Boolean Functions: 1,2,3 and more

Debajyoti Bera

Indraprastha Inst. of Information Technology (IIIT-Delhi)

dbera@iiitd.ac.in

12th January, 2019
Outline

1. Boolean functions
2. Query complexity
3. Complexity of $DJ$
4. Quantum vs. Classical
What is a Boolean function?

Mapping from $\{0, 1\}^* \rightarrow \{0, 1\}^*$

Can be used to model (almost) any mapping (over discrete domain and range)

This talk is about $m = 2^n$-bit to 1-bit Boolean functions.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f_1(x)$</th>
<th>$f_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1101</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>0001</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>1101</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Where to find them?

Schematic of Rosenblatt’s perceptron.
Examples

Consider $m$-to-1 bit $f()$.

$x = x_1 \ldots x_m$ and $\|x\| = m$ (number of bits).

$|x|$ denotes number of ones in $x$.

Total functions

$OR(x) = 1$ if any bit of $x$ is 1, $= 0$ otherwise

$AND(x) = 1$ if all bits of $x$ are 1, $= 0$ otherwise

$PARITY(x) = 1$ if odd number of bits of $x$ are 1, $= 0$ otherwise

$MAJORITY(x) = 1$ if more than $\frac{m}{2}$ bits of $x$ are 1

$Threshold_k(x) = 1$ if at least $k$ bits of $x$ are 1

$INDEX(x) = 1$ if $x$ can be written as $x = y \cdot z$ s.t.

- $\|y\| = \lceil \log_2(\|z\|) \rceil$
- $z_{int}(y) = 1$

Example: $INDEX(10 \cdot 1001) = 0$ and $INDEX(11 \cdot 0101) = 1$
Examples

Promise problems: Domain of $f$ is strict subset of $\{0, 1\}^*$

**Partial functions**

**DJ:** Given that $|x| \in \{0, \frac{n}{2}, n\}$, $DJ(x) = 0$ if $|x| = \frac{n}{2}$

**EWDP:** Given that $|x| \in \{k_l, k_u\}$, $EWDP_{k_l, k_u}(x) = 1$ if $|x| = k_u$

**WDP:** Given that $|x| \leq k_l$ or $|x| \geq k_u$, $EWDP_{k_l, k_u}(x) = 1$ if $|x| \geq k_u$
Boolean functions
Query complexity
Complexity of DJ
Quantum vs. Classical

Function problems

Consider inputs of length $2^n$.

$$x = \langle g(00\ldots0), g(00\ldots1), \ldots, g(11\ldots1) \rangle$$

Querying $x_j$ is same as querying $g$(bit repr. of $j$)

e.g. $x_5 \equiv g(0101)$

Given that $g(y) = a \cdot y \oplus b$, output $a, b$.

Given $b \in \{0, 1\}^n$, $\rho \in [0, 2^n]$, determine if $|\sum_y (-1)^{g(y) \oplus b \cdot y}| \geq \rho$

<table>
<thead>
<tr>
<th>$y$</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g(y)$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$(-1)^g(y)$</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>$(-1)^{g(y) \oplus (01) \cdot y}$</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Input \( g(y) \) denotes the “edge” function of a graph \( G \).

Suppose \( g(y) \) denotes the “edge” function of a graph \( G \).

\[ g(y_{\frac{n}{2}} \cdots y_{\frac{n}{2}} + 1 \cdots y_n) : \text{is there an edge between vertex } y_{\frac{n}{2}} \cdots y_{\frac{n}{2}} \text{ and vertex } y_{\frac{n}{2}} + 1 \cdots y_n \text{ of a } G \text{ with } 2^{n/2} \text{ vertices?} \]

Does \( G \) have a Hamiltonian circuit?
Outline

1. Boolean functions
2. Query complexity
3. Complexity of $DJ$
4. Quantum vs. Classical
Decision tree complexity

Input: $m$-bit $\{0, 1\}^*$-string $x$, or

$n$-bit Boolean function $g : \{0, 1\}^n \rightarrow \{0, 1\}$ (denote $2^n$ by $m$)

**Query Complexity**

$= \text{largest number of bits/g() queried on any input}$

Relevant when querying is “costlier” than local operations.

Trivially, query complexity is at most $m$.

Decision tree for $\text{VERIFY}_4(x)$
Query complexity of $INDEX$

$INDEX(x) = 1$ if $x$ can be written as $x = y \cdot z$ s.t. $z$ represents a “database” and $y$ is a position of database entry $y$th bit of $z$ should be 1

$|y| = \log_2(m)$

Query complexity $= 1 + \log_2(m)$
Quantum query complexity

Intermediate measurements can modify state.
So, measure only at end!
Error in computing Boolean functions

\[ X_0 : \text{set of inputs for which } f(x) = 0 \]
\[ X_1 : \text{set of inputs for which } f(x) = 1 \]

**Bounded error algorithms**

\[
\begin{align*}
\text{For } x \in X_0, & \quad \Pr[\text{Algo}(x) \to 1] \leq \rho_l < 1/2 \\
\text{For } x \in X_1, & \quad \Pr[\text{Algo}(x) \to 0] \leq \rho_u < 1/2
\end{align*}
\]

More chance of being correct than wrong.
Boolean functions
Query complexity
Complexity of DJ
Quantum vs. Classical

How good (or bad) is quantum?

\[ D(f) : \text{optimal } \# \text{bits queried by deterministic classical algo.} \]
\[ R(f) : \text{optimal } \# \text{bits queried by bounded-error classical ...} \]
\[ Q_E(f) : \text{optimal } \# \text{query-gates in “exact” quantum algo.} \]
\[ Q_2(f) : \text{optimal } \# \text{query-gates in bounded-error quantum ...} \]

Questions?

For a given \( f \), what are \( D(f) \), \( Q_E(f) \) and \( Q_2(f) \)?
Is there a general relationship between them?
Are there functions for which \( Q_E(f) \ll D(f) \)?
Are there functions for which \( Q_2(f) \ll R(f) \)?
Are there functions for which \( Q_E(f) \ll R(f) \)?

Input functions could be total or partial
Reducing algorithm error

Technique of choice: Repeat algorithm several times

\[ \Pr[\text{observing } |1\rangle] = \sin^2 \theta \]

Amplitude Amplification (AA)

\[ \Pr[\text{observing } |1\rangle] = \sin^2[\theta + 2t\theta] \]

2t can be replaced by rt for any fixed \( r \leq 2 \)
Using amplitude ampl. to reduce error

<table>
<thead>
<tr>
<th># calls to $U_x$</th>
<th>prob. of success</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/100</td>
</tr>
<tr>
<td>3</td>
<td>9/100</td>
</tr>
<tr>
<td>7</td>
<td>41/100</td>
</tr>
<tr>
<td>11</td>
<td>80/100</td>
</tr>
<tr>
<td>15</td>
<td>99.5/100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># calls to $U_x$</th>
<th>prob. of success</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/100</td>
</tr>
<tr>
<td>3</td>
<td>9/100</td>
</tr>
<tr>
<td>7</td>
<td>41/100</td>
</tr>
<tr>
<td>11</td>
<td>80/100</td>
</tr>
<tr>
<td>15</td>
<td>99.5/100</td>
</tr>
</tbody>
</table>

prob. of success $= 1/4 = \sin^2 30$

<table>
<thead>
<tr>
<th># calls to $U_x$</th>
<th>prob. of success</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/4</td>
</tr>
<tr>
<td>2</td>
<td>1 !!!</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># calls to $U_x$</th>
<th>prob. of success</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/2</td>
</tr>
<tr>
<td>2</td>
<td>1 !!!</td>
</tr>
</tbody>
</table>

prob. of success $= 1/2 = \sin^2 45$
Outline

1. Boolean functions
2. Query complexity
3. Complexity of $D.J$
4. Quantum vs. Classical
Complexity of $DJ$: $|x| = m/2$ or $|x| \in \{0, m\}$

Consider $m$-bit $y$ s.t. $y_i = x_i \oplus x_0$.

$|x| = m/2$ same as $|y| = m/2$ vs. $|x| \in \{0, m/2\}$ same as $|y| = 0$

Querying $y_i$ requires two queries to $x$

Apply AA to increase prob. of $|1\rangle$ (Bera, 2015)

If $|y| = 0$, then observe $|1\rangle$ with prob. 0.
After AA, observe $|1\rangle$ with prob. 0.

If $|y| = m/2$, then observe $|1\rangle$ with prob. 1/2.
After AA using 1 iteration, observe $|1\rangle$ with prob. 1.
Complexity of $DJ$

zero-error quantum algorithm using 6 query-gates

Quantum vs. Classical
zero-error deterministic algorithm needs $\frac{m}{2} + 1$ queries
6 query randomized algorithm has at least $1/2^{12}$ prob. of error

Deutsch-Jozsa’s circuit (Deutsch-Jozsa, 1992)
zero-error quantum algorithm using 1 query gate !!!
Outline

1. Boolean functions
2. Query complexity
3. Complexity of $DJ$
4. Quantum vs. Classical
Separation for partial functions

Deutsch-Jozsa’s problem ($DJ$)
- efficiently solvable by quantum circuit (exactly in constant queries)
- exact classical algo is inefficient (50% bits are queried)
- bounded-error classical algo is moderate (error $1/2^{2k}$ with $k$ queries)

Simon’s problem
- efficiently solvable by quantum circuit (bounded-error requires $\log m$ queries)
- bounded-error classical algo is bad (requires $\Omega(m)$ queries)
Separation for total functions

**Classical is not too worse *vis-a-vis.* quantum**

- Any $q$-query exact quantum algo.  
  $\implies q^3$-query exact classical algo.
- Any $q$-query bounded-error quantum algo.  
  $\implies q^6$-query exact classical algo.

**Quantum is somewhat better than classical**

- *OR:* bounded-error quantum makes $\sqrt{m}$ queries whereas bounded-error classical makes $m/2$ queries
- There exists an $f$ for which bounded-error classical algo. makes nearly $q^3$-queries whereas a bounded-error quantum algo. makes $q$ queries
Thank you

Questions?

Problems to work on . . .

- Optimal quantum query complexity of combinatorial (NP-complete, . . .) problems
- Fancy uses and extensions of AA (in ML?)
- Characterize query complexity based on degree, sensitivity, etc. properties of a Boolean function
- Find total functions for which quantum algo. makes $q$ queries and optimal bounded-error algo. makes $\geq q^4$ queries
- Quantum circuit complexity of problems

Join PhD@ IIIT-Delhi, New Delhi