Boosting

Richard Nock
NICTA

richard.nock@nicta.com.au
Outline

- The origins: the PAC model
- The boosting framework
- Boosting with many architectures
- Classification with many losses
- Boosting with many losses
- Boosting “beyond” the PAC requirements
What will be “omitted”

Two aspects will be left out intentionally:

- the regularisation approach to boosting, for which the links with the original theory are not always clear

\[ \mathbb{E}_S [\ell^r(yH_s(o))] = \mathbb{E}_S [\ell(yH_s(o))] + f(H_s) \]

- Boosting over multi-class problems, because various generalisations with various properties

\[ \mathbb{E}_S [\ell(yH_s(o))] \]

two values
THE ORIGINS
From SIAM News, Volume 46, Number 8, October 2013

PAC: A Universal Framework for Learning?


In 2010, Leslie Valiant received the Turing Award, the “Nobel prize of computer science.” The citation mentioned a number of contributions, the best known of which is the theory of probably approximately correct (PAC) learning, which Valiant first proposed in the early 1980s. In his new book Probably Approximately Correct, Valiant discusses the theory of PAC learning and its applications to artificial intelligence; he also introduces a more specialized concept—“evolvable learning”—which, he claims, characterizes Darwinian evolution.
PAC model of Valiant’84

- Domain of observations

\[ O = \{ \{ X, (B), 0 \} \} \]

- Class \( \in \{-1, +1\} \)
PAC model of Valiant’84

Example: \((o, y)\) - observation and class

\[
\begin{pmatrix}
X & 0 \\
X & X & X \\
0 & 0 \\
\end{pmatrix}, +1
\]
PAC model of Valiant’84

❖ Example: \((o, y)\) - observation and class

\[
\begin{pmatrix}
X & 0 \\
X & X \\
X & X \\
0 & 0 \\
\end{pmatrix}, +1
\]
**PAC model of Valiant’84**

- Classifier (=hypothesis): any function $\mathcal{O} \rightarrow \mathbb{R}$ computable in reasonable time (e.g. polynomial)

- For instance, let

$$\mathcal{O} = \left\{ \begin{array}{ccc}
    v_1 & v_2 & v_3 \\
    v_4 & v_5 & v_6 \\
    v_7 & v_8 & v_9
\end{array} \right\}$$

$d = 9$ is the dimension of domain
PAC model of Valiant’84

- **Monomial**: rule of the type “if - then - else”
  \[
  H_1(o) \doteq \text{if } (v_1 = X) \land (v_5 = X) \land (v_9 = X) \text{ then } +1 \text{ else } -1
  \]

- **Decision tree**: recursive partition of \( \mathcal{O} \)
  
  \[
  H_2(o) \doteq \begin{cases} 
  +1 & \text{if } v_1 = X \land v_5 = X \land v_9 = X \\
  -1 & \text{if } v_1 = X \land v_5 = X \land v_9 = X \\
  -1 & \text{if } v_1 = X \land v_5 = X \land v_9 = X
  \end{cases}
  \]

- **Linear** (combination of) **function(s)** = linear separators:
  
  \[
  H_3(o) \doteq \text{sign}(3H_1(o) - 2H_2(o))
  \]
PAC model of Valiant’84

- Branching program:
- Oblique decision tree:

Linear separators
Alternating decision trees

\[
\begin{align*}
H_1(o) &= \text{sign}(3H_1(o) - 2H_2(o)) \\
H_3(o) &= \text{if} (v_1 = X)^{(v_5 = X)}^{(v_9 = X)} \text{then } +1 \text{ else } -1
\end{align*}
\]
PAC model of Valiant’84

- Let $\mathcal{H} = \text{set of hypotheses having the same “representation formalism”}$
- $\text{DT} = \text{set of decision trees}$,
- $\text{LS} = \text{set of linear separators}$,
- etc.
PAC model of Valiant’84

- Let $H \in \mathcal{H}$ and $(o, y) \in \mathcal{O} \times \{-1, 1\}$

- **Loss** of $H$ on example $(o, y)$ measures “disagreement” between $H(o)$ and $y$

  Depends on fundamental parameter: **edge**, $yH(o)$

- **0/1 loss:**
  
  $\ell_{0/1}(y, H(o)) = 1_{yH(o)<0}$
Let $H \in \mathcal{H}$ and $(o, y) \in \mathcal{O} \times \{-1, 1\}$

**Loss** of $H$ on example $(o, y)$ measures “disagreement” between $H(o)$ and $y$
Depends on fundamental parameter: **edge**, $yH(o)$

**o/1 loss:**
\[
\ell_{0/1}(y, H(o)) = 1_{yH(o)<0}
\]

Just cares about sign
PAC model of Valiant’84

- One fixes the hypothesis class, \( \mathcal{H} \) and assumes that true labelling comes from some \( H_\ast \in \mathcal{H} \) on \( \mathcal{O} \)

\[
H_\ast(\cdot) = -1
\]

\[
H_\ast(\cdot) = +1
\]
PAC model of Valiant’84

- One fixes the hypothesis class, $\mathcal{H}$ and assumes that true labelling comes from some $\mathcal{H} \subseteq \Omega$. 

Unknown
PAC model of Valiant’84

- Sample i.i.d. examples from \( \mathcal{O} \times \{-1, +1\} \)
PAC model of Valiant’84

- Sample i.i.d. examples from \( \mathcal{O} \times \{-1, +1\} \)
PAC model of Valiant'84

- Sample i.i.d. examples from $\mathcal{O} \times \{-1, +1\}$
PAC model of Valiant’84

Sample i.i.d. examples from $\mathcal{O} \times \{ -1, +1 \}$ for $i$, (\begin{pmatrix} x & 0 \\ x & x \\ 0 & 0 \end{pmatrix}, +1) = (o_1, y_1)$
PAC model of Valiant’84

Sample i.i.d. examples from $\mathcal{O} \times \{-1, +1\}$ for $1, 2,$

\[
\begin{pmatrix}
X & 0 \\
X & X & X \\
0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
X & X & 0 \\
0 & X \\
0 & X & 0 \\
\end{pmatrix}
= (o_2, y_2)
\]
PAC model of Valiant'84

Sample i.i.d. examples from $\mathcal{O} \times \{-1, +1\}$ for $1, 2, ..., m$ ex.

$(\mathcal{D})$

$(o_m, y_m) = \begin{pmatrix} X & 0 \\ X & X & X & \end{pmatrix}, +1$

$(\mathcal{D})$

$(o_m, y_m) = \begin{pmatrix} X & X & 0 \\ 0 & X \\ 0 & X & 0 \end{pmatrix}, -1$

$(\mathcal{D})$

$(o_m, y_m) = \begin{pmatrix} X & X & X \\ 0 & 0 \\ X & 0 \end{pmatrix}, -1$
PAC model of Valiant’84

- One fixes the target class, $\mathcal{H}$
- $\mathcal{H}$ is PAC (Probably Approximately Correct) learnable iff there exists an algorithm $\mathcal{A}$ such that...
PAC model of Valiant’84

- $A$ has access to sampling $\mathcal{D}$ (for any, fixed but unknown),
A has access to sampling \( \mathcal{D} \) (unknown),

input:

\[ d = \dim(\mathcal{D}) \]

\[ 0 < \epsilon, \delta < 1 \]
PAC model of Valiant’84

- \( A \) has access to sampling \( \mathcal{D} \) (unknown), samples \( S \) examples,

\[
d = \text{dim}(\mathcal{D})
\]

\[
0 < \epsilon, \delta < 1
\]
PAC model of Valiant’84

- $A$ has access to sampling $\mathcal{D}$ (unknown), samples $\mathcal{S}$ examples,

$$d = \dim(\mathcal{D})$$
$$0 < \epsilon, \delta < 1$$

$$|\mathcal{S}| = \text{poly}(d, 1/\epsilon, 1/\delta)$$
A PAC model of Valiant’84

- A has access to sampling \( D \) (unknown), samples \( S \) examples, outputs \( H \in \mathcal{H} \),

\[
d = \text{dim}(D) \\
0 < \epsilon, \delta < 1
\]

\[|S| = \text{poly}(d, 1/\epsilon, 1/\delta)\]
PAC model of Valiant’84

- \( A \) has access to sampling \( \mathcal{D} \) (unknown), samples \( S \) examples, outputs \( H \in \mathcal{H} \), in polynomial time,

\[
d = \text{dim}(\mathcal{D})
\]

\[
0 < \varepsilon, \delta < 1
\]

\[
|S| = \text{poly}(d, 1/\varepsilon, 1/\delta)
\]
PAC model of Valiant’84

- $A$ has access to sampling $\mathcal{D}$ (unknown), samples $S$ examples, outputs $H \in \mathcal{H}$, in polynomial time, and such that $err(h) \leq \epsilon$

\[ d = \dim(\mathcal{D}) \]
\[ 0 < \epsilon, \delta < 1 \]

\[ |S| = \text{poly}(d, 1/\epsilon, 1/\delta) \]

\[ \mathbb{E}_\mathcal{D}[\ell^0/1(yH(o))] \leq \epsilon \]
A has access to sampling $\mathcal{D}$ (unknown), samples $\mathcal{S}$ examples, outputs $H \in \mathcal{H}$, in polynomial time, and such that $\text{err}(h) \leq \epsilon$

$$d = \dim(\mathcal{D})$$
$$0 < \epsilon, \delta < 1$$

$|\mathcal{S}| = \text{poly}(d, 1/\epsilon, 1/\delta)$

$\mathbb{E}_{\mathcal{D}}[\ell_{0/1}(yH(o))] \leq \epsilon$

with prob. \( \geq 1 - \delta \).
PAC model of Valiant’84

- PAC-learning = satisfying two constraints,
  - polynomial complexity (time, sample)
  - small true err with high probability
PAC model of Valiant’84

- PAC-learning = satisfying two constraints,
  - polynomial complexity (time, sample)
  - small true err with high probability
PAC model of Valiant’84

- Basic strategy:
  - Find **P-time algorithm** with reduced (e.g. 0) empirical risk
  - Sample **enough** examples

\[
\mathbb{E}_S[\ell^{0/1}(yH(o))] = 0
\]
PAC model of Valiant’84

- Basic strategy:
  - Find P-time algorithm with reduced (e.g. 0) empirical risk
  - Sample enough examples

\[ m = \Omega \left( \frac{1}{\varepsilon} \log \frac{1}{\delta} + \frac{\text{VC}(H)}{\varepsilon} \right) \]

\[ \mathbb{E}_{S}[\ell^{0/1}(y_H(o))] = 0 \]

Ehrenfeucht et al. 1989
PAC model of Valiant’84

Basic strategy:

- Find \textbf{P-time algorithm} with reduced (e.g. 0) empirical risk

\[ S \xrightarrow{A} H \]

- Sample \textbf{enough} examples

\[ m = \Omega \left( \frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{\text{VC}(H)}{\epsilon} \right) \]

Statistical = “easy”

\[ \mathbb{E}_S [\ell^0/1(yH(o))] = 0 \]
PAC model of Valiant’84

Problem: Computation

Most of the difficulties in proper PAC learning are due to the computational difficulty of finding a hypothesis in the particular form specified by the target class. For example, while Boolean

Haussler, 1990
Problem: Computation stuck in between easy ("trivial"), unknown and combinatorial hardness

- $\mathcal{H}_{\text{max}}$: Largest class
- $\mathcal{H}_{k,1<k<\text{max}}$: Many interesting classes
- $\mathcal{H}_1$: Very small class
PAC model of Valiant’84

Example: Feldman, 2007 (+ refs therein)

\[
\text{DNF} = \{ \lor \text{monomials} \} = 2^d\text{-term DNF}
\]

\[
\lor
\]

\[k\text{-term DNF} = \{ \lor_{\leq k} \text{monomials} \}
\]

\[
\lor
\]

monomials
WORKAROUND
Improper PAC-learning
Thoughts on Hypothesis Boosting.
Machine Learning class project, Dec. 1988
Michael Kearns

In this paper we present initial and modest progress on the Hypothesis Boosting Problem. Informally, this problem asks whether an efficient learning algorithm (in the distribution-free model of [V84]) that outputs an hypothesis whose performance is only slightly better than random guessing implies the existence of an efficient algorithm that outputs an hypothesis of arbitrary accuracy. The resolution of this question is of theoretical interest and possibly of practical importance. From the
Assumption: fix weak hypothesis class, $\mathcal{H}_w$ and assume the existence of an algorithm $A_w$ which is a PAC-learner, but “only” for $\epsilon = (1/2) - \gamma$, for some $\gamma = 1/poly(d, \ldots)$

$$\mathbb{E}_D[\ell^{0/1}(yH_w(o))] \leq \frac{1}{2} - \gamma$$
Remark: let us also weaken...

- ... the requirement on \( \delta \):
  \[
  \delta = \gamma
  \]

- ... and the (already) weak requirement on \( \epsilon \):
  \[
  \left| \mathbb{E}_D[\ell^{0/1}(yH_w(o))] - \frac{1}{2} \right| \geq \gamma
  \]
  (need additional assumption, e.g. \( H_w \) closed wrt negation)
Question: does this imply the existence of strong learner $A_s$?
Weak = Strong learning

- Answer: yes
- Not practical,

Schapire, 1990
Answer: yes

Schapire, 1990

Not practical, but hacks the general strong learner’s shape:
Answer: yes

Not practical, but hacks the general strong learner’s shape: use weak learner

\[ \text{Weak} = \text{Strong learning} \]
Answer: yes

Schapire, 1990

Not practical, but hacks the general strong learner’s shape: use weak learner

\[
\begin{align*}
&\mathcal{D} \quad \mathcal{S} \\
&\mathcal{A}_s \\
&\mathcal{A}_w \\
&H_w
\end{align*}
\]
Answer: yes  

Schapire, 1990

Not practical, but hacks the general strong learner’s shape: use iteratively weak learner
Answer: yes  
Schapire, 1990

Not practical, but hacks the general strong learner’s shape: use iteratively weak learner, output a function of all $H_w$

(in general, $\mathcal{H}_w \subseteq \mathcal{H}_s$)
Answer: yes

Not practical, but hacks the general strong learner’s shape: use iteratively weak learner, output a function of all $H_w$

... defines boosting as an iterated zero-sum game over edges
Key points:

- **All** classes mentioned above can be “boosted” using the **same** hi-level scheme.

- In practice, $|\mathcal{H}_w|$ small (e.g. $|\mathcal{H}_w| = O(d)$) so $\mathcal{A}_w$ is easy to implement (exhaustive search!)

- One can imagine nested weak / “moderate” / strong learners:

$$
\mathcal{H}_w \subseteq \mathcal{H}_{m_1} \subseteq \mathcal{H}_{m_2} \subseteq ... \subseteq \mathcal{H}_s
$$

$$
\mathcal{A}_w \quad \mathcal{A}_{s_1} \quad \mathcal{A}_{s_2} \quad ... \quad \mathcal{A}_s
$$
Key points:

- All classes mentioned above can be “boosted” using the same hi-level scheme.

- In practice, $|\mathcal{H}_w|$ small (e.g. $|\mathcal{H}_w| = O(d)$) so $\mathcal{A}_w$ is easy to implement (exhaustive search!)

- One can imagine nested weak / “moderate” / strong learners

  $\mathcal{A}_w = \{v_i, \forall i\}$

  ex: $\mathcal{A}_{s_1} = \text{DT} \Rightarrow \mathcal{H}_s = \text{Linear comb. of decision trees}$

  $\mathcal{A}_s = \text{LS}$
Key points:

- **All** classes mentioned above can be “boosted” using the **same** hi-level scheme.

- In practice, \(|\mathcal{H}_w|\) small (e.g. \(|\mathcal{H}_w| = O(d)\) ) so \(A_w\) is easy to implement (exhaustive search!)

- One can imagine nested weak / “moderate” / strong learners:
  \[
  A_w = \{v_i, \forall i\}
  \]
  ex: \(A_{s_1} = LS\) \(\Rightarrow \mathcal{H}_s = \text{Oblique decision trees}\)
  \(A_s = DT\)
AdaBoost in one (long) slide

- \( A_s : \) initialize \( w_1 = u \) and \( H_s = 0 \)
- Repeat for \( t = 1, 2, \ldots, T \)
  \[ h_t \leftarrow A_w(S, w_t) \]
  \[ \alpha_t \leftarrow \arg\min_{\alpha} Z_t(\alpha) \leftarrow E_{w_t}[\exp(-\alpha y h_t(x))] \]
  \[ w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x)) \]
- output \( H_s = \sum_t \alpha_t h_t \)
AdaBoost in one (long) slide

$A_s : \text{initialize } w_1 = u \text{ and } H_s = 0$

Repeat for $t = 1, 2, \ldots, T$

$h_t \leftarrow A_w(S, w_t)$

$\alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) = \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))]$

$w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x))$

output $H_s = \sum_t \alpha_t h_t$

$\exp(-y H_s(o)) = (\prod_t Z_t(\alpha_t)) w_1 = \prod_t Z_t(\alpha_t) u$

Freund & Schapire, 1997

Schapire & Singer, 1999
AdaBoost in one (long) slide

\[ A_s : \text{initialize } w_1 = u \text{ and } H_s = 0 \]
Repeat for \( t = 1, 2, \ldots, T \)

\[ h_t \leftarrow A_w(S, w_t) \]
\[ \alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) = \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))] \]
\[ w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x)) \]
output \( H_s = \sum_t \alpha_t h_t \)

\[ \exp(-y H_s(o)) = \prod_t Z_t(\alpha_t)w_1 = \prod_t Z_t(\alpha_t)u \]

\[ \ell^0/1(x) \leq \exp(-x) \]
AdaBoost in one (long) slide

\[ A_s : \text{initialize } w_1 = u \text{ and } H_s = 0 \]
Repeat for \( t = 1, 2, \ldots, T \)

\[ h_t \leftarrow A_w(S, w_t) \]
\[ \alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) = \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))] \]
\[ w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x)) \]
output \( H_s = \sum_t \alpha_t h_t \)

\[ \exp(-y H_s(o)) = (\prod_t Z_t(\alpha_t))(w_{t+1}/w_1) = (\prod_t Z_t(\alpha_t))(w_{t+1}/u) \]

\[ \mathbb{E}_S[\ell^{0/1}(y H(o))] \leq \mathbb{E}_S[\exp(-y H(o))] \leq \prod_t Z_t(\alpha_t) \]
AdaBoost in one (long) slide

\[ A_s : \text{initialize } \mathbf{w}_1 = \mathbf{u} \text{ and } H_s = 0 \]
Repeat for \( t = 1, 2, \ldots, T \)

\[ h_t \leftarrow A_w(S, \mathbf{w}_t) \]
\[ \alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) \doteq \mathbb{E}_{\mathbf{w}_t} [\exp (-\alpha y h_t(x))] \]
\[ \mathbf{w}_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \circ \exp (-\alpha_t y h_t(x)) \]
output \( H_s = \sum_t \alpha_t h_t \)

\[ \mathbb{E}_S[\ell^{0/1}(y H(o))] \leq \prod_t Z_t(\alpha_t) \]
**AdaBoost in one (long) slide**

- $A_s$: initialize $w_1 = u$ and $H_s = 0$
- Repeat for $t = 1, 2, ..., T$
  
  $h_t \leftarrow A_w(S, w_t)$
  
  $\alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) \doteq E_{w_t}[\exp(-\alpha y h_t(x))]$
  
  $w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha t y h_t(x))$
  
  output $H_s = \sum_t \alpha_t h_t$

$E_S[\ell^{0/1}(yH(o))] \leq \prod_t Z_t(\alpha_t)$

$$\alpha_t = \frac{1}{2} \log \frac{1+E_{w_t}[y h_t(o)]}{1-E_{w_t}[y h_t(o)]}$$
AdaBoost in one (long) slide

\[ \mathcal{A}_s: \text{initialize } \mathbf{w}_1 = \mathbf{u} \text{ and } H_s = 0 \]

Repeat for \( t = 1, 2, \ldots, T \)

\[ h_t \leftarrow \mathcal{A}_w(S, \mathbf{w}_t) \]

\[ \alpha_t \leftarrow \arg \min_\alpha Z_t(\alpha) \equiv \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))] \]

\[ \mathbf{w}_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \odot \exp(-\alpha_t y h_t(x)) \]

output \( H_s = \sum_t \alpha_t h_t \)

\[ \mathbb{E}_S[\ell^{0/1}(yH(o))] \leq \prod_t Z_t(\alpha_t) \]

\[ \alpha_t = \frac{1}{2} \log \frac{1 + \mathbb{E}_{w_t}[y h_t(o)]}{1 - \mathbb{E}_{w_t}[y h_t(o)]} \]

\[ Z_t(\alpha_t) \leq \frac{1}{2} \cdot \mathbb{E}_{w_t}[((1 + y h_t(x)) \exp(-\alpha_t) + (1 - y h_t(x)) \exp(\alpha_t))] \]
AdaBoost in one (long) slide

- \( A_s \): initialize \( \mathbf{w}_1 = \mathbf{u} \) and \( H_s = 0 \)
- Repeat for \( t = 1, 2, ..., T \)
  
  \[
  h_t \leftarrow A_w(\mathcal{S}, \mathbf{w}_t)
  \]

  \[
  \alpha_t \leftarrow \text{arg min}_\alpha Z_t(\alpha) = \mathbb{E}_{\mathbf{w}_t}[\exp(-\alpha y h_t(\mathbf{x}))]
  \]

  \[
  \mathbf{w}_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \circ \exp(-\alpha_t y h_t(\mathbf{x}))
  \]

- Output \( H_s = \sum_t \alpha_t h_t \)

\[
\mathbb{E}_S[\ell^{0/1}(y H(\mathbf{o}))] \leq \prod_t Z_t(\alpha_t)
\]

\[
h_t \in [-1, 1]
\]

\[
\alpha_t = \frac{1}{2} \log \frac{1+\mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]}{1-\mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]}
\]

\[
Z_t(\alpha_t) \leq \frac{1}{2} \cdot \mathbb{E}_{\mathbf{w}_t} [((1 + y h_t(\mathbf{x})) \exp(-\alpha_t) + (1 - y h_t(\mathbf{x})) \exp(\alpha_t))]
\]

\[
= \sqrt{1 - \mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]^2} \leq \exp(-(1/2)\mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]^2)
\]
AdaBoost in one (long) slide

- \( A_s : \) initialize \( \mathbf{w}_1 = \mathbf{u} \) and \( H_s = 0 \)

Repeat for \( t = 1, 2, \ldots, T \)

- \( h_t \leftarrow A_w(S, \mathbf{w}_t) \)
- \( \alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) = \mathbb{E}_{\mathbf{w}_t}[\exp(-\alpha y h_t(\mathbf{x}))] \)
- \( \mathbf{w}_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \circ \exp(-\alpha_t y h_t(\mathbf{x})) \)

output \( H_s = \sum_t \alpha_t h_t \)

Freund & Schapire, 1997
Schapire & Singer, 1999

\( \mathbb{E}_S[\ell^{0/1}(yH(\mathbf{o}))] \leq \prod_t Z_t(\alpha_t) \)

\( \alpha_t = \frac{1}{2} \log \frac{1 + \mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]}{1 - \mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]} \)
AdaBoost in one (long) slide

\[ \mathcal{A}_S: \text{initialize } \mathbf{w}_1 = \mathbf{u} \text{ and } H_s = 0 \]

Repeat for \( t = 1, 2, \ldots, T \)

\[
\begin{align*}
\mathbf{h}_t & \leftarrow \mathcal{A}_w(\mathcal{S}, \mathbf{w}_t) \\
\alpha_t & \leftarrow \arg \min_{\alpha} Z_t(\alpha) = \mathbb{E}_{\mathbf{w}_t}[\exp(-\alpha y h_t(\mathbf{x}))] \\
\mathbf{w}_{t+1} & \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \circ \exp(-\alpha_t y h_t(\mathbf{x}))
\end{align*}
\]

output \( H_s = \sum_t \alpha_t h_t \)

\[
\mathbb{E}_S[\ell^{0/1}(y H(\mathbf{o}))] \leq \exp(-(1/2)T \min_t \mathbb{E}_{\mathbf{w}_t}[y h_t(\mathbf{o})]^2)
\]
AdaBoost in one (long) slide

**Algorithm**

1. **Initialization**: Initialize $\mathbf{w}_1 = \mathbf{u}$ and $H_s = 0$

2. **Repeat** for $t = 1, 2, \ldots, T$
   - $h_t \leftarrow A_w(S, \mathbf{w}_t)$
   - $\alpha_t \leftarrow \arg \min_\alpha Z_t(\alpha) = \mathbb{E}_{\mathbf{w}_t} \left[ \exp \left( -\alpha y h_t(\mathbf{x}) \right) \right]$
   - $\mathbf{w}_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \circ \exp \left( -\alpha_t y h_t(\mathbf{x}) \right)$

3. **Output**: $H_s = \sum_t \alpha_t h_t$

---

Freund & Schapire, 1997

Schapire & Singer, 1999
AdaBoost in one (long) slide

\[ A_S : \text{initialize } \mathbf{w}_1 = \mathbf{u} \text{ and } H_S = 0 \]

Repeat for \( t = 1, 2, \ldots, T \)

\[
\begin{align*}
    h_t & \leftarrow A_w(S, \mathbf{w}_t) \\
    \alpha_t & \leftarrow \arg \min_\alpha Z_t(\alpha) \doteq \mathbb{E}_{\mathbf{w}_t} [\exp (-\alpha y h_t(x))] \\
    \mathbf{w}_{t+1} & \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot \mathbf{w}_t \circ \exp (-\alpha_t y h_t(x)) \\
    \text{output } H_S & = \sum_t \alpha_t h_t
\end{align*}
\]

\[ A_w(S, \mathbf{w}_t) \text{ Weak learner} \]

Fix \( \delta_w = \delta_s / T \)
AdaBoost in one (long) slide

\[ A_S : \text{initialize } w_1 = u \text{ and } H_s = 0 \]

Repeat for \( t = 1, 2, \ldots, T \)

\[
\begin{align*}
    h_t & \leftarrow A_w(S, w_t) \\
    \alpha_t & \leftarrow \arg \min_{\alpha} Z_t(\alpha) \equiv \mathbb{E}_{w_t}[\exp(-\alpha yh_t(x))] \\
    w_{t+1} & \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t yh_t(x)) \\
    \text{output } H_s & = \sum_t \alpha_t h_t
\end{align*}
\]

\[ A_w(S, w_t) \text{ Weak learner} \]

Fix \( \delta_w = \delta_s / T \)

With probability \( \geq 1 - \delta_s \)

\[
2 \cdot \mathbb{E}_{w_t}[\ell^{0/1}(yh_t(o))] \leq 1 - 2\gamma (\forall t)
\]
AdaBoost in one (long) slide

Freund & Schapire, 1997
Schapire & Singer, 1999

$A_s$ : initialize $w_1 = u$ and $H_s = 0$

Repeat for $t = 1, 2, \ldots, T$

\[
\begin{align*}
& h_t \leftarrow A_w(S, w_t) \\
& \alpha_t \leftarrow \text{arg min}_\alpha Z_t(\alpha) = \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))] \\
& w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \odot \exp(-\alpha_t y h_t(x))
\end{align*}
\]

output $H_s = \sum_t \alpha_t h_t$

$h_t \in \{-1, +1\} \Rightarrow 2 \cdot \mathbb{E}_{w_t}[\ell^{0/1}(y h_t(o))] = \mathbb{E}_{w_t}[1 - y h_t(o)] = 1 - \mathbb{E}_{w_t}[y h_t(o)]$

$A_w(S, w_t)$ Weak learner

Fix $\delta_w = \delta_s/T$

With probability $\geq 1 - \delta_s$

$2 \cdot \mathbb{E}_{w_t}[\ell^{0/1}(y h_t(o))] \leq 1 - 2\gamma (\forall t)$
AdaBoost in one (long) slide

\[ A_s : \text{initialize } w_1 = u \text{ and } H_s = 0 \]

Repeat for \( t = 1, 2, \ldots, T \)

\[
\begin{align*}
  h_t & \leftarrow A_w(S, w_t) \\
  \alpha_t & \leftarrow \arg \min_{\alpha} Z_t(\alpha) \triangleq E_{w_t}[\exp(-\alpha y h_t(x))] \\
  w_{t+1} & \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x))
\end{align*}
\]

output \( H_s = \sum_t \alpha_t h_t \)

\( (h_t \in \{-1, +1\}) \Rightarrow 2 \cdot E_{w_t}[\ell^{0/1}(y h_t(o))] = E_{w_t}[1 - y h_t(o)] = 1 - E_{w_t}[y h_t(o)] \Rightarrow E_{w_t}[y h_t(o)]^2 \geq 4\gamma^2 (\forall t) \)
**AdaBoost in one (long) slide**

- $A_s$: initialize $w_1 = u$ and $H_s = 0$
- Repeat for $t = 1, 2, ..., T$
  - $h_t \leftarrow A_w(S, w_t)$
  - $\alpha_t \leftarrow \arg \min \alpha Z_t(\alpha) = \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))]$
  - $w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x))$
- Output $H_s = \sum_t \alpha_t h_t$

$A_w(S, w_t)$ Weak learner

- Fix $\delta_w = \delta_s / T$
- With probability $\geq 1 - \delta_s$
  
$$2 \cdot \mathbb{E}_{w_t}[\ell^{0/1}(y h_t(o))] \leq 1 - 2\gamma (\forall t)$$

$h_t \in \{-1, +1\}$

$2 \cdot \mathbb{E}_{w_t}[\ell^{0/1}(y h_t(o))] = \mathbb{E}_{w_t}[1 - y h_t(o)] = 1 - \mathbb{E}_{w_t}[y h_t(o)]$

$\Rightarrow \mathbb{E}_{w_t}[y h_t(o)]^2 \geq 4\gamma^2 (\forall t)$

$\Rightarrow \mathbb{E}_S[\ell^{0/1}(y H_s(o))] \leq \exp(-2T \gamma^2)$
**AdaBoost in one (long) slide**

- **$A_s$** : initialize $w_1 = u$ and $H_s = 0$
- Repeat for $t = 1, 2, ..., T$
  - $h_t \leftarrow A_w(S, w_t)$
  - $\alpha_t \leftarrow \arg \min_{\alpha} Z_t(\alpha) \doteq \mathbb{E}_{w_t}[\exp(-\alpha y h_t(x))]$
  - $w_{t+1} \leftarrow \frac{1}{Z_t(\alpha_t)} \cdot w_t \circ \exp(-\alpha_t y h_t(x))$
- Output $H_s = \sum_t \alpha_t h_t$

**$A_w(S, w_t)$** Weak learner
- Fix $\delta_w = \delta_s / T$
- With probability $\geq 1 - \delta_s$
  - $2 \cdot \mathbb{E}_{w_t}[\ell^{0/1}(y h_t(o))] \leq 1 - 2\gamma (\forall t)$

Fix $T \geq \frac{1}{2\gamma^2 \log m}$ **$H_s$ is consistent**

$m = O\left(\frac{1}{\epsilon_s} \log \frac{1}{\delta_s} + \frac{|\mathcal{H}_w|}{\epsilon_s} \log \frac{1}{\epsilon_s}\right)$ \Rightarrow **$A_s$ is PAC**

- Freund & Schapire, 1997
- Schapire & Singer, 1999
- Vapnik, 1982
AdaBoost is a strong / PAC learning algorithm with the following features:

$$|S| = O \left( \frac{1}{\epsilon_s} \log \frac{1}{\delta_s} + \frac{|\mathcal{H}_w|}{\epsilon_s} \log \frac{1}{\epsilon_s} \right)$$

parameters $\epsilon_s, \delta_s$
Summary

- AdaBoost is a strong / PAC learning algorithm with the following features:

\[
T = O \left( \frac{1}{2\gamma^2} \log m \right) \\
= O \left( \text{poly}(d, \ldots) \log m \right) \\
= O \left( \text{poly}(d, \log(\epsilon_s^{-1}, \delta_s^{-1}, |\mathcal{H}_w|, \ldots)) \right)
\]
Summary

- What we know of AdaBoost so far
  - Directly minimises a **surrogate** of the empirical risk
  - Can we do better than rate:
    \[
    \mathbb{E}_S[\ell_{0/1}(yH_s(o))] \leq \exp(-2T\gamma^2)
    \]
  - If so, \( \text{NP} \subseteq \text{DTIME}[n^{\log \log n}] \)

Nock & Nielsen, 2004
Summary

- Adaboost
  - is a formal boosting algorithm
  - has virtually unbeatable convergence rate under the weak learning assumption
  - induces a linear separator

good, but at the age of “deep learning”, one loves complex “architectures”
ANY OTHER ARCHITECTURES ?
(Many) other architectures!

- The most popular DT induction algorithms proceed in a top-down fashion...

\[ \emptyset \longrightarrow \{+1\} \]

\[ \emptyset \rightarrow v_5 \]

\[ B \rightarrow \begin{cases} +1, & \text{if } \emptyset = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \end{cases} \]

\[ +1, -1, +1 \]

\[ \mathcal{L}(H) \]

- ... and minimize an expected \textbf{concave} function to pick splits:

\[ \varepsilon^\phi(S, H) = \sum_{k \in \mathcal{L}(H)} |S_k| \cdot \phi\left(\frac{|S_k^{+1}|}{|S_k|}\right) \]

\[ S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \]

\[ (S_k = S_k^{+1} \cup S_k^{-1}) \]
Certainly, AdaBoost does not seem similar to C4.5, CART, etc...:

- No weights (DT - CART, C4.5) vs weights (LS - AdaBoost)
- Minimisation of convex surrogate (LS) vs concave (DT)
- ... and of course the representation formalism
From AdaBoost to...

... but a DT can be represented using several LS:

\[ v_1 = 0 \]
\[ v_2 = 0 \]
\[ v_2 = 1 \]
\[ v_1 = 1 \]

\[ h_1 \]
\[ h_2 \]
\[ h_3 \]
\[ h_4 \]
\[ h_5 \]

\[ (\alpha_1, h_1) = (-2, +1) \]
\[ (\alpha_2, h_2) = (3, +1) \]
\[ (\alpha_3, h_3) = (3, +1) \]
\[ (\alpha_4, h_4) = (0, +1) \]
\[ (\alpha_5, h_5) = (2, -1) \]

\[ +1 \]
Let us use AdaBoost to fit the $\alpha$s ("AdaBoost-DT")!

Natural splitting function: minimize $\mathbb{E}_S [\exp(-y H(o))]$

What splitting function does it bring?

What DT is built? Any weak/strong guarantee?
Let us minimise $\mathbb{E}_S[\exp(-yH(o))]$

Weight update in leaf $t$

For weight update, assuming $S_t^{-1} \neq \emptyset$ and $S_t^{+1} \neq \emptyset$, we have

$$\frac{\partial}{\partial \alpha} \mathbb{E}_{w_t}[\exp(-\alpha yh_t(x))] = 0 \quad \text{(picking $\alpha_t$ in AdaBoost)}$$
From AdaBoost to...

\[ \frac{\partial \mathbb{E}_{w_t} [\exp(-\alpha y h_t(x))]}{\partial \alpha} = 0 \]

\[ \sum_{(o,+1) \in S_t} w_t((o,+1)) h_t \exp(-\alpha h_t) + \sum_{(o,-1) \in S_t} w_t((o,-1)) h_t \exp(\alpha h_t) = 0 \]

(because \( h_t \) constant and classifies only \( S_t \))

Reorganise, drop \( h_t \), divide by \( Z_t(\alpha_t) \), obtain

\[ \sum_{(o,+1) \in S_t} \frac{1}{Z_t} \cdot w_t((o,+1)) \exp(-\alpha h_t) = \sum_{(o,-1) \in S_t} \frac{1}{Z_t} \cdot w_t((o,-1)) \exp(\alpha h_t) \]
From AdaBoost to...

\[ \frac{\partial \mathbb{E}_{w_t} [\exp(-\alpha y h_t(x))]}{\partial \alpha} = 0 \]

\[ \sum_{(o, +1) \in S_t} w_t((o, +1)) h_t \exp(-\alpha h_t) + \sum_{(o, -1) \in S_t} w_t((o, -1)) h_t \exp(h_t) = 0 \]

(because \( h_t \) constant, say 1, and classifies only \( S_t \))

Reorganise, drop \( h_t \), divide by \( Z_t(\alpha_t) \), obtain

\[ \sum_{(o, +1) \in S_t} \frac{1}{Z_t} \cdot w_t((o, +1)) \exp(-\alpha h_t) = \sum_{(o, -1) \in S_t} \frac{1}{Z_t} \cdot w_t((o, -1)) \exp(h_t) \]

\[ w_{t+1}(S_t^{+1}) = w_{t+1}(S_t^{-1}) \]
Hence,

\[ 0 = \frac{1}{2} \log \frac{w_{t+1}(S_t^+)}{w_{t+1}(S_t^-)} \]
Hence,

\[
0 = \frac{1}{2} \log \frac{w_{t+1}(S_t^+)}{w_{t+1}(S_t^-)}
\]

Consider prediction for one example arriving at leaf \( t \)

\[
w_{t+1}((o, +1)) = \frac{1}{Z_t} w_t((o, +1)) \exp(-\alpha_t h_t)
\]

\[
= \frac{1}{m \prod_{j \in P_t(o)} Z_j} \exp \left( - \sum_{j \in P_t(o)} \alpha_j h_j \right)
\]
Hence,

\[ 0 = \frac{1}{2} \log \frac{w_{t+1}(S^+_t)}{w_{t+1}(S^-_t)} \]

Get total prediction for +1 and -1 examples arriving at leaf \( t \)

\[ w_{t+1}(S^+_t) = \frac{|S^+_t|}{m \prod_{j \in P_t(o)} Z_j} \exp \left( - \sum_{j \in P_t(o)} \alpha_j h_j \right) \]

\[ w_{t+1}(S^-_t) = \frac{|S^-_t|}{m \prod_{j \in P_t(o)} Z_j} \exp \left( \sum_{j \in P_t(o)} \alpha_j h_j \right) \]
Combining both we get:

\[ 0 = \frac{1}{2} \log \frac{w_{t+1}(S_t^+)}{w_{t+1}(S_t^-)} = \log \frac{|S_t^+|}{|S_t^-|} - 2 \sum_{j \in P_t(o)} \alpha_j h_j \]
Combining both we get:

\[
0 = \frac{1}{2} \log \frac{w_{t+1}(S_t^+)}{w_{t+1}(S_t^-)} = \log \frac{|S_t^+|}{|S_t^-|} - 2 \sum_{j \in \mathcal{P}_t(o)} \alpha_j h_j
\]

This is just $H_T(o)$ for any observation $o$ that comes to leaf $t$.

... and the prediction of DT-AdaBoost simplifies to

\[
H_T(o) = \frac{1}{2} \log \frac{|S_t^+|}{|S_t^-|}
\]
From AdaBoost to...

- **Summary**: prediction can be computed without weights, and aggregate.

- **What about** \( \mathbb{E}_S[\exp(-yH(o))] \)?

- **We have**

\[
\mathbb{E}_S[\exp(-\alpha y h_T(x))] = \sum_{t \text{ leaf}} \frac{|S_t|}{|S|} \left( \frac{|S_t^+|}{|S_t|} \exp(-H_T(t)) + \frac{|S_t^-|}{|S_t|} \exp(H_T(t)) \right)
\]

\[
H_T(o) = \frac{1}{2} \log \frac{|S_t^+|}{|S_t^-|}
\]
Summary: prediction can be computed without weights, and aggregate.

What about \( \mathbb{E}_S[\exp(-yH(o))] \)?

We have

\[
\mathbb{E}_S[\exp(-\alpha y h_T(x))] = \sum_{t \text{ leaf}} \frac{|S_t|}{|S|} \cdot \left( \frac{|S_t^1|}{|S_t|} \exp(-H_T(t)) + \frac{|S_t^{-1}|}{|S_t|} \exp(H_T(t)) \right)
\]

\[
= \sum_{t \text{ leaf}} \frac{|S_t|}{|S|} \cdot 2 \sqrt{\frac{|S_t^1|}{|S_t|}} \left( 1 - \frac{|S_t^1|}{|S_t|} \right)
\]
**From AdaBoost to...**

- **Summary:** prediction can be computed without weights, and aggregate.

- **What about** \( \mathbb{E}_S[\exp(-yH(o))] \)?

- **So, AdaBoost-DT’s expected exponential loss is**

\[
\mathbb{E}_S[\exp(-\alpha y h_T(x))] = \mathbb{E}_{t \sim \text{leaf}}[\phi(p_t^+)]
\]

\[
p_t = \frac{|S_t|}{|S|}
\]

\[
H_T(o) = \frac{1}{2} \log \frac{|S_t^{+1}|}{|S_t^{-1}|}
\]

\[
p_t^+ = \frac{|S_t^{+1}|}{|S_t|}
\]

\[
\phi(x) = 2\sqrt{x(1-x)}
\]
From AdaBoost to...

❖ Summary: prediction can be computed without weights, and aggregate.

❖ What about \( \mathbb{E}_S[\exp(-yH(o))] \)?

❖ So, AdaBoost-DT’s expected exponential loss is

\[
\mathbb{E}_S[\exp(-\alpha yh_T(x))] = \mathbb{E}_{t \sim \text{leaf}}[\phi(p_t^+)]
\]

convex concave
So, Adaboost–DT predicts at the leaves $H_T(o) = \frac{1}{2} \log \frac{|S_t^{+1}|}{|S_t^{-1}|}$

and repeatedly minimises $E_{t \sim \text{leaf}}[\phi(p_t^+)]$, which depends only on leaves (and again, no weight appears)

We get a DT with real-valued predictions at the leaves:

Let us pick the splits following the min of $E_{t \sim \text{leaf}}[\phi(p_t^+)]$
From AdaBoost to CART-C4.5!

- We get an algorithm from the CART-C4.5 family!

\[ \mathbb{E}_{t \sim \text{leaf}}[\phi(p_t^+)] \]

- The splitting criterion was first analysed in Kopans & Mansour, 1999

- They proved (long proof) that it yields a boosting algorithm

- ... can we do the same from AdaBoost's analysis?

\[ \mathbb{E}_{S}[-yH(o)] \leq \prod_{t} Z_t(\alpha_t) \]
Proof: let us first rewrite
\[ \exp(-yH_s(o)) = (\prod_{t} Z_t(\alpha_t))w_1 = \prod_{t} Z_t(\alpha_t)u \]

So in fact
\[ \exp(-yH_s(o)) = \left( \prod_{j \in \mathcal{P}(o)} Z_j \right)u \]

So in fact
\[ \mathbb{E}_S[\exp(-yH_s(o))] = \mathbb{E}_{t \sim \text{leaf}} \left[ \prod_{j \in \mathcal{P}(t)} Z_j \right] \]

\[ p_t = \frac{|S_t|}{|S|} \]
Proof: we obtain $\alpha_t = \frac{1}{2} \log \frac{w_t(S_t^{+1})}{w_t(S_t^{-1})}$, and thus

$$Z_t \leq w_t(S \setminus S_t) + 2w_t(S_t) \sqrt{\frac{w_t(S_t^{+1})}{w_t(S_t)}} \left(1 - \frac{w_t(S_t^{+1})}{|w_t(S_t)|}\right)$$

What is the weak learning assumption?
From AdaBoost to CART-C4.5!

❖ Proof: we obtain \( \alpha_t = \frac{1}{2} \log \frac{w_t(S_t^{+1})}{w_t(S_t^{-1})} \), and thus

\[
Z_t \leq w_t(S \setminus S_t) + 2w_t(S_t) \sqrt{\frac{w_t(S_t^{+1})}{w_t(S_t)}} \left( 1 - \frac{w_t(S_t^{+1})}{|w_t(S_t)|} \right)
\]

❖ The weak learner \( A_w \) “receives” \( S_t \), so distribution \( w_t \) is \textbf{truncated} in \( S_t \). He returns a \textbf{constant} \( h_t \).

❖ Hence, with probability \( \geq 1 - \delta_w \),

\[
\frac{1}{w_t(S_t)} \min_{y \in \{-1,+1\}} \sum_{(o,y) \in S_t} w_t((o,y)) \leq \frac{1}{2} - \gamma
\]
From AdaBoost to CART-C4.5!

We get, for some $y$, \[ \frac{w_t(S_t^y)}{w_t(S_t)} \leq \frac{1}{2} - \gamma, \text{ and so:} \]

\[
Z_t \leq w_t(S \setminus S_t) + 2w_t(S_t) \sqrt{\frac{w_t(S_t^{+1})}{w_t(S_t)}} \left(1 - \frac{w_t(S_t^{+1})}{|w_t(S_t)|}\right)
\]

\[
\leq w_t(S \setminus S_t) + w_t(S_t) \sqrt{1 - 4\gamma^2}
\]

\[
\leq \sqrt{1 - 4w_t(S_t)\gamma^2}
\]

\[
\leq \exp(-2w_t(S_t)\gamma^2)
\]

\[
\leq \exp(-2\gamma^2/t)
\]

because $\phi(x) = \phi(1 - x)$

because $\sqrt{1 - x}$ concave

because $1 - x \leq \exp(-x)$

because one leaf “heavy”

\[
E_S[\exp(-yH(o))] = E_t \text{ leaf } \prod_{j \in P(t)} Z_j \leq T^{-\gamma^2}
\]
So far

- weak learning = strong learning
- AdaBoost + DT induction algorithms are boosting algorithms
- Malleable analysis that can fit to various architectures to prove fast empirical convergence under a weak learning assumption
So far

- weak learning = strong learning
- AdaBoost + DT induction algorithms are boosting algorithms
- Malleable analysis that can fit to various architectures to prove fast empirical convergence under a weak learning assumption
- Oblique decision trees *Henry & al., 2007*
So far

- weak learning = strong learning
- AdaBoost + DT induction algorithms are boosting algorithms
- Malleable analysis that can fit to various architectures to prove fast empirical convergence under a weak learning assumption
  - Oblique decision trees *Henry & al., 2007*
  - Alternating decision trees *Freund & Mason, 1999*
- etc.
So far

- weak learning = strong learning
- AdaBoost + DT induction algorithms are boosting algorithms
- Malleable analysis that can fit to various architectures to prove fast empirical convergence under a weak learning assumption
- Moreover, analysis shows that it is a good idea to minimize

\[ \mathbb{E}_S[\varphi(yH(o))] \]

- ...when \( \varphi(z) = \exp(-z) \)
ANY OTHER INTERESTING ?
Many other interesting $\varphi$!

- A sufficient condition for classification, called classification calibration, is necessary and sufficient for any $\varphi$ to be “interesting” for classification

*Bartlett & al., 2006*
Many other interesting $\varphi$!

- Four risks:

  - $\varphi$-loss: $\min \mathbb{E}_S[\varphi(yH(o))]$ vs $\mathbb{E}_D[\varphi(yH(o))]$
  - 0/1-loss: $\mathbb{E}_S[\ell_{0/1}(yH(o))]$ vs $\mathbb{E}_D[\ell_{0/1}(yH(o))]$
Classification Calibration

❄ Conditions under which the minimisation of $\mathbb{E}_D[\varphi(yH(o))]$ implies the minimisation of $\mathbb{E}_D[\ell^{0/1}(yH(o))]$

$$|\mathbb{E}_D[\varphi(yH(o))] - \inf_{h} \mathbb{E}_D[\varphi(yh(o))]| \to_A 0$$

❄ We prove $\psi(C - D) \leq A - B$ for non-decreasing $\psi$
Reason on pointwise loss ($\eta \in [0, 1]$)

\[
U(\eta) = \inf_{\alpha} \eta \varphi(\alpha) + (1 - \eta) \varphi(-\alpha)
\]

\[
U^-(\eta) = \inf_{\alpha: \alpha(2\eta - 1) \leq 0} \eta \varphi(\alpha) + (1 - \eta) \varphi(-\alpha)
\]

If $\eta = \mathbb{P}_D[y = +1|o]$, $U(\eta)$ is the optimal $\varphi$-risk on $o$

$U^-(\eta)$ is the optimal disagreeing $\varphi$-risk on $o$

**Def**

$\varphi$ is classification-calibrated if $U(\eta) < U^-(\eta)$

**Thm**

$|\mathbb{E}_D[\varphi(yH(o))] - \inf_h \mathbb{E}_D[\varphi(\varphi h(o))]| \to_A 0$ implies $|\mathbb{E}_D[\ell^{0/1}(yH(o))] - \inf_h \mathbb{E}_D[\ell^{0/1}(\varphi h(o))]| \to 0$

iff $\varphi$ is classification-calibrated

*Bartlett & al., 2006*
Classification Calibration

Proof: Let \( \psi(z) = \text{co}(U^-(1 + z)/2) - U((1 + z)/2) \) and \( \delta(H, o) = 1_{H(o)(2\eta(o) - 1) \leq 0} \).

Then,

\[
\begin{align*}
\text{co}\tilde{\psi} & \left( \mathbb{E}_D[\ell^0/1(yH(o))] - \inf_{h} \mathbb{E}_D[\ell^0/1(yh(o))] \right) \\
& = \text{co}\tilde{\psi}(\mathbb{E}_D[\delta(H, o)|2\eta - 1|])
\end{align*}
\]

For \( H = +1 \Rightarrow \Delta = (1 - \eta) - \min\{\eta, 1 - \eta\} = \left\{ \begin{array}{ll} 1 - 2\eta & \text{if } \eta < 1/2 \\ 0 & \text{otherwise} \end{array} \right. \)

For \( H = -1 \Rightarrow \Delta = \eta - \min\{\eta, 1 - \eta\} = \left\{ \begin{array}{ll} 0 & \text{if } \eta < 1/2 \\ 2\eta - 1 & \text{otherwise} \end{array} \right. \)
Proof: Let \( \psi(z) = \text{co}(U^-(1 + z)/2) - U((1 + z)/2) \) and \( \delta(H, o) = 1_{H(o)(2\eta(o) - 1) \leq 0} \). Then,

\[
\text{co}\tilde{\psi}\left(\mathbb{E}_D[\ell^0/1(yH(o))] - \inf_h \mathbb{E}_D[\ell^0/1(yh(o))]\right)
= \text{co}\tilde{\psi}(\mathbb{E}_D[\delta(H, o)|2\eta - 1|])
\leq \mathbb{E}_D[\delta(H, o)\text{co}\tilde{\psi}(|2\eta - 1|)]
\]

\( \psi \) convex

\( \psi(0) = 0 \)
Classification Calibration

Proof: Let $\psi(z) = \co (U^{-}((1 + z)/2) - U((1 + z)/2))$ and $\delta(H, o) = 1_{H(0)(2\eta(o) - 1) \leq 0}$

Then,

$$\co \tilde{\psi} \left( \mathbb{E}_D[\ell^{0/1}(yH(o))] - \inf_h \mathbb{E}_D[\ell^{0/1}(yh(o))] \right)$$

$$= \co \tilde{\psi}(\mathbb{E}_D[\delta(H, o)|2\eta - 1|])$$

$$\leq \mathbb{E}_D[\delta(H, o)\tilde{\psi}(|2\eta - 1|)]$$

$$\leq \tilde{\psi}(|2\eta - 1|)$$

$$0 \leq \tilde{\psi} \leq \co \tilde{\psi}$$
Classification Calibration

Proof: Let $\psi(z) = \text{co} (U^-(1 + z)/2) - U((1 + z)/2)$ and $\delta(H, o) = 1_{H(o)(2\eta(o) - 1) \leq 0}$. Then,

$$\text{co} \tilde{\psi} \left( \mathbb{E}_D[\ell^{0/1}(yH(o))] - \inf_h \mathbb{E}_D[\ell^{0/1}(yh(o))] \right)$$

$$= \text{co} \tilde{\psi}(\mathbb{E}_D[\delta(H, o)|2\eta - 1|])$$

$$\leq \mathbb{E}_D[\delta(H, o)\text{co} \tilde{\psi}(|2\eta - 1|)]$$

$$\leq \mathbb{E}_D[\delta(H, o)\tilde{\psi}(|2\eta - 1|)]$$

$$= \mathbb{E}_D[\delta(H, o)(U^-(\eta(o)) - U(\eta(o)))]$$

$$= \mathbb{E}_D[\delta(H, o) \left( \inf_{\alpha: \alpha(2\eta - 1) \leq 0} \eta \varphi(\alpha) + (1 - \eta)\varphi(-\alpha) - U(\eta(o)) \right)]$$

definition
**Classification Calibration**

Proof: Let $\psi(z) = \text{co} \left( U^-(1 + z)/2) - U((1 + z)/2) \right)$ and $\delta(H, o) = 1_{H(o)(2\eta(o)-1) \leq 0}$.

Then,

$$\text{co} \psi \left( \mathbb{E}_D[\ell^{0/1}(yH(o))] - \inf_{h} \mathbb{E}_D[\ell^{0/1}(yh(o))] \right)$$

$$= \text{co} \psi(\mathbb{E}_D[\delta(H, o)|2\eta - 1|])$$

$$\leq \mathbb{E}_D[\delta(H, o)\text{co} \psi(|2\eta - 1|)]$$

$$\leq \mathbb{E}_D[\delta(H, o)\psi(|2\eta - 1|)]$$

$$= \mathbb{E}_D[\delta(H, o)(U^-(\eta(o)) - U(\eta(o)))$$

$$= \mathbb{E}_D[\delta(H, o) \left( \inf_{\alpha: \alpha(2\eta - 1) \leq 0} \eta\varphi(\alpha) + (1 - \eta)\varphi(-\alpha) - U(\eta(o)) \right)$$

$$\leq \mathbb{E}_D[\varphi(yH(o))] - \inf_{h} \mathbb{E}_D[\varphi(yh(o))]$$

immediate when $\delta(., .) = 1$

otherwise, $(\eta\varphi(H) + (1 - \eta)\varphi(-H) - H(\eta)) \geq 0$
Classification Calibration

We get \( \psi(C - D) \leq A - B \)

To get the theorem, one remarks that \( \psi \) is continuous, \( \psi(0) = 0 \) and classification calibration is equivalent to \( \psi(z) > 0, \forall z > 0 \)

Thus every sequence of classifiers \( H_1, H_2, \ldots, H_T \) such that

\[
\mathbb{E}_D[\varphi(yH_T(o))] \to_T \inf_{h} \mathbb{E}_D[\varphi(yh(o))]
\]

guarantees

\[
\mathbb{E}_D[\ell^{0/1}(yH_T(o))] \to_T \inf_{h} \mathbb{E}_D[\ell^{0/1}(yh(o))]
\]

whenever \( \varphi \) is classification calibrated

When \( \varphi(z) = \exp(-z) \), a fast minimisation exists wrt WLA
ANY OTHER “FAST” ᴏ?
Many other fast $\varphi$!

- All proper losses(*) with identical class-wise misclassification costs admit an efficient boosting algorithm

- Loss induces a dually flat geometry over a set of weights

- Boosting is a geometric game over this set of weights

[Collins & al., 2002] [Murata & al., 2004]
[Nock & Nielsen, 2009]
Boosting - dually flat coordinate system induced by surrogate loss

Loss = proper scoring rule, symmetric, + diff assumptions

\[ \Rightarrow \text{loss} = \varphi(yH) \propto D_\phi(y_0/1 \| (\nabla \phi)^{-1}(H)) \]

(weights)

(posterior est.)

(real valued class.)
Boosting ∀ φ, overview

**Boosting**

\[ h_1, h_2, ..., h_j \]

\[ H_0 \rightarrow H_j \rightarrow H_{j+1} \rightarrow H_\infty \]

**weights**

\[ w_0 \rightarrow w_j \rightarrow w_{j+1} \rightarrow w_\infty \]

\[ w_0 \propto 1 \]

\[ \mathbb{R}^m \]

\[ y_i h_t(o_i) \]

\[ \text{Mean operator} = \sum_i w_\infty, i y_i o_i \]

\[ \text{ker} M^T \]

Boosting “exhausts” the mean operator

*Kueck & de Freitas, 2005*  
*Patrini & al., 2014*
Boosting $\forall \varphi$, divergence 101

Let $D : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, verifying:

$$D(w \| w') \geq 0, \forall w, w'$$

$$\forall \{w_1, w_2, ..., w_m\}, \arg \min_w \sum_i D(w_i \| w) = \frac{1}{m} \sum_i w_i$$

Then $\exists \varphi : \mathbb{R} \rightarrow \mathbb{R}$ strictly convex differentiable, such that:

$$D(w \| w') = \varphi(w) - \varphi(w') - (w - w') \nabla \varphi(w')$$

$$= D_\varphi(w \| w')$$

Bregman divergence
We can represent graphically the divergence $D_{\varphi}(w||w')$: 

$D_{\varphi}(w||w')$
Boosting $\forall \varphi$, divergence 101

- $D_\varphi (w \| w')$ admits famous examples
  - Kullback-Leibler, for $\varphi (w) = w \log w - w$
  - Itakura-Saito, for $\varphi (w) = - \log w$
  - Mahalanobis, for $\varphi (w) = w^2$
- $D_\varphi (w \| w')$ satisfies identity of indiscernibles, not symmetry (except Mahalanobis), not triangle inequality
Boosting $\forall \varphi$, loss - divergence

.character

The following holds:

$$
\varepsilon^{\varphi}(S, H) = \sum_i \varphi(y_i H(o_i)) = g(m) + \sum_i D_{\tilde{\varphi}}(0\|w_i)
$$

$$
= g(m) + D_{\tilde{\varphi}}(0\|w)
$$

$$
\tilde{\varphi}(w) = \varphi^*(-w), \varphi^*(w) = \sup_{w'} \{ww' - \varphi(w')\}
$$

Nock & Nielsen, 2009
Boosting $\forall \varphi$, edge matrix

- Storing all the boosting information: $(M)_{it} = -y_i^* h_t(o_i)$

$$M \doteq -
\begin{bmatrix}
y_1^* h_1(o_1) & \cdots & y_1^* h_t(o_1) & \cdots & y_1^* h_T(o_1) \\
\vdots & & \vdots & & \vdots \\
y_i^* h_1(o_i) & \cdots & y_i^* h_t(o_i) & \cdots & y_i^* h_T(o_i) \\
\vdots & & \vdots & & \vdots \\
y_m^* h_1(o_m) & \cdots & y_m^* h_t(o_m) & \cdots & y_m^* h_T(o_m)
\end{bmatrix}$$

- all information about example $i$
- all information about weak classifier $t$
Boosting $\forall \varphi$, edge matrix

- Two important quantities:
  
  $-(M\alpha)_i = y_i^* \sum_{t=1}^{T} \alpha_t h_t(o_i)$, edge of $H$ on $(o_i, y_i^*)$

  $-(M^\top w)_t = \sum_{i=1}^{m} w_i y_i^* h_t(o_i)$, edge of $h_t$ on $(S, w)$
Best classifier(s)

\[
\min_H \varepsilon^\varphi(S, H) = \min_{w \in \mathbb{W}} D_\varphi(0\|w) = D_\varphi(0\|w_\infty)
\]

Initial classifier

Set of possible weights that can be generated through \(\mathcal{H}(\text{closed})\)

\[
\ker M^\top = \{u : M^\top u = 0\}
\]
Boosting \( \forall \varphi \), Bregman-Pythagore

\[
D_{\tilde{\varphi}}(0\|w) = D_{\tilde{\varphi}}(0\|w_\infty) \\
+ D_{\tilde{\varphi}}(w_\infty\|w), \\
\forall w \in \overline{W}
\]

\[
(\varepsilon^{\varphi}(S, H) = g(m) + D_{\tilde{\varphi}}(0\|w)
\]

Shows that \( w_\infty \) corresponds to optimal classifier
Boosting $\forall \varphi$, Bregman-Pythagoras

Bregman ball = classifiers with same SR
$\{ w' : D_{\varphi}(w_\infty || w') = D_{\varphi}(w_\infty || w) \}$

\[
\varepsilon^\varphi(S, H) = g(m) + D_{\varphi}(0 || w) \\
= g(m) + D_{\varphi}(0 || w_\infty) + D_{\varphi}(w_\infty || w)
\]

Fixed when $(S, \mathcal{H})$ fixed
Boosting \( \forall \varphi \), initialisation

\[ w_0 = -\nabla \varphi(0) 1 \]

**Principle:** \( H_0 = 0 \ (\alpha_0 = 0) \)

\[ w_i \overset{\dagger}{=} -\nabla \varphi(y H(o)) \]

\[ = -\nabla \varphi(y \cdot 0) \]

\[ = -\nabla \varphi(0) \]
Principle: pick a set of weak classifiers indexes \( \mathcal{T}_j \subseteq \{1, 2, \ldots, T\} \) (in general, \(|\mathcal{T}_j| = 1\))

Define \( M|_{\mathcal{T}_j} \) by suppressing columns \( \not\in \mathcal{T}_j \)

\[
\ker M^\top \subseteq \ker M|_{\mathcal{T}_j}^\top
\]
Compute $\alpha_j$ such that:

$\mathbf{w}_{j+1} \in \ker \mathbf{M}_{|\mathcal{T}_j}^T$

and

$D_{\tilde{\varphi}}(0\|\mathbf{w}_{j+1}) < D_{\tilde{\varphi}}(0\|\mathbf{w}_j)$

Thus,

$\varepsilon_{\varphi}(\mathcal{S}, H_{j+1}) < \varepsilon_{\varphi}(\mathcal{S}, H_j)$
Boosting $\forall \phi$, iteration $j$

- Compute $\alpha_j$ via solving for $\delta$: $\mathbb{R}^m$
  \[ \forall t \in T_j, \sum_i m_{it} \nabla \phi(-(M(\alpha_j + \delta))_i) = 0 \]
  (solution guaranteed to exist)

- Update:
  \[ \alpha_{j+1} = \alpha_j + \delta \]

- Update ensures Bregman-Pythagoras:
  \[ D_\phi(0\|w_j) = D_\phi(0\|w_{j+1}) + D_\phi(w_{j+1}\|w_j) \]
  ... and thus decrease of SR as long as $D_\phi(w_{j+1}\|w_j) > 0$
Boosting $\forall \varphi$, convergence

- When update always produces $\delta = 0$, $\forall \mathcal{T}_j$

$$\delta = 0, \forall \mathcal{T}_j$$

$\Leftrightarrow \; \mathbf{w}_j \in \ker M_{\mathcal{T}_j}^T, \forall \mathcal{T}_j$

$\Leftrightarrow \; \mathbf{w}_j \in \ker M$

$\Leftrightarrow \; \mathbf{w}_j = \mathbf{w}_\infty$

$\Leftrightarrow \; D_\varphi(\mathbf{0} \parallel \mathbf{w}_j) = \min_{\mathbf{w} \in \overline{\mathcal{W}}} D_\varphi(\mathbf{0} \parallel \mathbf{w})$

$\Leftrightarrow \; \varepsilon_\varphi(\mathcal{S}, H_j) = \min_H \varepsilon_\varphi(\mathcal{S}, H)$

- Converged to global optimum

$\mathfrak{w}_0 = -\nabla \varphi(0) \mathbf{1}$
When it is not the case, we can show that the WLA implies

\[ D_\varphi(w_{j+1} \| w_j) \geq Q \]

\[ Q = a_\varphi \left( \frac{\gamma \min_j Z_j}{2 \max_t \| m_t \|_2} \right)^2 \]

+ constraints on \( m \) yield formal boosting

Collins & al., 2002
Murata & al., 2004
Nock & Nielsen, 2009
Boosting $\forall \varphi$, for trees

- The trick LS / DT of AdaBoost still works

- Yields a family of boosting algorithms parameterised by the entropy

\[ \varepsilon^\phi(S, H) = \sum_{k \in \mathcal{L}(H)} |S_k| \cdot \phi\left( \frac{|S_{k+1}|}{|S_k|} \right) \]

- Popular examples:
  \[ \varphi(z) = \log(1 + \exp(-z)) \Rightarrow \text{c4.5} \]
  \[ \varphi(z) = (1 - z)^2 \Rightarrow \text{CART} \]
  \[ \varphi(z) = \log(1 + \exp(-z)) \Rightarrow \text{Breiman \& al., 1984} \]
  \[ \varphi(z) = (1 - z)^2 \Rightarrow \text{Quinlan, 1993} \]

\[ \phi_{\text{CART}}(x) = 4x(1 - x) \]
\[ \phi_{\text{c4.5}}(x) = -x \log_2(x) - (1 - x) \log_2(1 - x) \]
Many interesting losses yield efficient boosting algorithms.

Each of them not just minimizes $\ell^{0/1}(\cdot)$ (which considers only the sign of $\text{arg}$), but a surrogate function of $yH(o)$.

When $yH(o) \leq 0$, wrong label.

Theory of boosting is pleased, but experiments show that more is actually achieved

$$\mathbb{E}_D[\ell^{0/1}(yH(o))]$$

$$\mathbb{E}_S[\ell^{0/1}(yH(o))]$$

Schapire & al., 1998
Many interesting losses yield efficient boosting algorithms.

Each of them not just minimizes $\ell^{0/1}(\cdot)$ (which considers only the sign of $\arg$), but a surrogate function of $yH(o)$.

When $yH(o) \leq 0$, wrong label.

Theory of boosting is pleased, but experiments show that more is actually achieved.

\[ E_D[\ell^{0/1}(yH(o))] \]
\[ E_S[\ell^{0/1}(yH(o))] \]

Training error zero, test error keeps on decreasing!
Many interesting losses yield efficient boosting algorithms

Each of them not just minimizes $\ell_{0/1}(.)$ (which considers only the sign of arg), but a surrogate function of $yH(o)$.

When $yH(o) \leq 0$, wrong label.

Counting the event of wrong sign is apparently not enough to capture the “good” behaviour of boosting algorithms.
ANY OTHER “EVENT”

\[ \text{1(predicate}(yH(o)) \text{ WHOSE OCCURRENCE IS MINIMISED?)} \]
Many other events!

- Margin theories

- Analyse events $1_{yH(o)\leq\theta}$, with $\theta > 0$

- If the proportion of “bad” edges is small for “large” $\theta$, then the sign is right and there is a “form of” confidence in that classification (as measured by $|H(.)|$)

- How does it play in training / testing?
(Last slide of the MLSS course material)
The following holds, \( \forall \theta \):

\[
\mathbb{E}_S[\ell^{0/1}(yH_s(o) - \theta)] \leq 2^T \prod_t \sqrt{\mathbb{E}_S[\ell^{0/1}(y_{h_t}(o))]^{1-\theta}(1 - \ell^{0/1}(y_{h_t}(o)))^{1+\theta}}
\]

- So, if \( \mathbb{E}_S[\ell^{0/1}(y_{h_t}(o))] \leq (1/2) - \gamma \), for each \( t \), then

\[
\mathbb{E}_S[\ell^{0/1}(yH_s(o) - \theta)] \leq \left( \sqrt{(1 - 2\gamma)^{1-\theta}(1 + 2\gamma)^{1+\theta}} \right)^T
\]

- If \( \theta < \gamma \), then the quantity in parenthesis is < 1

- So, we can “boost” all bad edges “below” the weak learning threshold, and AdaBoost maximises a minimal “margin” \( yH(.) \)

\[
H(o) = \frac{1}{\sum_t \alpha_t} \cdot \sum_t \alpha_t h_t(o)
\]
**AdaBoost**

- **Proof:** \( \forall (o, y) \), if \( yH(o) \leq \theta \) then
  \[
  y \sum_t \alpha_t h_t(o) \leq \theta \sum_t \alpha_t
  \]

- **So,**
  \[
  \exp \left( -y \sum_t \alpha_t h_t(o) + \theta \sum_t \alpha_t \right) \geq 1
  \]

- **Compute the expectation and simplify**

  \[
  \mathbb{E}_S[\ell^{0/1}(yH_s(o) - \theta)] \leq \mathbb{E}_S[\exp(-y \sum_t \alpha_t h_t(o) + \theta \sum_t \alpha_t)]
  \]

  \[
  = \exp \left( \theta \sum_t \alpha_t \right) \cdot \prod_t Z_t(\alpha_t) 1^T u
  \]
Generalisation

Suppose $\text{VC}(\mathcal{H}_w) = d$. Then for all $\theta > 0$, and $m \geq d$,

$$
\mathbb{E}_D[\ell^{0/1}(y H_s(o))] \leq \mathbb{E}_S[\ell^{0/1}(y H_s(o) - \theta)] + O \left( \sqrt{ \frac{d \log^2 (m/d)}{m \theta^2} + \frac{1}{m} \log \frac{1}{\delta} } \right)
$$
Generalisation

Suppose $\text{VC}(\mathcal{H}_w) = d$. Then for all $\theta > 0$, and $m \geq d$

$$
\mathbb{E}_D[\ell^{0/1}(y_Hs(o))] \leq \mathbb{E}_S[\ell^{0/1}(y_Hs(o) - \theta)] + O\left(\sqrt{\frac{d \log^2(m/d)}{m \theta^2}} + \frac{1}{m} \log \frac{1}{\delta}\right)
$$

In AdaBoost, this is minimised exponentially fast
Generalisation

So, does maximising the minimum margin yields to better generalisation?
So, does maximising the minimum margin yields to better generalisation? No

There exists a minimum margin maximisation algorithm which obtains poor performances in generalisation

Breiman, 1999
So, does maximising the minimum margin yields to better generalisation?

There exists a minimum margin maximisation algorithm which obtains poor performances in generalisation.

Warning: Reyzin & Schapire, 2006
Generalisation

The experiments of Breiman used CART

...but...
Generalisation

Breiman did not control complexity of trees

VC($\mathcal{H}_w$ finite) $\leq \log |\mathcal{H}_w|$
More arguments

- The margin notion is appealing, since it puts "confidence" in the equation.

- The minimum margin argument, as well as many other statistical properties of boosting are questioned and debated (like controlling the complexity, overfitting, regularization, density estimation, ...).

- If the minimum margin does not work in general to explain the lack of overfitting, are there other "margin functions" that work? 

Mease & Wyner, 2008
Average margin

Replace min margin by functions of the cumulative margin distribution:

\[ \theta(q) = \sup\{ \theta : \mathbb{E}_S[\ell^{0/1}(yH_s(o) - \theta)] \leq q \} \]
Rademacher complexity

- Keep min margin criterion, but make the complexity-dependent parameters finer:

\[ \mathbb{E}_D[\ell^{0/1}(yH_s(o))] \leq \mathbb{E}_S[\ell^{0/1}(yH_s(o) - \theta)] + O \left( \frac{1}{\theta} \sum_t \alpha_t R_m(\mathcal{H}_{k_t}) + \frac{2}{\theta} \sqrt{\frac{\log p}{m}} + g(1/\theta, 1/\delta, m, p) \right) \]

- with \( \mathcal{H}_w = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \ldots \cup \mathcal{H}_p \)

- and \( R_m(\mathcal{H}_{k_t}) = \mathbb{E}_\sigma \left[ \sup_{h \in \mathcal{H}_{k_t}} \left| \frac{2}{m} \sum_i \sigma_i h(o_i) \right| \right] \) is the empirical Rademacher complexity of \( \mathcal{H}_{k_t} \), measuring the “richness” of \( \mathcal{H}_{k_t} \)

\[ R_m(\mathcal{H}_{k_t}) = O(\sqrt{1/m} \log \Pi_{\mathcal{H}_{k_t}}(m)) \]

\[ \Pi_{\mathcal{H}_{k_t}}(m) \leq (em/d)^d \]
**Conclusion**

- Boosting is a powerful methodology borne out of a learning model, (in part) because of the lack of positive results in this model.

- Valiant recently proposed another “learning” model.
Bibliography

L. Reyzin and R. Schapire. How boosting the margin can also boost classifier complexity. ICML 2006.
Questions ?