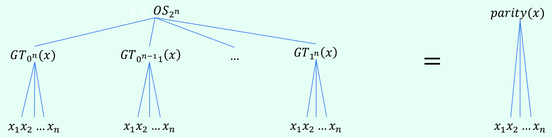
Approximate degree lower bounds for oracle identification problems

Mark Bun, Nadezhda Voronova
Boston University

Ordered search

Ordered search problem OS_{2^n}:
Given: a string $0^x\ 1^{2^n-x}$
Goal: output x
Could be solved in $O(n)$ exactly using binary search.

Decision version: output $parity(x)$, that is the XOR of bits of x written in binary notation



$GT_i(x) = 1$ if and only if $i > x$

Approximate degree

Approximate degree
Let $f : D \to \{0,1\}$ where $D \subseteq \{0,1\}^n$. A polynomial $p\colon \{0,1\}^n \to \mathbb{R}$ is an ε-**approximation** to f if
- $-\varepsilon \le p(x) \le 1+\varepsilon$ for all $x \in \{0,1\}^n$, and
- $|p(x) - f(x)| < \varepsilon$ for $x \in D$.

Approximate degree $\widetilde{deg}_\varepsilon(f)$ of f is the least degree of a polynomial that approximates f to error ε.

Why consider?
- "Proto-complexity" model
- Can prove unconditional lower bounds
- Related to learning theory, differential privacy, query complexity, communication complexity, circuit complexity, cryptography, etc.

Our main result

- New framework for proving lower bounds on $\widetilde{deg}$
- By applying this framework, we obtained:

For any $\frac{1}{2} > \gamma > 0$ (that could depend on n)

$$\widetilde{deg}_{\frac{1}{2}-\gamma}(OS_{2^n}) = \Omega\left(\frac{n}{\log^2 n} - \log\frac{1}{\gamma}\right)$$

In particular, for $\gamma > 2^{-n^{.99}}$

$$\widetilde{deg}_{\frac{1}{2}-\gamma}(OS_{2^n}) = \Omega(\frac{n}{\log^2 n})$$

Previous state-of-the-art: $\widetilde{deg}_{\frac{1}{2}-\gamma}(OS_{2^n}) = \Omega\left(\sqrt{n} - \log\frac{1}{\gamma}\right)$ by [BdW99]

References

Or scan this QR-code:

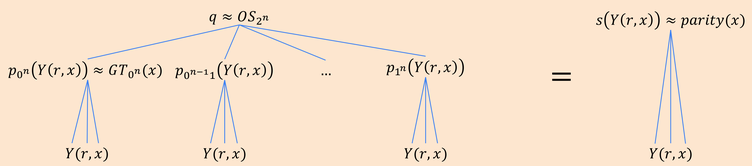
[BdW99] Harry Buhrman Ronald de Wolf. A lower bound for quantum search of an and ordered list.
[She12] Alexander A Sherstov. Making polynomials robust to noise.
[CIG+12] Richard Cleve, Kazuo Iwama, Francois Le Gall, Harumichi Nishimura, Seiichiro Tani, Junichi Teruyama, and Shigeru Yamashita. Reconstructing strings from substrings with quantum queries.

Our framework

Goal: Find such distribution $Y(r,x)$ pameterized by a sample $r \leftarrow R$ such that:

Claim 1. With probability $\frac{2}{3}$ over $r \leftarrow R$ for all $i \in \{0,1\}^n$ there exists a polynomial of $Y(r,x)$ of degree $O(\log n)$ that $\frac{1}{3}$-approximates $GT_i(x)$.

Claim 2. With probability $\frac{2}{3}$ over $r \leftarrow R$ every polynomial of $Y(r,x)$ that ε-approximates $parity(x)$ requires degree $\Omega(\frac{n}{\log n})$.



Claim 1 + Claim 2 + [She12] =

Theorem:
For any $\frac{1}{2} > \gamma > 0$ (that could depend on n)

$$\widetilde{deg}_{\frac{1}{2}-\gamma}(OS_{2^n}) = \Omega(\frac{n}{\log^2 n} - \log\frac{1}{\gamma})$$

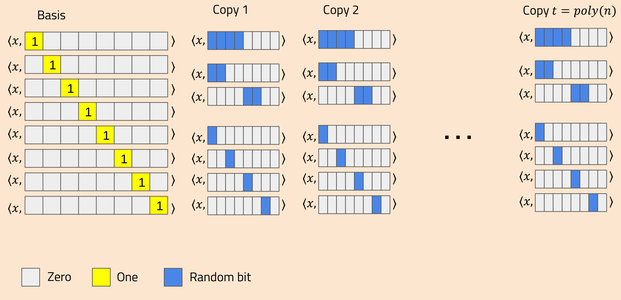
Proof by contradiction

Inspired by [BdW99]

Upper bound (Claim 1)

We choose $Y(r,x)$ to be a function that takes a sample $r \leftarrow R$ which is a collection of n-bit strings, and x and outputs inner products of x with each of the n-bit strings:

$$Y(r,x)_i = \langle r_i, x\rangle \bmod 2$$
$$r_i \in \{0,1\}^n$$
$$r \leftarrow R, r = (r_1, r_2, \dots r_m)$$



Our approach is based on:
- *Communication protocol for Greater-Than*
- *Newman's theorem on private vs public randomness in communication complexity*
- *Fact: every randomized T-query algorithm with bounded error implies a polynomial of degree T with bounded error*

Algorithm:

Input: $i \in \{0,1\}^n$, oracle access to $Y(r,x)$
Goal: compute $GT_i(x)$

1. Sample $j \leftarrow [t]$ u.a.r
2. Simulate communication protocol using inner products from copy j
3. Query most significant bit using basis and compare it to the value in i

The algorithm above computes $GT_i(x)$ with small error with high probability over the sample $r \leftarrow R$.
Can be converted into an approximating polynomial.

Lower bound (Claim 2)

- **Proof idea 1:** We could consider polynomials over $\{-1,1\}$ instead of $\{0,1\}$
- **Proof idea 2:** All monomials of degree $< n$ are orthogonal to parity in $\{-1,1\}$ basis.
- **Proof idea 3:** Getting a monomial of x of degree n by multiplying $\frac{n}{\log n}$ bits of $Y(r,x)$ is improbable.

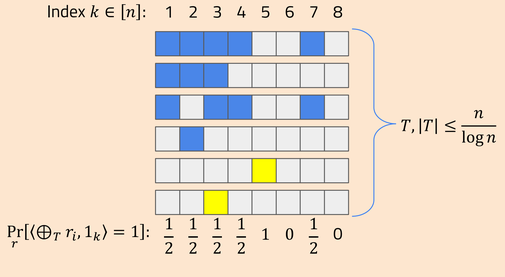
Generalization:
This result is generalizable for any R with a good structure!

Good structure: all bits in $Y(r,x)$ are inner products of x and n-bit strings where each bit is either zero or independent unbiased random bit, except for small number of inner products of x with basis strings

Generalized result:
Approximating parity of x to error $\frac{1}{2} > \varepsilon > 0$ by a polynomial of $Y(r,x)$ requires $\Omega(\frac{n}{\log(size\ of\ oracle\ Y)})$ degree .

Key lemma:
Getting the all-ones string by taking bit-wise XOR of $\frac{n}{\log n}$ strings from a sample $r \leftarrow R$ is improbable over the choice of R.

Proof:



Another application of framework: Hidden string

Hidden string problem $HS_{2^{n+1}}$:
Given: collection of bits $\tau_s(x)$, where $\tau_s(x) = 1$ if and only if s is a substring of hidden $x \in \{0,1\}^n$
Goal: output x
Could be approximated to constant error by $O(n)$-degree polynomial (implied by [CIG+12]).

Decision version: output $parity(x)$, that is the XOR of bits of x written in binary notation

Our results:
Using the same framework for an intermediate problem and a reduction from that problem to $HS_{2^{n+1}}$, we obtain the following:

For any $\frac{1}{2} > \gamma > 0$ (that could depend on n)

$$\widetilde{deg}_{\frac{1}{2}-\gamma}(HS_{2^{n+1}}) = \Omega(\frac{n}{\log^2 n} - \log\frac{1}{\gamma})$$

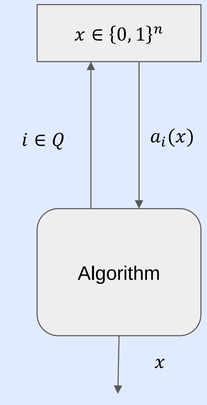
Query complexity of oracle identification problems

Query model
Given: oracle access to $x \in \{0,1\}^n$
Goal: compute $f(x)$
Cost: # of queries

Models:
- Deterministic
- Randomized: access to randomness
- Quantum: query in superposition

Error regimes:
- Bounded error: correct with $\Pr \ge 2/3$
- Unbounded error: correct with $\Pr \ge \frac{1}{2} + \gamma$

Why consider?
- Strips away implementation details.
- Can prove unconditional lower bounds!



Polynomial method for lower bounds:

The approximate degree to error ε of a function is always at most half of its quantum query complexity with error ε.

Quantum query complexity results

Regime	Problem	Previous work	This work (decision problem)
Unbounded error $\le \frac{1}{2} - \gamma$	Ordered search	$\Theta\left(n - \log\frac{1}{\gamma}\right)$	$\Omega(\frac{n}{\log^2 n} - \log 1/\gamma)$
	Hidden string	$O\left(n - \log\frac{1}{\gamma}\right)$, $\Omega(\gamma^2 n/\log^2 n)$	$\Omega(\frac{n}{\log^2 n} - \log 1/\gamma)$
Bounded error $\le \frac{1}{3}$	Ordered search	$\Theta(n)$	$\Omega(\frac{n}{\log^2 n})$
	Hidden string	$O(n)$, $\Omega(n/\log^2 n)$	$\Omega(\frac{n}{\log^2 n})$