An Almost Optimal PAC Algorithm

Hans U. Simon
hans.simon@rub.de

Department of Mathematics and Horst Görtz Institute for IT-Security
Ruhr-University Bochum

Conference on Learning Theory — COLT 2015
Outline

1. The Problem and the Main Result
2. A New Policy for Choosing a Hypothesis
3. Proof of the Main Result
4. Final Remarks
PAC Learning and Sample Complexity

Bounds from [EHKV1989, BEHW1989]:

\[ m \geq \Omega \left( \frac{1}{\varepsilon} \left( d + \ln \left( \frac{1}{\delta} \right) \right) \right) \]
\[ m \leq O \left( \frac{1}{\varepsilon} \left( d \cdot \log(1/\varepsilon) + \ln(1/\delta) \right) \right) \]

This GAP had survived for 26 years but will be considerably narrowed within this talk!
Bounds from [EHKV1989, BEHW1989]:

\[
\begin{align*}
m & \geq \Omega \left( \frac{1}{\varepsilon} \left( d + \ln \left( \frac{1}{\delta} \right) \right) \right) \\
m & \leq O \left( \frac{1}{\varepsilon} \left( d \cdot \log(1/\varepsilon) + \ln(1/\delta) \right) \right)
\end{align*}
\]

This GAP had survived for 26 years but will be considerably narrowed within this talk!
Warmuth’s Open Problem from COLT 2004

Do there exist optimal PAC learners?

**Warmuth’s Conjecture:**
The 1-inclusion graph algorithm [HLW 1994] is optimal.

— Confirmed for intersection-closed classes [D2014] —
Warmuth’s Open Problem from COLT 2004

Do there exist optimal PAC learners?

**Warmuth’s Conjecture:**
The 1-inclusion graph algorithm [HLW 1994] is optimal.

— Confirmed for intersection-closed classes [D2014] —
Our Contribution

Theorem

Let $\log^{(K)}(z) = \underbrace{\log \ldots \log}_{K\text{-times}}(z)$ and let $\ell_K(z) = \max\{2, \log^{(K)}(z)\}$.

With this notation:
For every $K \geq 1$, there exists a PAC learner $L_K$ that needs only

$$m \leq O \left( \frac{1}{\varepsilon} \left( d \cdot \ell_K \left( \frac{1}{\varepsilon} \right) + \log \left( \frac{1}{\delta} \right) \right) \right)$$

labeled random examples (constants depending on $K$ hidden in the $O$-notation).
Our Contribution (continued)

Alternative statement of our main result:

**Theorem**

As for $L_K$, we have:

$$\varepsilon \leq O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \ell_K \left( \frac{m}{d} \right), \log \left( \frac{1}{\delta} \right) \right\} \right)$$

Previously best bound [BEHW1989]:

$$\varepsilon \leq \varepsilon_{ub}(m, d, \delta) \quad \overset{\text{def}}{=} \quad \frac{4}{m} \cdot \max \left\{ d \cdot \log \left( \frac{2em}{d} \right), \log \left( \frac{2}{\delta} \right) \right\}$$

$$= O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \log \left( \frac{m}{d} \right), \log \left( \frac{1}{\delta} \right) \right\} \right)$$
Our Contribution (continued)

Alternative statement of our main result:

**Theorem**

As for $L_K$, we have:

\[
\varepsilon \leq O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \ell_K \left( \frac{m}{d} \right) , \log \left( \frac{1}{\delta} \right) \right\} \right)
\]

Previously best bound [BEHW1989]:

\[
\varepsilon \leq \varepsilon_{ub}(m, d, \delta) \overset{\text{def}}{=} \frac{4}{m} \cdot \max \left\{ d \cdot \log \left( \frac{2em}{d} \right) , \log \left( \frac{2}{\delta} \right) \right\} = O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \log \left( \frac{m}{d} \right) , \log \left( \frac{1}{\delta} \right) \right\} \right)
\]
Outline

1. The Problem and the Main Result
2. A New Policy for Choosing a Hypothesis
3. Proof of the Main Result
4. Final Remarks
The Old and the New Policy

Classical Upper Bound in the PAC Model

version space of total sample $S$

pick any consistent hypothesis

$h_1 \rightarrow \cdots \rightarrow h_5$

$h$ optimal up to factor $\log(1/\eps)$

Almost Optimal Upper Bound in the PAC Model

version spaces wrt subsamples

$h_1 \rightarrow \cdots \rightarrow h_5$

$h$ pick any consistent hypothesis

$h_1 \rightarrow \cdots \rightarrow h_5$

majority vote

$h$ optimal up to factor

$\log \cdots \log(1/\eps)$

Hans U. Simon

An Almost Optimal PAC Algorithm
Let $L$ be a consistent and proper PAC learner. Let $S \in (X \times \{0, 1\})^{(2K-1)m}$ be a given sample. $L_K$ proceeds as follows:

1. Decompose $S$ into $2K - 1$ subsamples $S_1, \ldots, S_{2K - 1}$ of size $m$, respectively.
2. For $k = 1, \ldots, 2K - 1$, let $h_k = L(S_k)$.
3. Return the hypothesis $h = L_K(S)$ that always goes with the majority of $h_1, \ldots, h_{2K - 1}$.
Learner Transformation: from $L$ to $L_K$

Let $L$ be a consistent and proper PAC learner. Let $S \in (X \times \{0, 1\})^{(2^K-1)m}$ be a given sample. $L_K$ proceeds as follows:

1. Decompose $S$ into $2K - 1$ subsamples $S_1, \ldots, S_{2K-1}$ of size $m$, respectively.
2. For $k = 1, \ldots, 2K - 1$, let $h_k = L(S_k)$.
3. Return the hypothesis $h = L_K(S)$ that always goes with the majority of $h_1, \ldots, h_{2K-1}$.
Learner Transformation: from $L$ to $L_K$

Let $L$ be a consistent and proper PAC learner. Let $S \in (X \times \{0, 1\})^{(2^K-1)m}$ be a given sample. $L_K$ proceeds as follows:

1. Decompose $S$ into $2K - 1$ subsamples $S_1, \ldots, S_{2K-1}$ of size $m$, respectively.
2. For $k = 1, \ldots, 2K - 1$, let $h_k = L(S_k)$.
3. Return the hypothesis $h = L_K(S)$ that always goes with the majority of $h_1, \ldots, h_{2K-1}$.
Let $L$ be a consistent and proper PAC learner. Let $S \in (X \times \{0, 1\})^{(2K-1)m}$ be a given sample. $L_K$ proceeds as follows:

1. Decompose $S$ into $2K - 1$ subsamples $S_1, \ldots, S_{2K-1}$ of size $m$, respectively.
2. For $k = 1, \ldots, 2K - 1$, let $h_k = L(S_k)$.
3. Return the hypothesis $h = L_K(S)$ that always goes with the majority of $h_1, \ldots, h_{2K-1}$.
The Problem and the Main Result
A New Policy for Choosing a Hypothesis
Proof of the Main Result
Final Remarks

Outline

1. The Problem and the Main Result
2. A New Policy for Choosing a Hypothesis
3. Proof of the Main Result
4. Final Remarks
Learning algorithm \textit{and} analysis proceed in stages.

\begin{align*}
S_1 & \ldots & S_{k-1} \\
\downarrow & & \downarrow \\
h_1 & \ldots & h_{k-1} \\
\underbrace{\text{Event } E \subseteq X} & & \underbrace{P(x\mid E)} \\
S_k & \ldots & S_{2K-1} \\
\downarrow & & \downarrow \\
h_k & \ldots & h_{2K-1} \\
\rightarrow & & \text{maj. vote} \\
S_k \cap E & \sim & P^{(2K-1)m} \quad \text{(subsamples)} \\
& & \ \text{(algorithm)} \\
& & \ \text{(hypotheses)}
\end{align*}

Classical generalization error bounds wrt $P(x\mid E)$ do apply in stage $k$. 

Hans U. Simon

An Almost Optimal PAC Algorithm
A Central Feature of the Proof

Learning algorithm and analysis proceed in stages.

\[ S_1 \ldots S_{k-1} \downarrow \downarrow \]
\[ h_1 \ldots h_{k-1} \]

\[ S_k \downarrow \downarrow \]
\[ h_k \ldots h_{2K-1} \rightarrow \text{maj. vote} \]

Event \( E \subseteq X \)

\[ P(x|E) \]

\[ S_k \cap E \]

Classical generalization error bounds wrt \( P(x|E) \) do apply in stage \( k \).
Step 1: Decomposition of the Total Error Set

Notations:

- $E$ denotes the error set of the majority vote $h$.
- For $j = 1, \ldots, 2K - 1$, $E_j$ denotes error set of $h_j$.
- For $J \subseteq \{1, \ldots, 2K - 1\}$, let $E_J = \bigcap_{j \in J} E_j$.
- Let $M$ denote the family of subsets $\{1, \ldots, 2K - 1\}$ that have cardinality $K$.

With these notations:

$$E = \bigcup_{J \in M} E_J$$
Step 1: Decomposition of the Total Error Set

Notations:
- $E$ denotes the error set of the majority vote $h$.
- For $j = 1, \ldots, 2K - 1$, $E_j$ denotes error set of $h_j$.
- For $J \subseteq \{1, \ldots, 2K - 1\}$, let $E_J = \bigcap_{j \in J} E_j$.
- Let $M$ denote the family of subsets $\{1, \ldots, 2K - 1\}$ that have cardinality $K$.

With these notations:

$$E = \bigcup_{J \in M} E_J$$
Step 2: Decomposition of $P(E_J)$

W.l.o.g. $J = \{1, \ldots, K\}$.

Consider the following sequence of parameters:

$$
\varepsilon_k \overset{\text{def}}{=} P \left( \bigcap_{l=1}^{k} E_l \right) = \prod_{l=1}^{k} P \left( E_l \mid \bigcap_{l=1}^{k-1} E_l \right)
$$

Specifically:

$$
\varepsilon_K = P(E_J) = P \left( \bigcap_{k=1}^{K} E_k \right) = \prod_{k=1}^{K} P \left( E_k \mid \bigcap_{l=1}^{k-1} E_l \right)
$$

Subsample $S'_k$ of $S_k$ with points hitting $\bigcap_{l=1}^{k-1} E_l$ has expected size $\varepsilon_{k-1} m$. 

bound $\varepsilon_{ub}$ applies!
Step 2: Decomposition of $P(E_J)$

W.l.o.g. $J = \{1, \ldots, K\}$.

Consider the following sequence of parameters:

$$
\varepsilon_k \overset{\text{def}}{=} P(\bigcap_{l=1}^{k} E_l) = \prod_{l=1}^{k} P\left(E_l \mid \bigcap_{l=1}^{k-1} E_l\right)
$$

Specifically:

$$
\varepsilon_K = P(E_J) = P\left(\bigcap_{k=1}^{K} E_k\right) = \prod_{k=1}^{K} P\left(E_k \bigcap_{l=1}^{k-1} E_l\right)
$$

Subsample $S'_k$ of $S_k$ with points hitting $\bigcap_{l=1}^{k-1} E_l$ has expected size $\varepsilon_{k-1} m$. 

bound $\varepsilon_{ub}$ applies!
Step 3: Setting Up a Recursion on $\varepsilon_k$

Recall that $\varepsilon_K = P(E_J)$.
With high probability, the parameters $(\varepsilon_k)_{k=1,\ldots,K}$ evolve according to the following recursion:

1. $\varepsilon_1 \leq \varepsilon_{ub}(m, d, \delta)$.
2. $\varepsilon_k \leq \varepsilon_{k-1} \cdot \varepsilon_{ub}\left(\frac{1}{2}\varepsilon_{k-1} m, d, \delta\right)$. 
Step 3: Setting Up a Recursion on $\varepsilon_k$

Recall that $\varepsilon_K = P(E_J)$.

With high probability, the parameters $(\varepsilon_k)_{k=1,\ldots,K}$ evolve according to the following recursion:

1. $\varepsilon_1 \leq \varepsilon_{ub}(m, d, \delta)$.
2. $\varepsilon_k \leq \varepsilon_{k-1} \cdot \varepsilon_{ub}\left(\frac{1}{2}\varepsilon_{k-1} m, d, \delta\right)$. 
Recall that $\varepsilon_K = P(E_J)$.
With high probability, the parameters $(\varepsilon_k)_{k=1,\ldots,K}$ evolve according to the following recursion:

1. $\varepsilon_1 \leq \varepsilon_{ub}(m, d, \delta)$.
2. $\varepsilon_k \leq \varepsilon_{k-1} \cdot \varepsilon_{ub}\left(\frac{1}{2}\varepsilon_{k-1} m, d, \delta\right)$.
The solution of the recursion for the parameters $\varepsilon_k$ has the following order of magnitude:

$$\varepsilon_k \leq O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \ell_k \left( \frac{m}{d} \right), \log \left( \frac{1}{\delta} \right) \right\} \right)$$

Specifically:

$$\Pr(E_J) = \varepsilon_k \leq O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \ell_k \left( \frac{m}{d} \right), \log \left( \frac{1}{\delta} \right) \right\} \right)$$
Step 5: Putting Everything Together

Since the total error set satisfies $E = \bigcup_{J \subseteq M} E_J$, the following holds with high probability:

$$\Pr(E) \leq \binom{2K - 1}{K} \Pr(E_J) \leq O \left( \frac{1}{m} \cdot \max \left\{ d \cdot \ell_K \left( \frac{m}{d} \right), \log \left( \frac{1}{\delta} \right) \right\} \right)$$

This coincides with our main result (in its second, alternative, form).
Outline

1. The Problem and the Main Result
2. A New Policy for Choosing a Hypothesis
3. Proof of the Main Result
4. Final Remarks
Final Remarks

- Comparison to [Hanneke 2009]
- Efficiency Issues
- Open Question:
  Is $L_K$ an optimal PAC-learner for some choice of $K$?
Final Remarks

- Comparison to [Hanneke 2009]
- Efficiency Issues

Open Question:
Is $L_K$ an optimal PAC-learner for some choice of $K$?
Final Remarks

- Comparison to [Hanneke 2009]
- Efficiency Issues
- **Open Question:**
  Is $L_K$ an optimal PAC-learner for some choice of $K$?
Questions ?