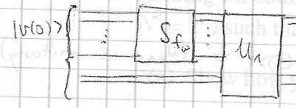
So the measurement of $v_{\omega_1}^T$ and $v_{\omega_2}^T$ is the same with high probability.
But we need to achieve different measurements for $v_{\omega_1}^T$ and $v_{\omega_2}^T$. $\square$

Second method - Polynomial Method



(General algorithm, as in 1st method)

Define $y_j = f(j-1)$, $j-1 \in \{0,1\}^n$, $y_j \in \{0,1\}$.

It is obvious that computing $OR(y_1, y_2, \dots, y_N)$ is simpler than solving the unstructured search. So we will suggest a lower bound for OR.

$S_f: |x\rangle \rightarrow (-1)^{f(x)}|x\rangle = (1-2y_x)|x\rangle$

Assume: $|v(0)\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.

Then $|v(0)\rangle \xrightarrow{S_f} \sum_x \underbrace{\alpha_x (1-2y_x)}_{\alpha_x^1(y_1,\dots,y_N)} |x\rangle$

$\alpha_x^1(y_1,\dots,y_N) \rightarrow$ polynomial in $y_1,\dots,y_N$ of degree 1.

After U_1 we get a superposition of $|x\rangle$, with coefficients $\alpha_x(\alpha_1,\dots,\alpha_N)$ - polynomial of degree 1 - U_1 acting on a superposition is a linear combination.

So after T steps, $|final\rangle = \sum_x \alpha_x^f |x\rangle$, where $\alpha_x^f(y_1,\dots,y_N)$ is a polynomial of degree T.

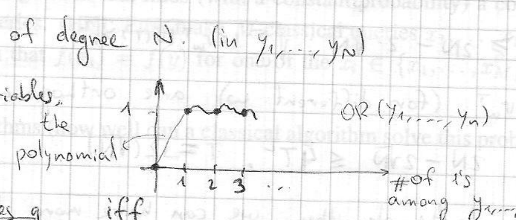
If we measure only the first bit the probability to get 1 is

$pr(\text{"1"}) = \sum_{x\in\{1\}\times\{0,1\}^{n-1}} |\alpha_x|^2$ - is a polynomial of degree $2T$.

If $OR(y_1,\dots,y_N) = 1$, we want to achieve $Pr(\text{"1"}) \geq 0.9$, otherwise $Pr(\text{"0"}) \leq 0.1$.

$OR(y_1,\dots,y_N)$ is a polynomial of degree N. (in $y_1,\dots,y_N$)

Def: Let $y_1,\dots,y_N$ be Boolean variables, and $g(y_1,\dots,y_N): \{0,1\}^N \rightarrow \{0,1\}$.



A polynomial $p(y_1,\dots,y_N)$ <u>ε-approximates g</u> iff

$|p(y_1,\dots,y_N) - g(y_1,\dots,y_N)| \leq \varepsilon \quad \forall y_1,\dots,y_N \in \{0,1\}$. Also the <u>$\varepsilon$-degree of g</u> is:

<u>$\deg_\varepsilon p$</u> $= \min_g \{\deg p : p\ \varepsilon\text{-approximates } g\}$, and <u>$\deg g$</u> $= \deg_0 g$

Def: <u>$Q_E(g)$</u> = minimum number of queries (to $y_1,\dots,y_N$) to compute g exactly.

<u>$Q_\varepsilon(g)$</u> - minimum number of queries to compute g with probability $1-\varepsilon$ $\forall y_1,\dots,y_N$.

Another analysis (for Grover's algorithm):

In it, D is called "reflection around the mean".

For $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$, the mean is $\langle a\rangle = \frac{1}{N}\sum_{x\in\{0,1\}^n} \alpha_x$

Recall the state from Grover's algorithm: $|S\rangle = \frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle$

$\langle S|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x \alpha_x = \sqrt{N}\cdot\langle a\rangle$

$D|\psi\rangle = (2|S\rangle\langle S| - \mathbb{1})|\psi\rangle = 2\cdot|S\rangle\cdot\sqrt{N}\cdot\langle a\rangle - |\psi\rangle = 2\frac{1}{\sqrt{N}}\sum_x$

$= 2\cdot\sum_x\left(\frac{1}{\sqrt{N}}\cdot\sqrt{N}\cdot\langle a\rangle - \alpha_x\right)|x\rangle = \sum_x\left(\langle a\rangle + (\langle a\rangle - \alpha_x)\right)|x\rangle$

Recall that every state during Grover's algorithm has real amplitudes.

The ampitudes start equal. Every turn S_f flips the altitude of w (which is $f(w)=1$), $\alpha_w \to -\alpha_w$. D reflects the amplitudes around the mean: $\alpha_w \to 2\langle a\rangle - \alpha_w$. For a long time $\langle a\rangle$ stays around $\frac{1}{\sqrt{N}}$ (Or so we're told). So $\alpha_w^{t+1} \to (2t+1)\frac{1}{\sqrt{N}}$. Hence we need $O(\sqrt{N})$ turns to make it's amplitude close to 1.

Polynomial Method - Continuation

Notice: $\frac{1}{2}\deg_\varepsilon(g) \leq Q_\varepsilon(g)$, for $g = OR$. Also $\deg(OR) = N$:

$OR(y_1,\dots,y_N) = 1-(1-y_1)\cdot\ldots\cdot(1-y_N)$. Notice that OR is symmetric, that is for any $\sigma\in S_n$ $OR(\sigma(y_1,\dots,y_N)) = OR(y_1,\dots,y_N)$.

Homework 3: $\frac{1}{N!}\cdot\sum_{\sigma\in S_N} p(\sigma(y_1,\dots,y_N))$ is symmetric, and if p ε-approximates OR, then so does this polynomial.

Hence, there exists a symmetric polynomial $p(y_1,\dots,y_N)$ that ε-appoximates OR.

Homework 3: A symmetric polynomial p depends only on $y = y_1+\dots+y_N$.

Now we have $P(y)$ that ε-approximates $OR(y)$, and $\deg P = \deg p(y_1,\dots,y_N)$.

Theorem (Paturi): Let g be a non-constant symmetric Boolean function on $\{0,1\}^N$ (expressed as a polynomial in y). Define $\Gamma(g) = \min\{|2k-N+1| : g(k)\neq g(k+1)\}$

Then $\deg_\varepsilon g = \Theta\left(\sqrt{N(N-\Gamma(g))}\right)$

$\Gamma(OR) = |0-N+1| = N-1 \Rightarrow Q_\varepsilon(OR) \geq \Omega\left(\sqrt{N(N-N+1)}\right) = \Omega(\sqrt{N})$

$k=0$

$\square$

Remark: In our case we need a lower bound for computing OR with the promise: either all y_j are 0, or exactly one of them is 1.

For this function: $\Gamma(OR^{promise}) \leq N-1$

$$\Rightarrow Q_\varepsilon = \Omega\left(\sqrt{N(N-\Gamma)}\right) \geq \Omega(\sqrt{N})$$

. So the lower bound doesn't change.

Example: $PARITY(y_1, \dots, y_N) = \sum_{i=1}^{N} y_i \bmod 2$. This function is symmetric, so we can use Paturi's Theorem. $\Rightarrow Q_\varepsilon(PARITY) \geq \frac{1}{2} \deg_\varepsilon(PARITY) = \Theta(N)$.

$\Gamma(g) = 1$

$(p(y) \neq p(y+1) \ \forall y)$

Example: $MAJ(y_1, \dots, y_N) = 1 \iff \sum y_i > \frac{N}{2}$, N is even.

$$\Gamma(MAJ) = 1 \quad \left(p\left(\tfrac{N}{2}\right) \neq p\left(\tfrac{N}{2}+1\right)\right) \Rightarrow Q_\varepsilon(MAJ) = \Theta(N).$$

Quantum Communication Complexity

Alice $\longleftrightarrow$ Bob

$x \in \{0,1\}^n$ $\qquad$ $y \in \{0,1\}^n$

Internal computation is not counted - we only care how many bits/qubits to send. Goal: Both Alice and Bob want to find $f(x,y) \in \{0,1\}$

Typically: $f: D \subseteq \{0,1\}^n \times \{0,1\}^n$ - partial. If $D = \{0,1\}^n \times \{0,1\}^n$ - total function.

Trivial upper bound: n. (send x to Bob, let him compute internally).

Def: $D(f)$ - deterministic c.c (communication complexity).

$$D(f) = \min_{\text{Protocoles}} \left(\max_{x,y \in \{0,1\}^n} |\text{communication}| \right)$$

(in bits)

Example: $EQ(x,y) = 1 \iff x = y$. Here $D(EQ) = n$. (Or $n+1$, if we want that both A and B know the answer).

Def: $R(f)$ - randomized c.c.

$$R(f) = \min_{\text{protocoles}} \left(\max_{x,y \in \{0,1\}^n} |\text{communication}|, \ \text{s.t.} \Pr_{\text{randomness}}[\text{output} = f(x,y)] \geq \frac{2}{3} \right)$$

Private randomness - (r_A, r_B). - each of them generate random bits, no share.

Public randomness - r - predetermined common random bits

We write then $R^{pub}(f)$, $R^{priv}(f)$ for the 2 types of randomness.

Public randomness can simulate private randomness. =>

$$R^{pub}(f) \leq R^{priv}(f) \leq D(f).$$

<u>Def</u>: $\underline{Q(f)} = \min_{\substack{\text{quantum}\\\text{protocols}}} \left(\max_{x,y} |\text{communication}| \text{ s.t } \forall_{x,y} \Pr[\text{out} = f] \geq \frac{2}{3} \right)$

$Q(f) \leq R^{priv}(f)$. (private randomness can be simulated by Hadamard + measure)

<u>Example</u>: $R^{pub}(EQ) = ?$

Alice has x, Bob has y. Both share random, long enough r.

- Alice sends $x \cdot r \pmod 2$ (1 bit) to Bob, who compares it to $y \cdot r \pmod 2$.
- If $x = y$ we have equality. If $x \neq y$, w.p $\frac{1}{2}$ we have equality and w.p $\frac{1}{2}$ we have inequality.

So repeat it t times and get probability of error $\left(\frac{1}{2}\right)^t$.

$R^{pub}(EQ) = O(1)$.

~~Example~~

<u>Theorem</u> (Newman): $R^{pub}(f) = R^{priv}(f) \pm O(\log n)$

<u>Example</u>: $R^{priv}(EQ) = O(\log n)$.

Protocol using Error-correcting codes: $E: \{0,1\}^n \to \{0,1\}^m$, $m > n$.

Fix ε. There exists $E: \{0,1\}^n \to \{0,1\}^m$: $\forall x \neq y$ $\text{dist}(E(x), E(y)) \geq \left(\frac{1}{2} - \varepsilon\right)n$, $m = O(n)$.

Hamming Distance

- Alice, Bob compute $E(x), E(y)$.
- Alice chooses random place j in $E(x)$, sends $(j, E(x)_j)$ - $O(\log n)$ bits.

If $x = y$: $E(x)_j = E(y)_j$. If $x \neq y$ $E(x)_j \neq E(y)_j$ w.p $\frac{1}{2} - \varepsilon$.

Repeat it constant t times.

<u>Simultaneous message passing model</u> (SMP)

$$\underset{x}{\text{Alice}} \xrightarrow{m_x} \text{Referee} \xleftarrow{m_y} \underset{y}{\text{Bob}}$$
$$\downarrow$$
$$f(x,y)$$

<u>Def</u>: $\underline{D^{||}(f)} = \min_{\text{protocols} \leftarrow \mathbb{P}} \max_{x,y} (|m_x| + |m_y|)$. Similarly define $\underline{R^{||pub}(f)}$, $\underline{R^{||priv}(f)}$