Quantum Property Testing

Harry Buhrman (CWI, UvA)
Lance Fortnow (NEC)
Ilan Newman (Haifa)
Hein Röhrig (CWI)

Property Testing

In classical Algorithms the typical decision problems is:

For a fixed property $\mathcal{P}$ and a given input x, decide whether x belongs to $\mathcal{P}$ or not.

Sometimes we don't care about an exact answer as there is a 'gray area', or, sometimes, there is not enough time for exact decision. Then the following 'approximation' may be used:

decide whether x has $\mathcal{P}$ or it is very 'far' from having $\mathcal{P}$.

Examples

- **'Decide whether the used car I am going to buy is running OK'.**
- **'Decide whether the home page I am at, is relevant to my needs'.**

Examples

- **’Decide whether the used car I am going to buy is running OK’.**
- **’Decide whether the home page I am at, is relevant to my needs’.**
- **Working with huge data, e.g. genome data, WWW.:**
  - **difficult to store.**
  - **cannot look at entire input.**

Examples

- **'Decide whether the used car I am going to buy is running OK'.**
- **'Decide whether the home page I am at, is relevant to my needs'.**
- **Working with huge data, e.g. genome data, WWW.:**
  - **difficult to store.**
  - **cannot look at entire input.**
- **'Decide if the election was won by A (versus B)'.**

Property Testing - definitions

We encode inputs as strings; $x \in \{0,1\}^n$ and a property is just a collection of inputs (these that have the property). Namely, $\mathcal{P} \subseteq \{0,1\}^n$.

Being far: **is measured by hamming distance, namely** $dist(x,y) = |\{i|\ x_i \neq y_i\}|$ **and** $dist(x,\mathcal{P}) = min_{y \in \mathcal{P}} dist(x,y)$.

Property Testing - definitions

We encode inputs as strings; $x \in \{0,1\}^n$ and a property is just a collection of inputs (these that have the property). Namely, $\mathcal{P} \subseteq \{0,1\}^n$.

Being far: **is measured by hamming distance, namely** $dist(x,y) = |\{i|\ x_i \neq y_i\}|$ **and** $dist(x,\mathcal{P}) = min_{y \in \mathcal{P}} dist(x,y)$.

For a $\epsilon < 1$ we say that x is ϵ-far form $\mathcal{P}$ if $dist(x,\mathcal{P}) \geq \epsilon n$.

ϵ-Tests

An (ϵ, q)-test for a fixed property $\mathcal{P}$ is a randomized algorithm that for unknown input x queries at most q bits of x and:

- **If $x \in \mathcal{P}$ then the algorithm accepts it with probability $\geq 2/3$.**
- **If x is ϵ-far from $\mathcal{P}$ then it gets rejected with probability $\geq 2/3$.**

ϵ-Tests

An (ϵ, q)-test for a fixed property $\mathcal{P}$ is a randomized algorithm that for unknown input x queries at most q bits of x and:

- **If $x \in \mathcal{P}$ then the algorithm accepts it with probability $\geq 2/3$.**
- **If x is ϵ-far from $\mathcal{P}$ then it gets rejected with probability $\geq 2/3$.**

Interested: $q = o(n)$, better yet $q = poly(\log n)$ or even better $q = O(1)$.

A concrete Example

Given: **a list** $x_1, x_2, ..., x_n$ **of Integers.**

Property: **The list is sorted,** $x_1 \leq x_2 \leq ... \leq x_n$**.**

Require $\Omega(n)$ **time (= queries) for probabilistic algorithms.**

Can be done by $\Theta(\sqrt{n})$ **quantum algorithm.**

Approximation

Given: a list $x_1, x_2, ..., x_n$ of Integers.

Question: Is the list (almost) sorted, i.e, can change at most ϵ fraction of the numbers to make it sorted.

Can test in $O(1/\epsilon \cdot \log n)$ queries, [Ergun, Kannan, Kumar, Rubinfeld, Viswanathan 2000, Fischer 2001].

Background

Property testing was first defined by Rubinfeld and Sudan [96] who were mainly motivated by the connection to program checking.

The study of this object for combinatorial objects (mainly for graph properties) was introduced by Goldreich, Goldwasser and Ron [96], pointing the connection to approximation algorithms, PAC learning, PCP, etc.

Goldreich et al. showed that the graph property of being bipartite is testable in $O(1)$ queries.

Since then property testing became a very active area with many interesting results.

Classically Testable Properties

- **Linearity test** ($\forall x, y,\ f(x) + f(y) = f(x + y)$) [Blum, Luby and Rubinfeld 93, Bellare, Coppersmith, Hastad, Kiwi and Sudan 95].
- **Graph Properties**—colorability, not containing a forbidden subgraph, connectivity, acyclicity, rapidly mixing, max cut, ... [Goldreich, Goldwasser and Ron 87, Alon, Fischer, Krivelevich and Szegedy 99, Parnas and Ron 99, Bender and Ron 2000, Fischer 2001, Alon 2001]....
- **Monotonicity** [Goldreich, Goldwasser, Lehman, Ron, Dodis, Raskhodnikova and Samorodnitsky 99, Lehman, Fischer, Newman, Rubinfeld, Raskhodnikova and Samorodnitsky 2002 ..].
- **Set properties**—equality, distinctness, ... [Ergun, Kannan, Kumar, Rubinfeld and Viswanathan 98..].
- **Geometric properties**—metrics, clustering, convex hulls,... [Parnas and Ron 99, Alon, Dar, Parnas and Ron 2000, Czumaj and Sohler 2002...].

- **Membership in low-complexity languages—regular languages, constant-width branching programs, context-free languages [Alon, Krivelevich, Newman, and Szegedy 99, ...].**

Quantum Circuits / Algorithms

We think of an algorithm as a state transformer.

Classical algorithm:

- **A state is a bit-vecor; values of all variables / intermidiate gates.**
- k **variables - states are vecots in** F_2^k**.**

Quantum Circuits / Algorithms

We think of an algorithm as a state transformer.

Classical algorithm:

- **A state is a bit-vecor; values of all variables / intermidiate gates.**
- k **variables - states are vecots in** F_2^k**.**

Randomized Algorithm:

- **A state is a convex combination of basic-states.**
- k **variabls - states are vectors in** R^{2^k}**,**

  $\psi = \sum_{j \in \{0,1\}^k} \alpha_j v_j$

  with $\alpha_j \in \mathbb{R}$ **and** $\sum_{j \in \{0,1\}^k} \alpha_j = 1$.

Quantum Circuits / Algorithms

Quantum: k**-qbits - 2^k basic states,** $v_j, \; j \in \{0,1\}^k$**.**

- **States: are 2^k dimensional vectors, that are linear combinations of basic states.** $|\psi\rangle = \sum_{j\in\{0,1\}^k} \alpha_j |j\rangle$ **with** $\alpha_j \in \mathbb{C}$ **and** $\sum_{j\in\{0,1\}^k} |\alpha_j|^2 = 1$.

Quantum Circuits / Algorithms

Quantum: k**-qbits - 2^k basic states,** $v_j,\ j \in \{0,1\}^k$**.**

- **States: are 2^k dimensional vectors, that are linear combinations of basic states.** $|\psi\rangle = \sum_{j \in \{0,1\}^k} \alpha_j |j\rangle$ **with** $\alpha_j \in \mathbb{C}$ **and** $\sum_{j \in \{0,1\}^k} |\alpha_j|^2 = 1$.

- **Gate: unitary operator U (length-preserving matrix).**

Quantum Circuits / Algorithms

Quantum: k**-qbits - 2^k basic states,** $v_j, \; j \in \{0,1\}^k$.

- **States: are 2^k dimensional vectors, that are linear combinations of basic states.** $|\psi\rangle = \sum_{j \in \{0,1\}^k} \alpha_j |j\rangle$ **with** $\alpha_j \in \mathbb{C}$ **and** $\sum_{j \in \{0,1\}^k} |\alpha_j|^2 = 1$.

- **Gate: unitary operator U (length-preserving matrix).**

- **Output: measurement $\mathcal{M}$; for final state** $\sum_{j \in \{0,1\}^k} \beta_j |j\rangle$

$$\Pr[\textbf{output } 1] = \sum_{j \in 1\{0,1\}^{k-1}} |\beta_j|^2$$

Quantum Black-Box Algorithms

Gates:

- **computational gates, e.g.,**

$$\mathrm{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- **queries to oracle [Beals, Buhrman, Cleve, Mosca and de Wolf 98] for** $x \in \{0,1\}^n$

$$O_x : |j, b\rangle \mapsto |j, b \oplus x_j\rangle \qquad (j \in \{0,1\}^{\log n}, b \in \{0,1\})$$

Quantum Property Tester

Given a fixed property $\mathcal{P} \subseteq \{0,1\}^n$.

- **Input:** n **bits/values** $x = x_1 x_2 \ldots x_n$.
- **Quantum tester circuit, that starts with the state** $|00..0\rangle$. **Uses** O_x **oracle gates.**
- **If** $x \in \mathcal{P}$, **tester accepts.**
- **If** x **is** ϵ**-far from** $\mathcal{P}$, **tester rejects w.h.p.**
- **Complexity: number of # oracle query gates** O_x

Motivation

Show gaps between quantum algorithms and Classical Ones.

Results

- **Complexity separations: give properties s.t.**

quantum	classical	
$\mathrm{O}(1)$	$\Omega(\log n)$	**(random Hadamard codewords)**
$O(\log n)$	$n^{\Omega(1)}$	**(Simon)**
$n^{\Omega(1)}$		**(pseudo-random numbers)**

$n =$ **number of values in input**

First attempt for a 1 – vs. $\log n$ gap

Inner Product Over F_2: For $x, y \in \{0,1\}^k$:

$$< x, y > := \sum_{\ell=1}^{k} y_\ell j_\ell (mod\ 2).$$

Hadamard code of $y \in \{0,1\}^{\log n}$:

$$h(y) := x_0 \dots x_{n-1} \textbf{ with } x_j =< y, j >$$

Candidate for a 'classically hard' property: Being a Hadamard codeword, namely $\mathcal{P} = \{h(y) |\ y \in \{0,1\}^{\log n}\}$.

$\exists$ **quantum black-box algorithm to find** y **with one application of** $O_{h(y)}$ **[Bernstein Vazirani].**

Classically: need $\log n$ **queries in order to find** y **form** $h(y)$ **(information theory).**

Testing Hadamard Codewords

Catch: Classical Tester (does not need to know y).

- **for** $O(1/\epsilon)$ **many pairs** j, j'**: query** x_j, $x_{j'}$, **and** $x_{j \oplus j'}$.
- **reject if** $x_j \oplus x_{j'} \neq x_{j \oplus j'}$ **for any of the pairs. otherwise: accept.**

A better candidate

For a subset $A \subseteq \{0,1\}^{\log n}$**, let** $P_A = \{h(y) |\ y \in A\}$**. Namely,** P_A **contains the Hadamard codewords of vectors in a predefined subset** A**.**

The subset of Choice - random.

A better candidate

For a subset $A \subseteq \{0,1\}^{\log n}$, let $P_A = \{h(y) |\ y \in A\}$. Namely, P_A contains the Hadamard codewords of vectors in a predefined subset A.

The subset of Choice - random.

Quantum Test

- **In one query find y such that $h(y) = x$.**
- **Check that $y \in A$**
- **Test for random $i \leq n$ that $x_i =< y, i >$.**

A better candidate

For a subset $A \subseteq \{0,1\}^{\log n}$, let $P_A = \{h(y)|\ y \in A\}$. Namely, P_A contains the Hadamard codewords of vectors in a predefined subset A.

The subset of Choice - random.

Quantum Test

- **In one query find y such that $h(y) = x$.**
- **Check that $y \in A$**
- **Test for random $i \leq n$ that $x_i =< y, i >$.**

A $\Omega(\log n)$ classical lower bound can be proven.

Exponential Separation

A property $\mathcal{P}$ such that

$\mathcal{P}$ is quantum-testable with $O(\log n)$ queries.

Any classical tester need $n^{\Omega(1)}$ queries.

Exponential Separation

Simon's Promise problem:

Input: $f : \{0,1\}^n \longrightarrow \{0,1\}^n$, **such that.**

- f **is** 2 **to** 1.
- **There is an** $s \in \{0,1\}^n - \{(0,...0)\}$, **such that for every** $x, \;\; f(x) = f(x \oplus s)$.

Goal: **Find** $s \neq (0,...,0)$.

Quantum - in $O(n)$ queries [Simon 97,Brassard Høyer 97] .

Classical - $\Omega(2^{n/2})$ **(birthday paradox).**

Brassard Høyer Algorithm

- **There are $n-1$ rounds.**
- **The ith round produces a vector $z_i \in \{0,1\}^n$ for which,**

  (a) $< z, s >= 0$

  (b) z_i **is linearly independent of** $\{z_1, ..., z_{i-1}\}$**.**

After $n-1$ times can find s

Exact !

Our Property

$\mathcal{P} = \{f : \{0,1\}^n \longrightarrow \{0,1\} \mid \ \textit{such that } \exists s \neq (0,...,0),\ \forall x,\ f(x) = f(x \oplus s)\}$.

Quantum - in $O(n \log n)$ queries.

Classical - $\Omega(2^{n/2})$.

Quantum Lower bounds for Property Testing

- **Most of the properties $\mathcal{P}$ of n bit strings, of size $2^{n/20}$ require $\Omega(n)$ quantum queries.**
- **The range of d-wise independent n-bit generator requires $d/2$ random queries.**

we use the Polynomial method.

Polynomial method

Let $f : \{0,1\}^n \longrightarrow \{0,1\}$**.**

[Beals, Buhrman, Cleave, Mosca, d' Wolf 98] If f **has a q-query quantum algorithm, then there is a multilinear polynomial** p **that approximates** f**.**

For all $x \in \{0,1\}^n$**,**

$$|p(x) - f(x)| \leq 1/3$$

This is true for promise problems too - in particular for property testing

Proving Q-lower bounds for property testing

To prove lower bounds for a property $\mathcal{P}$, need to show that every real, multilinear polynomial p for which,

- **For all** $x \in \mathcal{P}, \quad p(x) \geq 2/3$**.**
- **For all** $x, \; dist(x, \mathcal{P}) \geq \epsilon n, \quad p(x) \leq 1/3$**.**
- **For all** $x \in \{0,1\}^n, \; p(x) \in [0,1]$

Has high degree.

Open Problems

- **Gaps of 1 vs. $\Omega(n)$?**
- **Natural properties.**
- **Characterization of efficient Q-testers in terms of polynomials ?**