Lower Bounds and Hardness Amplification for Learning Shallow Monotone Formulas

Vitaly Feldman  
IBM Almaden

Homin K. Lee  
UT Austin

Rocco A. Servedio  
Columbia
Example: Machine Translation

The woman bought the dress with cash.

La femme a payé la robe en liquide.
Example: Machine Translation

The woman bought the dress with cash.

La femme a payé la robe en liquide.
Example: Machine Translation

The woman bought the dress with cash.

La femme a payé la robe en liquide.
Example: Rule-based MT

The woman bought the dress with cash.

Syntactic/Semantic Ambiguity!
Example: Rule-based MT

The woman bought the dress with cash.

Syntactic/Semantic Ambiguity!
Example: Rule-based MT

The woman bought the dress with cash.

Syntactic/Semantic Ambiguity!
Example: Rule-based MT

La femme a acheté la robe avec les rubans.
La femme a payé la robe en liquide.

The woman bought the dress with cash.
A Brief History of MT

late 1970s

- Linguistic rules
- Semantics

- Translation examples
- Probabilities
A Brief History of MT

- Linguistic rules
- Semantics

- Translation examples
- Probabilities

late 1970s

“Ignorant,” yet successful!
Big Questions

- What are the limits of learning from examples?

- What are the limits of learning from statistical estimates of examples?
Our Results

The class of poly(n)-size, depth-3 monotone formulas is not SQ-learnable to error:

$$\frac{1}{(\log n)^c}.$$ 

The class of poly(n)-size, depth-4 monotone formulas is not SQ-learnable to error:

$$\frac{1}{2} - 2^{-(\log n)^c}.$$
Constant-depth Formulas

- arbitrary fan-in
- size = # of gates
- monotone = no negations
Statistical Query Model

\( f : X \rightarrow R \) target function

\( s(x, f(x)) \) efficiently computable statistic

\( E[s(x, f(x))] \pm \tau \) output of SQ\(_f(s,\tau)\) oracle

\( h : X \rightarrow R, \Pr[f \neq h] < \varepsilon \) hypothesis of learner

[Kearns-93]
Statistical Query Model

• Every* learning algorithm can be couched as an SQ algorithm.
  
  e.g., Perceptron, Winnow, Fourier coefficient approximations, L1-polynomial regression, PCA, ID3 etc.

• SQ algorithms are noise tolerant.

• The SQ model has unconditional lower bounds.

* except those using Gaussian elimination
SQ Dimension

\[ \text{SQdim}(C) = \max d \text{ s.t. } \{ f_1, f_2, \ldots, f_d \} \subseteq C, \quad \forall i \neq j, \ E[f_i \cdot f_j] < 1/d \]

\text{SQdim}(C) \text{ characterizes weak SQ learning.}
Parity

\[ C_n = \{ \text{PAR}_S : S \subseteq [n] \} \]

\[ \text{PAR}_S(x) = \bigoplus_{i \in S} x_i \]

\[ \text{SQdim}(C_n) = 2^{\Omega(n)} \]

[Kearns-93]
Known Results

non-mono poly\((n)\)-size depth-\(d\) formulas

cryptographic hardness

[DLMSWW-08]
<table>
<thead>
<tr>
<th>Known Results</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>non-mono poly($n$)-size depth-$d$ formulas</td>
<td>cryptographic hardness</td>
</tr>
<tr>
<td>non-mono poly($n$)-size depth-2 formulas</td>
<td>SQdim = $n^{\Omega(\log n)}$</td>
</tr>
</tbody>
</table>
## Known Results

<table>
<thead>
<tr>
<th></th>
<th>[DLMSWW-08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-mono poly$(n)$-size</td>
<td>cryptographic hardness</td>
</tr>
<tr>
<td>depth-$d$ formulas</td>
<td></td>
</tr>
<tr>
<td>non-mono poly$(n)$-size</td>
<td>SQdim= $n^{\Omega(\log n)}$</td>
</tr>
<tr>
<td>depth-2 formulas</td>
<td></td>
</tr>
<tr>
<td><em>All</em> monotone functions</td>
<td>SQ-learnable to error $\frac{1}{2} - \Omega(\log n)/\sqrt{n}$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[KLV-87,BBL-98,OW09]</td>
</tr>
</tbody>
</table>
## Known Results

<table>
<thead>
<tr>
<th>Type of Formulas</th>
<th>Cryptographic Hardness</th>
<th>SQ-dim</th>
<th>SQ-learnable to Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-mono poly(n)-size depth-(d) formulas</td>
<td></td>
<td>(n^{\Omega(\log n)})</td>
<td>(\frac{1}{2} - \Omega(\log n) / \sqrt{n})</td>
</tr>
<tr>
<td>Non-mono poly(n)-size depth-2 formulas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All monotone functions</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What about \textit{strong} SQ learning of monotone functions?
Strong SQ Dimension

$$\text{SSQdim}(C, \varepsilon) = \max_g \text{SQdim}(C \setminus \text{B}(\text{sgn}(g), \varepsilon) - g$$

$$\text{SSQdim}(C)$$ characterizes strong SQ learning.
Strong SQ Dimension

$SSQdim(C, \varepsilon) = \max_g SQdim(C \setminus B(\text{sgn}(g), \varepsilon) - g)$

$SSQdim(C)$ characterizes strong SQ learning.

$[\text{Simon-07, Szörényi-09, Feldman-09}]$
Strong SQ Dimension

$f : \{0,1\}^n \rightarrow \{-1,+1\}$

$SSQdim(C, \varepsilon) = \max_g SQdim(C \setminus B(\text{sgn}(g), \varepsilon) - g)$

$g : \{0,1\}^n \rightarrow [-1,+1]$ \n
$SSQdim(C)$ characterizes strong SQ learning.

[Simon-07, Szörényi-09, Feldman-09]
Strong SQ Dimension

\[ f : \{0,1\}^n \rightarrow \{-1,+1\} \]

\[ \text{SSQdim}(C, \varepsilon) = \max_g \text{SQdim}(C \setminus B(\text{sgn}(g), \varepsilon) - g) \]

\[ g : \{0,1\}^n \rightarrow [-1,+1] \]

\[ h : \text{Pr}[h \neq \text{sgn}(g)] < \varepsilon \]

\[ \text{SSQdim}(C) \text{ characterizes strong SQ learning.} \]

[Simon-07,Szörényi-09,Feldman-09]
Strong SQ Dimension

\[ \text{SSQdim}(C, \varepsilon) = \max_g \text{SQdim}(C \setminus B(\text{sgn}(g), \varepsilon) - g) \]

\[ f : \{0,1\}^n \rightarrow \{-1,+1\} \]

\[ f-g : f \in C \setminus B(...) \]

\[ g : \{0,1\}^n \rightarrow [-1,+1] \]

\[ h : \Pr[h \neq \text{sgn}(g)] < \varepsilon \]

SSQdim(C) characterizes strong SQ learning.

[Simon-07, Szörényi-09, Feldman-09]
First Result

There exists a class $C$ of poly($n$)-size, depth-3 monotone formulas with:

$$\text{SSQdim}(C, 1/(\log n)^c) = n^{\omega(1)}.$$
Slice Functions

$C$ will be a class of slice funcs over the first $m$ vars:

\[
\text{slice}(\text{PAR}_S) = \begin{cases} 
+1 & \text{if } |x| > m/2 \\
\text{PAR}_S & \text{if } |x| = m/2 \\
|S| \leq m & -1 \text{ if } |x| < m/2
\end{cases}
\]

$C$ can be computed by poly$(n)$-size depth-3 circuits.
Slice Functions

\[
\text{SSQdim}(C, \epsilon) = \max_g \text{SQdim}(C \setminus B(\text{sgn}(g), \epsilon) - g)
\]

\[
g = \text{slice}(0) \quad \epsilon = o(1/\sqrt{m}) \Rightarrow C \setminus B(\text{sgn}(g), \epsilon) = C
\]

For \( m = \log^{2-c} n \), \( \text{SQdim}(C - g) = n^{\Theta(\log^{1-c} n)} \).
Slice Functions

$$SSQ\dim(C, \varepsilon) = \max_g SQ\dim(C \setminus B(\text{sgn}(g), \varepsilon) - g)$$

$$g = \text{slice}(0) \quad \varepsilon = o(1/\sqrt{m}) \implies C \setminus B(\text{sgn}(g), \varepsilon) = C$$

For $$m = \log^{2-c} n$$, $$SQ\dim(C - g) = n^{\Theta(\log^{1-c} n)}.$$
The class of poly($n$)-size, depth-4 monotone formulas is not SQ-learnable to error:

$$\frac{1}{2} - 2^{-(\log n)^c}.$$
XOR Lemma

If $f$ is hard to compute on a small fraction of inputs,

$$f^\oplus = \oplus(f, f, \ldots, f)$$

is hard on almost all inputs.

[Yao-82]
Hardness Amplification

**Thm [O’Donnell-02]:** If no poly-size circuit can compute \( f \) on a \((1-\delta)\) fraction of inputs, then no poly-size circuit can compute \( f^K \) on a \( \frac{1}{2} + \frac{1}{2} \text{NoiseStab}_\delta(K)^{\frac{1}{2}} \) fraction of inputs.

- \( f^K = K(f,f,\ldots,f) \)
- \( \text{NoiseStab}_\delta(K) = E[ K(x) \cdot K(x+\eta) ] \)
  (each \( \eta_i \) set to 1 with prob. \( \delta \))
Hardness Amp for Learning

**Thm**: If no poly-time alg. can learn $C$ to accuracy $(1-\delta)$, then no poly-time alg. can learn $C^K$ to accuracy better than:

$$\frac{1}{2} + \frac{1}{2}\text{NoiseStab}_\delta(K)^{\frac{1}{2}}.$$  

$C^K = \{f^K, f \in C\}$

Generalizes: [BonehLipton-03,DLMSWW-08]
Hardness Amp for SQ

**Thm:** If no poly-time **SQ**-alg. can learn $C$ to accuracy $(1-\delta)$, then no poly-time **SQ**-alg. can learn $C^K$ to accuracy better than:

$$\frac{1}{2} + \frac{1}{2} \text{NoiseStab}_{\delta}(K)^{\frac{1}{2}}.$$

$C^K = \{f^K, f \in C\}$
Picking a Combiner

XOR (⊕) has very low noise stability.

... but \( f \oplus \) is not a monotone function.
Monotone Combiner

Let $K = \text{Tribes CNF} + \text{Talagrand CNF}$

$\text{NoiseStab}_{1/\sqrt{m}}(K) = (m/k)^c$

[McMullinO’Donnell-03]
The Depth-4 Construction

Take $k$ copies of slice(PAR) on $m$ inputs.
## Summary

<table>
<thead>
<tr>
<th>Monotone Formulas of Depth:</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(\log n)$</td>
<td>OWF: $\varepsilon &gt; \frac{1}{2} - o(1)$ [DLMSWW-08]</td>
</tr>
<tr>
<td>4</td>
<td>SQ: $\varepsilon &gt; \frac{1}{2} - o(1)$</td>
</tr>
<tr>
<td>3</td>
<td>SQ: $\varepsilon &gt; 1/(\log n)^c$</td>
</tr>
<tr>
<td>?</td>
<td>2</td>
</tr>
</tbody>
</table>