound.

  - but AdaBoost does not maximize the margin!

  - some margin maximizing algorithms such as arc-gv are outperformed by AdaBoost! (Reyzin and Schapire, 2006)
Suspicions

- Complexity of hypotheses used:
  - arc-gv tends to use deeper decision trees to achieve a larger margin.

- Notion of margin:
  - minimal margin perhaps not the appropriate notion.
  - margin distribution is key.

  can we shed more light on these questions?
Question

Main question: how can we design ensemble algorithms that can succeed even with very deep decision trees or other complex sets?

- theory.
- algorithms.
- experimental results.
Base Classifier Set $\mathcal{H}$

- Decomposition in terms of sub-families or their union.
Ensemble Family

- Non-negative linear ensembles $\mathcal{F} = \text{conv}(\bigcup_{k=1}^{p} H_k)$:

$$f = \sum_{t=1}^{T} \alpha_t h_t$$

with $\alpha_t \geq 0$, $\sum_{t=1}^{T} \alpha_t \leq 1$, $h_t \in H_{k_t}$. 
**Ideas**

- Use hypotheses drawn from $H_k$ with larger $k$s but allocate more weight to hypotheses drawn from smaller $k$s.
  - how can we determine quantitatively the amounts of mixture weights apportioned to different families?
  - can we provide learning guarantees guiding these choices?
Learning Guarantee

**Theorem:** Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{F}$:

$$R(f) \leq \hat{R}_{S,\rho}(f) + \frac{4}{\rho} \sum_{t=1}^{T} \alpha_t \mathcal{R}_m(H_{k_t}) + \tilde{O} \left( \sqrt{\frac{\log p}{\rho^2 m}} \right).$$

(Cortes, MM, and Syed, 2014)
Consequences

- Complexity term with explicit dependency on mixture weights.
  - quantitative guide for controlling weights assigned to more complex sub-families.
  - bound can be used to inspire, or directly define an ensemble algorithm.
Set-Up

- $H_1, \ldots, H_p$: disjoint sub-families of functions taking values in $[-1, +1]$.

- Further assumption (not necessary): symmetric sub-families, i.e. $h \in H_k \iff -h \in H_k$.

- Notation:
  - $r_j = \mathcal{R}_m(H_{k_j})$ with $h_j \in H_{k_j}$.
Learning bound suggests seeking $\alpha \geq 0$ with $\sum_{t=1}^{T} \alpha_t \leq 1$ to minimize

$$\frac{1}{m} \sum_{i=1}^{m} y_i \sum_{t=1}^{T} \alpha_t h_t(x_i) \leq \rho + \frac{4}{\rho} \sum_{t=1}^{T} \alpha_t r_t.$$
Convex Surrogates

Let $u \mapsto \Phi(-u)$ be a decreasing convex function upper bounding $u \mapsto 1_{u \leq 0}$, with $\Phi$ differentiable.

Two principal choices:

- Exponential loss: $\Phi(-u) = \exp(-u)$.
- Logistic loss: $\Phi(-u) = \log_2(1 + \exp(-u))$. 
Optimization Problem

(Cortes, MM, and Syed, 2014)

Moving the constraint to the objective and using the fact that the sub-families are symmetric leads to:

\[
\min_{\alpha \in \mathbb{R}^N} \frac{1}{m} \sum_{i=1}^{m} \Phi \left( 1 - y_i \sum_{j=1}^{N} \alpha_j h_j(x_i) \right) + \sum_{t=1}^{N} (\lambda r_j + \beta) |\alpha_j|,
\]

where \(\lambda, \beta \geq 0\), and for each hypothesis, keep either \(h\) or \(-h\).
DeepBoost Algorithm

- Coordinate descent applied to convex objective.
  - non-differentiable function.
  - definition of maximum coordinate descent.
Direction & Step

- Maximum direction: definition based on the error

$$
\epsilon_{t,j} = \frac{1}{2} \left[ 1 - \mathbb{E}_{i \sim D_t} [y_i h_j(x_i)] \right],
$$

where $D_t$ is the distribution over sample at iteration $t$.

- Step:
  - closed-form expressions for exponential and logistic losses.
  - general case: line search.
**Pseudocode**

```
DEEPBOOST(S = ((x₁, y₁),..., (xₘ, yₘ)))
  1. for i ← 1 to m do
  2.     D₁(i) ← \frac{1}{m}
  3. for t ← 1 to T do
  4.     for j ← 1 to N do
  5.       if (\alpha_{t-1,j} \neq 0) then
  6.           d_j ← (\epsilon_{t,j} - \frac{1}{2}) + \text{sgn}(\alpha_{t-1,j}) \frac{\Lambda_{j,m}}{2S_t}
  7.       elseif (|\epsilon_{t,j} - \frac{1}{2}| \leq \frac{\Lambda_{j,m}}{2S_t}) then
  8.           d_j ← 0
  9.       else d_j ← (\epsilon_{t,j} - \frac{1}{2}) - \text{sgn}(\epsilon_{t,j} - \frac{1}{2}) \frac{\Lambda_{j,m}}{2S_t}
 10.     k ← \arg\max_{j\in[1,N]} |d_j|
 11.     \epsilon_t ← \epsilon_{t,k}
 12.     if \left( |(1 - \epsilon_t)e^{\alpha_{t-1,k}} - \epsilon_t e^{-\alpha_{t-1,k}}| \leq \frac{\Lambda_{k,m}}{S_t} \right) then
 13.       \eta_t ← -\alpha_{t-1,k}
 14.     elseif \left( |(1 - \epsilon_t)e^{\alpha_{t-1,k}} - \epsilon_t e^{-\alpha_{t-1,k}}| > \frac{\Lambda_{k,m}}{S_t} \right) then
 15.       \eta_t ← \log \left( -\frac{\Lambda_{k,m}}{2\epsilon_t S_t} + \sqrt{\left( \frac{\Lambda_{k,m}}{2\epsilon_t S_t} \right)^2 + 1 - \epsilon_t} \right)
 16.     else \eta_t ← \log \left( +\frac{\Lambda_{k,m}}{2\epsilon_t S_t} + \sqrt{\left( \frac{\Lambda_{k,m}}{2\epsilon_t S_t} \right)^2 + 1 - \epsilon_t} \right)
 17.     \alpha_t ← \alpha_{t-1} + \eta_t e_k
 18.     S_{t+1} ← \sum_{i=1}^{m} \Phi' \left( 1 - y_i \sum_{j=1}^{N} \alpha_{t,j} h_j(x_i) \right)
 19.     for i ← 1 to m do
 20.     \quad D_{t+1}(i) ← \Phi' \left( 1 - y_i \sum_{j=1}^{N} \alpha_{t,j} h_j(x_i) \right)
 21.     f ← \sum_{j=1}^{N} \alpha_{t,j} h_j
 22. return f
```

\[ \Lambda_j = \lambda r_j + \beta. \]
Connections with Previous Work

For $\lambda = \beta = 0$, DeepBoost coincides with

- AdaBoost (Freund and Schapire 1997), run with union of sub-families, for the exponential loss.

- additive Logistic Regression (Friedman et al., 1998), run with union of sub-families, for the logistic loss.

For $\lambda = 0$ and $\beta \neq 0$, DeepBoost coincides with

- L1-regularized AdaBoost (Raetsch, Mika, and Warmuth 2001), for the exponential loss.

- coincides with L1-regularized Logistic Regression (Duchi and Singer 2009), for the logistic loss.
Rad. Complexity Estimates

- Benefit of data-dependent analysis:
  - empirical estimates of each $\mathcal{R}_m(H_k)$.
  - example: for kernel function $K_k$,
    \[ \hat{\mathcal{R}}_S(H_k) \leq \frac{\sqrt{\text{Tr}[K_k]}}{m}. \]
  - alternatively, upper bounds in terms of growth functions,
    \[ \mathcal{R}_m(H_k) \leq \sqrt{\frac{2 \log \Pi_{H_k}(m)}{m}}. \]
Experiments (1)

- Family of base classifiers defined by boosting stumps:
  - boosting stumps $H^{stumps}_1$ (threshold functions).
  - in dimension $d$, $\prod_{H^{stumps}_1}(m) \leq 2md$, thus
    $$\mathcal{R}_m(H^{stumps}_1) \leq \sqrt{\frac{2 \log(2md)}{m}}.$$  
  - decision trees of depth 2, $H^{stumps}_2$, with the same question at the internal nodes of depth 1.
    - in dimension $d$, $\prod_{H^{stumps}_2}(m) \leq (2m)^2 \frac{d(d-1)}{2}$, thus
      $$\mathcal{R}_m(H^{stumps}_2) \leq \sqrt{\frac{2 \log(2m^2d(d-1))}{m}}.$$
Experiments (1)

- Base classifier set: $H_1^{\text{stumps}} \cup H_2^{\text{stumps}}$.

- Data sets:
  - same UCI Irvine data sets as (Breiman 1999) and (Reyzin and Schapire 2006).
  - OCR data sets used by (Reyzin and Schapire 2006): ocr17, ocr49.
  - MNIST data sets: ocr17-mnist, ocr49-mnist.

- Experiments with exponential loss.

## Data Statistics

<table>
<thead>
<tr>
<th></th>
<th>breastcancer</th>
<th>ionosphere</th>
<th>german (numeric)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>699</td>
<td>351</td>
<td>1000</td>
</tr>
<tr>
<td>Attributes</td>
<td>9</td>
<td>34</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>diabetes</th>
<th>ocr17</th>
<th>ocr49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>768</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>Attributes</td>
<td>8</td>
<td>196</td>
<td>196</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ocr17-mnist</th>
<th>ocr49-mnist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>15170</td>
<td>13782</td>
</tr>
<tr>
<td>Attributes</td>
<td>400</td>
<td>400</td>
</tr>
</tbody>
</table>
Experiments - Stumps Exp Loss

(Cortes, MM, and Syed, 2014)

**Table 1. Results for boosted decision stumps and the exponential loss function.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AdaBoost $H_1$ stumps</th>
<th>AdaBoost $H_2$ stumps</th>
<th>AdaBoost-L1</th>
<th>DeepBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error</td>
<td>Error (std dev)</td>
<td>Error</td>
<td>Error (std dev)</td>
</tr>
<tr>
<td></td>
<td>(std dev)</td>
<td></td>
<td>(std dev)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg tree size</td>
<td>Avg tree size</td>
<td>Avg tree size</td>
<td>Avg tree size</td>
</tr>
<tr>
<td></td>
<td>Avg no. of trees</td>
<td>Avg no. of trees</td>
<td>Avg no. of trees</td>
<td>Avg no. of trees</td>
</tr>
<tr>
<td>breastcancer</td>
<td>0.0429 (0.0248)</td>
<td>0.0437 (0.0214)</td>
<td>0.0408 (0.0223)</td>
<td>0.0373 (0.0225)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.436</td>
<td>1.215</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>43.6</td>
<td>21.6</td>
</tr>
<tr>
<td>ionosphere</td>
<td>0.1014 (0.0414)</td>
<td>0.075 (0.0413)</td>
<td>0.0708 (0.0331)</td>
<td>0.0638 (0.0394)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.392</td>
<td>1.168</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>39.35</td>
<td>17.45</td>
</tr>
<tr>
<td>german</td>
<td>0.243 (0.0445)</td>
<td>0.2505 (0.0487)</td>
<td>0.2455 (0.0438)</td>
<td>0.2395 (0.0462)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.54</td>
<td>1.76</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>54.1</td>
<td>76.5</td>
</tr>
<tr>
<td>diabetes</td>
<td>0.253 (0.0330)</td>
<td>0.260 (0.0518)</td>
<td>0.254 (0.0486)</td>
<td>0.253 (0.0510)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.9975</td>
<td>1.9975</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ocr17</td>
<td>0.0085</td>
<td>0.0072</td>
<td>0.0075</td>
<td>0.0070</td>
</tr>
<tr>
<td></td>
<td>(0.0048)</td>
<td></td>
<td>(0.0068)</td>
<td>(0.0048)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.086</td>
<td>1.369</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>37.8</td>
<td>36.9</td>
</tr>
<tr>
<td>ocr49</td>
<td>0.0555</td>
<td>0.0167</td>
<td>0.0122</td>
<td>0.0275</td>
</tr>
<tr>
<td></td>
<td>(0.0095)</td>
<td></td>
<td>(0.0095)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.99</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>99.3</td>
<td>96</td>
</tr>
<tr>
<td>ocr17-mnist</td>
<td>0.0056</td>
<td>0.0017</td>
<td>0.0046</td>
<td>0.0040</td>
</tr>
<tr>
<td></td>
<td>(0.0014)</td>
<td></td>
<td>(0.0013)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1.99</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ocr49-mnist</td>
<td>0.0414</td>
<td>0.0209</td>
<td>0.0200</td>
<td>0.0177</td>
</tr>
<tr>
<td></td>
<td>(0.00438)</td>
<td></td>
<td>(0.00438)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1.9975</td>
<td>1.9975</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Experiments (2)

- Family of base classifiers defined by decision trees of depth $k$. For trees with at most $n$ nodes:

$$ R_m(T_n) \leq \sqrt{\frac{(4n + 2) \log_2(d + 2) \log(m + 1)}{m}}. $$

- Base classifier set: $\bigcup_{k=1}^{K} H_{trees}^k$.

- Same data sets as with Experiments (1).

- Both exponential and logistic loss.

## Experiments - Trees Exp Loss

*(Cortes, MM, and Syed, 2014)*

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Error</th>
<th>(std dev)</th>
<th>Avg tree size</th>
<th>Avg no. of trees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.0267</td>
<td>(0.00841)</td>
<td>29.1</td>
<td>67.1</td>
</tr>
<tr>
<td>breastcancer</td>
<td>AdaBoost-L1</td>
<td>0.0264</td>
<td>(0.0098)</td>
<td>28.9</td>
<td>51.7</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.0243</strong></td>
<td><strong>(0.00797)</strong></td>
<td>20.9</td>
<td>55.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.004</td>
<td>(0.00316)</td>
<td>15.0</td>
<td>88.3</td>
</tr>
<tr>
<td>ocr17</td>
<td>AdaBoost-L1</td>
<td>0.003</td>
<td>(0.00100)</td>
<td>30.4</td>
<td>65.3</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.002</strong></td>
<td>*(0.00100)**</td>
<td>26.0</td>
<td>61.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.0661</td>
<td>(0.0315)</td>
<td>29.8</td>
<td>75.0</td>
</tr>
<tr>
<td>ionosphere</td>
<td>AdaBoost-L1</td>
<td>0.0657</td>
<td>(0.0257)</td>
<td>31.4</td>
<td>69.4</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.0501</strong></td>
<td>*(0.0316)**</td>
<td>26.1</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.0180</td>
<td>(0.00555)</td>
<td>30.9</td>
<td>92.4</td>
</tr>
<tr>
<td>german</td>
<td>AdaBoost-L1</td>
<td>0.0175</td>
<td>(0.00357)</td>
<td>62.1</td>
<td>89.0</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.0175</strong></td>
<td>*(0.00510)**</td>
<td>30.2</td>
<td>83.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.239</td>
<td>(0.0165)</td>
<td>3</td>
<td>91.3</td>
</tr>
<tr>
<td>diabetes</td>
<td>AdaBoost-L1</td>
<td>0.239</td>
<td>(0.0201)</td>
<td>7</td>
<td>87.5</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.234</strong></td>
<td>*(0.0148)**</td>
<td>16.0</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.00471</td>
<td>(0.0022)</td>
<td>15</td>
<td>88.7</td>
</tr>
<tr>
<td>ocr17-mnist</td>
<td>AdaBoost-L1</td>
<td>0.00471</td>
<td>(0.0021)</td>
<td>33.4</td>
<td>66.8</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.00409</strong></td>
<td>*(0.0021)**</td>
<td>22.1</td>
<td>59.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.249</td>
<td>(0.0272)</td>
<td>3</td>
<td>45.2</td>
</tr>
<tr>
<td>diabetes</td>
<td>AdaBoost-L1</td>
<td>0.240</td>
<td>(0.0313)</td>
<td>3</td>
<td>28.0</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.230</strong></td>
<td>*(0.0399)**</td>
<td>5.37</td>
<td>19.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AdaBoost</td>
<td>0.0198</td>
<td>(0.00500)</td>
<td>29.9</td>
<td>82.4</td>
</tr>
<tr>
<td>ocr49-mnist</td>
<td>AdaBoost-L1</td>
<td>0.0197</td>
<td>(0.00512)</td>
<td>66.3</td>
<td>81.1</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td><strong>0.0182</strong></td>
<td>*(0.00551)**</td>
<td>30.1</td>
<td>80.9</td>
</tr>
</tbody>
</table>
# Experiments - Trees Log Loss

(Cortes, MM, and Syed, 2014)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Error</th>
<th>std dev</th>
<th>Avg tree size</th>
<th>Avg no. of trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>breastcancer</td>
<td>LogReg</td>
<td>0.0351</td>
<td>0.0101</td>
<td>15</td>
<td>65.3</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.0264</td>
<td>0.0120</td>
<td>59.9</td>
<td>16.0</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.0264</td>
<td>0.00876</td>
<td>14.0</td>
<td>23.8</td>
</tr>
<tr>
<td>ocr17</td>
<td>LogReg</td>
<td>0.00300</td>
<td>0.00100</td>
<td>15.0</td>
<td>75.3</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.00400</td>
<td>0.00141</td>
<td>14.0</td>
<td>75.3</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.00250</td>
<td>0.000866</td>
<td>7</td>
<td>22.1</td>
</tr>
<tr>
<td>breastcancer</td>
<td>LogReg</td>
<td>0.0740</td>
<td>0.0236</td>
<td>7</td>
<td>44.7</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.0600</td>
<td>0.0219</td>
<td>30.0</td>
<td>25.3</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.0430</td>
<td>0.0188</td>
<td>18.4</td>
<td>29.5</td>
</tr>
<tr>
<td>ocr17</td>
<td>LogReg</td>
<td>0.0205</td>
<td>0.00654</td>
<td>31.0</td>
<td>63.5</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.0200</td>
<td>0.00245</td>
<td>31.0</td>
<td>54.0</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.0170</td>
<td>0.00361</td>
<td>63.2</td>
<td>37.0</td>
</tr>
<tr>
<td>german</td>
<td>LogReg</td>
<td>0.2330</td>
<td>0.0114</td>
<td>7</td>
<td>72.8</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.2320</td>
<td>0.0123</td>
<td>7</td>
<td>66.8</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.2250</td>
<td>0.0103</td>
<td>14.4</td>
<td>67.8</td>
</tr>
<tr>
<td>ocr17-mnist</td>
<td>LogReg</td>
<td>0.00422</td>
<td>0.00191</td>
<td>15</td>
<td>71.4</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.00417</td>
<td>0.00188</td>
<td>15</td>
<td>55.6</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.00399</td>
<td>0.00211</td>
<td>25.9</td>
<td>27.6</td>
</tr>
<tr>
<td>diabetes</td>
<td>LogReg</td>
<td>0.2500</td>
<td>0.0374</td>
<td>3</td>
<td>46.0</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.2460</td>
<td>0.0356</td>
<td>3</td>
<td>45.5</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.2460</td>
<td>0.0356</td>
<td>3</td>
<td>45.5</td>
</tr>
<tr>
<td>ocr49-mnist</td>
<td>LogReg</td>
<td>0.0211</td>
<td>0.00412</td>
<td>28.7</td>
<td>79.3</td>
</tr>
<tr>
<td></td>
<td>LogReg-L1</td>
<td>0.0201</td>
<td>0.00433</td>
<td>33.5</td>
<td>61.7</td>
</tr>
<tr>
<td></td>
<td>DeepBoost</td>
<td>0.0201</td>
<td>0.00411</td>
<td>72.8</td>
<td>41.9</td>
</tr>
</tbody>
</table>

(Cortes, MM, and Syed, 2014)
Margin Distribution

Ion: AdaBoost–L1, fold = 6

Ion: AdaBoost, fold = 6

Ion: DeepBoost, fold = 6

Cumulative Distribution of Margins

Normalized Margin

Normalized Margin

Normalized Margin

Normalized Margin
Multi-Class Learning Guarantee

**Theorem**: Fix \( \rho > 0 \). Then, for any \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for all \( f = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{F} \):

\[
R(f) \leq \widehat{R}_{S, \rho}(f) + \frac{8c}{\rho} \sum_{t=1}^{T} \alpha_t \mathcal{R}_m(\Pi_1(H_{k_t})) + O \left( \sqrt{\frac{\log p}{\rho^2 m}} \log \left[ \frac{\rho^2 c^2 m}{4 \log p} \right] \right).
\]

- with \( c \) number of classes.
- and \( \Pi_1(H_k) = \{ x \mapsto h(x, y) : y \in \mathcal{Y}, h \in H_k \} \).

(Kuznetsov, MM, and Syed, 2014)
Extension to Multi-Class

- Similar data-dependent learning guarantee proven for the multi-class setting.
  - bound depending on mixture weights and complexity of sub-families.

- Deep Boosting algorithm for multi-class:
  - similar extension taking into account the complexities of sub-families.
  - several variants depending on number of classes.
  - different possible loss functions for each variant.
Conclusion

- **Deep Boosting**: ensemble learning with increasingly complex families.
  - data-dependent theoretical analysis.
  - algorithm based on learning bound.
  - extension to multi-class.
  - ranking and other losses.
  - enhancement of many existing algorithms.
  - compares favorably to AdaBoost and additive Logistic Regression or their L1-regularized variants in experiments.
References


References


References

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

