Quantum majority and other Boolean functions with quantum inputs

Harry Buhrman Noah Linden Laura Mančinska Ashley Montanaro Maris Ozols

1 Introduction

Error reduction is a fundamental, natural, and useful primitive in algorithm design whose goal is to reduce the error that an algorithm makes, while not deteriorating its performance by much. An (efficient) randomised or quantum algorithm $\mathcal{A}$ will output the correct answer with reasonable success probability: $1 - \epsilon$. Often, the error ϵ is some small constant, but it may be a function depending in the size of the input or some other parameter.

The goal is to design a new algorithm $\mathcal{A}'$, that computes the same function as $\mathcal{A}$, has a comparable running time, but has a much smaller error. The standard way of doing this is by running algorithm $\mathcal{A}$ n times, yielding n answers $a_1, \ldots, a_n$. The output of the new algorithm $\mathcal{A}'$ is the *majority* value among the n outputs: $\mathrm{MAJ}(a_1, \ldots, a_n)$. The running time of this new algorithm is n times the original running time plus the time it takes to compute the majority value. Standard tools from probability theory, like *e.g.* the Chernoff bound, can be used to show that the new ϵ is exponentially smaller (in n) than the old error. This strategy of computing the majority value usually works, but in some situations may be quite complicated to prove. See for example the extensive literature on parallel repetition in interactive proof systems. This procedure also works when the error is due to imperfections in the hardware. As long as the final majority computation is error free, hardware errors can also be reduced this way.

This approach works equally well for quantum algorithms whose output is *classical*. For example BQP, the complexity class of computational problems that can be solved by a polynomial-time quantum algorithm with constant error is independent of the precise value of ϵ. As long as the error is not too big, the above majority procedure can be used to reduce the error to some arbitrary small value. However, when the output of the algorithm is *quantum,* it is not known or even clear how to implement the above described majority voting procedure. Is it possible to devise an efficient error reduction procedure that runs the quantum algorithm n times and produces the desired output state with higher probability or fidelity than the original algorithm? We make a start on how to approach this problem and open up a new area that studies this question not only for the majority function but for arbitrary functions. We call this the computation of Boolean functions with *quantum inputs*.

2 Quantum Majority Vote

Consider the following simplified version of the problem where the algorithm can produce only two orthogonal states $|\psi_0\rangle$ or $|\psi_1\rangle$. Note that we don't know which states they are, only that they are orthogonal. Moreover, assume that the algorithm is more likely to produce one of the two states. After running such algorithm n times we have the n-qubit product state

$$|\Psi\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_n\rangle \tag{1}$$

where each qubit $|\phi_i\rangle$ is in one of two orthogonal states $|\psi_0\rangle$ or $|\psi_1\rangle$. Ideally we need to return either $|\psi_0\rangle$ or $|\psi_1\rangle$, depending on which state occurs more often in the product. The difficulty

lies in the fact that we do not know the identity of $|\psi_0\rangle$ or $|\psi_1\rangle$. We refer to this problem as *quantum majority vote*. One can show that generally this task cannot be achieved perfectly due to linearity of quantum mechanics.

The simplest strategy is to select a random qubit of $|\Psi\rangle$ and output its state $|\phi_i\rangle$. While this procedure is more likely to output the most frequent state, it does not reduce the worst-case error or fidelity as was intended. Using tools from representation theory, such as Schur–Weyl duality and the Schur transform, we devise an optimal algorithm for this task that improves the naive guessing strategy. In fact, promised that at least $2/3$ of the qubit states are the same, the output fidelity of our algorithm gets arbitrarily close to 1.

Theorem 1. *For any odd n, if $|x|$ (the number of 1's in x) is arbitrary, then the optimal worst-case fidelity $F_{\mathrm{MAJ}}(n) = 1/2 + \Theta(1/\sqrt{n})$. If we promise that $\left||x| - \frac{n}{2}\right| \geq \frac{n}{3}$ then $F_{\mathrm{MAJ}}(n) = 1 - \Theta(1/n)$.*

This is quite surprising as it shows that in the quantum case success amplification with linearly small error is possible even without knowing the input/output basis. Moreover, our algorithm can be implemented efficiently. On the other hand, the rate of reduction is not as good as in the classical setting where we can reduce the error exponentially.

We remark that an alternative notion of majority voting is given by the closely related concept of purification [CEM99, KW01]. However, our viewpoint can be generalized to a more wide-ranging framework of Boolean functions that have quantum inputs, as we discuss next.

3 Boolean Functions with Quantum Inputs

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ gets as input an n-bit string $x = x_1 \dots x_n$ and outputs a bit $f(x)$. We replace the classical input x by a quantum state as in eq. (1): $|\Psi\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_n\rangle$ where $|\phi_i\rangle = |\psi_{x_i}\rangle$. That is, each input bit x_i is replaced by one of two orthogonal qubit states $|\psi_0\rangle$ or $|\psi_1\rangle$, depending on x_i. The goal is to quantumly compute f on this state and output a state ρ which is equal, or as close as possible, to $|\psi_{f(x)}\rangle$. Since we do not know what $|\psi_0\rangle$ and $|\psi_1\rangle$ are, only that they are orthogonal, there is not a unique way to assign them to the bits 0 and 1. Therefore, it only makes sense to explore this task for Boolean functions that are self-dual or *covariant*: $f(\neg x_1, \dots, \neg x_n) = \neg f(x_1, \dots, x_n)$. For simplicity, we focus on *symmetric* functions in which case $f(x)$ only depends on $|x|$. Note that for odd n, the majority function $\mathrm{MAJ}(x_1, \dots, x_n)$ is a natural example of a function that is both covariant and symmetric.

One-bit Boolean functions with quantum inputs and outputs have already been studied and even experimentally implemented. In [BHW99] Bužek, Hillery and Werner ask for the best quantum approximation of what they call the universal-NOT gate. This gate is defined as the anti-unitary map sending an arbitrary qubit state $|\psi\rangle = \binom{\alpha}{\beta}$ to its perpendicular state $|\psi^\perp\rangle = \binom{\overline{\beta}}{-\overline{\alpha}}$. The best quantum algorithm for this problem achieves the worst-case output fidelity of $2/3$ [BHW99].

We consider the problem of covariant symmetric functions in the n-qubit setting. Our main result is a generic algorithmic template which we can tune with a set of parameters $\mathbf{t}$. We provide an efficient linear program that computes the optimal setting of $\mathbf{t}$ and a gate-efficient implementation of the corresponding quantum algorithm $\mathcal{A}_{\mathbf{t}}$.

Theorem 2 (informal)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a symmetric and covariant Boolean function. For some choice of interpolation parameters $\mathbf{t}$, our template algorithm $\mathcal{A}_{\mathbf{t}}$ is optimal for computing f. Moreover, the optimal worst-case fidelity for computing f can be computed by a simple linear program of size $\lfloor n/2 \rfloor + 1$ whose optimal solution also yields the optimal choice of parameters $\mathbf{t}$. The resulting algorithm $\mathcal{A}_{\mathbf{t}}$ can be implemented using $O(n^4 \log n)$ elementary quantum gates.*

Template Algorithm $\mathcal{A}_{\mathbf{t}}$

Input: Quantum state $U^{\otimes n}|x\rangle$ with an unknown $x \in \{0,1\}^n$ and $U \in \mathrm{U}(2)$.

Output: An approximation of $U|f(x)\rangle$.

Parameters: A vector of *interpolation parameters* $\mathbf{t} = (t_\lambda : \lambda \vdash n)$ where each $t_\lambda \in [0,1]$.

Step 1: *Weak Schur sampling*: apply Schur transform U_{Sch} and measure $\lambda \vdash n$.

Step 2: Discard the permutation register.

Step 3: Apply Φ_{Tr} with probability t_λ and Φ_{UNOT} with probability $(1 - t_\lambda)$, where λ is the measurement outcome from Step 1.

Φ_{UNOT} essentially corresponds to the quantum universal-NOT discussed above while Φ_{Tr} corresponds to tracing out all but one of the qubits. Using this structure of the optimal algorithm, one can design optimal interpolation parameters $\mathbf{t}$ for any covariant symmetric n-bit function. Further analysis of the linear program can then establish optimal quantum algorithms for infinite families of Boolean function, one for every n. We have performed this analysis for the majority function MAJ discussed above in the context of error reduction for quantum algorithms that have a quantum output. For majority, it is optimal to apply Φ_{Tr} with probability one in Step 3 so the resulting algorithm is especially simple. Another natural function is the PARITY function, which is covariant if n is odd, where we have numerical evidence that strongly suggests that the optimal fidelity to compute PARITY for arbitrary n is $1/2 + \Theta(1/n)$. This may have further applications to hardness amplification in the quantum setting.

Example. *We now illustrate the optimal algorithm $\mathcal{A} := \mathcal{A}_{(1,1)}$ for 3-bit majority. As input we recieve $U^{\otimes 3}|x\rangle$ for some unknown unitary $U \in \mathrm{U}(2)$ and a 3-bit string x. In Step 1 we obtain partition label $\lambda = (\lambda_1, \lambda_2) \vdash 3$, with probability $p_\lambda(x)$. Upon measuring λ and after discarding in Step 2 we are left with a state of dimension $\lambda_1 - \lambda_2 + 1$, which can be seen as corresponding to*

$$U^{\otimes(\lambda_1-\lambda_2)}|s(|x| - \lambda_2)\rangle$$

from the $(\lambda_1 - \lambda_2)$-qubit symmetric space, where $|s(w)\rangle$ is the symmetric state of Hamming weight w. There are only two ways to partition three elements into two parts: $\lambda = (2,1)$ and $\lambda = (3,0)$. For inputs $x = 000, 001$ and all measurement outcomes λ we list the associated probability $p_\lambda(x)$ and the resulting state after Step 2 in the table below.

	$x = 000$	$x = 001$
$\lambda = (2,1)$	$p_\lambda(x) = 0$	$p_\lambda(x) = \frac{2}{3}$
	—	$U\|0\rangle$
$\lambda = (3,0)$	$p_\lambda(x) = 1$	$p_\lambda(x) = \frac{1}{3}$
	$U^{\otimes 3}\|000\rangle$	$U^{\otimes 3}\frac{\|001\rangle+\|010\rangle+\|100\rangle}{\sqrt{3}}$

In case of majority for both $x = 000$ and $x = 001$, the correct output is $U|0\rangle$. Examining the states resulting after Step 2 from the above table, it seems reasonable to simply return one of the qubits as this allows to achieve fidelity 1 in all cases except for $(\lambda = (3,0), x = 001)$ where we get fidelity $\frac{2}{3}$. This is exactly what $\mathcal{A}$ does in Step 3, since for majority $t_\lambda = 1$ for all λ. Overall, the algorithm $\mathcal{A}$ achieves worst-case fidelity

$$\min\left\{0 + 1 \cdot 1,\ \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot \frac{2}{3}\right\} = \frac{8}{9}. \tag{2}$$

References

[BHW99] Vladimír Bužek, Mark Hillery, and Reinhard F. Werner. Optimal manipulations with qubits: Universal-NOT gate. *Phys. Rev. A*, 60(4):R2626–R2629, Oct 1999. arXiv:quant-ph/9901053, doi:10.1103/PhysRevA.60.R2626.

[CEM99] J. Cirac, A. Ekert, and C. Macchiavello. Optimal purification of single qubits. *Phys. Rev. Lett.*, 82:4344–4347, 1999. arXiv:quant-ph/9812075.

[KW01] M. Keyl and R. Werner. The rate of optimal purification procedures. *Annales Henri Poincaré*, 2:1–26, 2001. arXiv:quant-ph/9910124.