Quantum Algorithms for Classical Probability Distributions

Aleksandrs Belovs
Faculty of Computing, University of Latvia, Riga, Latvia
aleksandrs.belovs@lu.lv

Abstract

We study quantum algorithms working on classical probability distributions. We formulate four different models for accessing a classical probability distribution on a quantum computer, which are derived from previous work on the topic, and study their mutual relationships.

Additionally, we prove that quantum query complexity of distinguishing two probability distributions is given by their inverse Hellinger distance, which gives a quadratic improvement over classical query complexity for any pair of distributions.

The results are obtained by using the adversary method for state-generating input oracles and for distinguishing probability distributions on input strings.

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases quantum query complexity, quantum adversary method, distinguishing probability distributions, Hellinger distance

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.16

Related Version A full version of the paper is available at `https://arxiv.org/abs/1904.02192`.

Funding This research is partly supported by the ERDF grant number 1.1.1.2/VIAA/1/16/113.

Acknowledgements Most of all I would like to thank Anrás Gilyén for the suggestion to work on this problem. I am also grateful to Frédéric Magniez, Shalev Ben-David, and Anurag Anshu for useful discussions, as well as to anonymous reviewers for their comments. Part of this research was performed while at the Institute for Quantum Computing in Waterloo, Canada. I would like to thank Ashwin Nayak for hospitality.

1 Introduction

It is customary for a quantum algorithm to receive its input and produce its output in the form of a classical string of symbols, quantized in the form of an oracle. This is purely classical way to store information, and, given intrinsic quantum nature of quantum algorithms, this might be not the best interface for many tasks. Moreover, even *classical* algorithms make use of other interfaces as well. For instance, classical algorithms can receive and produce *samples* from some probability distribution. In this paper we study quantum algorithms working with classical probability distributions.

1.1 Models

We analyse previously used models of accessing classical probability distributions by quantum algorithms. We prove and conjecture some relations between them. We give more detail in Section 3, but for now let us very briefly introduce the models.

In one of the models, used in, e.g., [16, 18, 27, 25], the probability distribution is encoded as a frequency of a symbol in a given input string, which the quantum algorithm accesses via the standard input oracle. In another model, e.g., [17, 2, 6], the input probability distribution

is given through a quantum oracle that prepares a state in the form $\sum_a \sqrt{p_a}|a\rangle$. Finally, one more model, used in [26, 20, 19], is similar but with additional state tensored with each $|a\rangle$.

This is the latter model that we champion in this paper. We find this model particularly relevant because of our belief that an input oracle should be easily interchangeable with a quantum subroutine, see discussion in [26]. It is relatively easy to see what it means for a quantum algorithm to output a probability distribution: just measure one of the registers of its final state. The latter model precisely encompasses all such subroutines. We conjecture that this model is equivalent to the first model, see also [19], where a similar conjecture is made.

1.2 Distinguishing two Probability Distributions

Additionally, we study the problem of distinguishing two probability distributions. This might be the most fundamental problem one can formulate in these settings. Given two fixed probability distributions p and q, and given an input oracle encoding one of them, the task is to detect which one, p or q, the oracle encodes. To the best of our knowledge, this particular problem has not been studied in quantum settings, although similar problems of testing the distance between two distributions [16] and testing whether the input distribution is equal to some fixed distribution [18] have been already studied.

Classically one needs $\Theta\big(1/\operatorname{d_H}(p,q)^2\big)$ samples to solve this problem for any p and q, where $\operatorname{d_H}$ stands for Hellinger distance. This result is considered "folklore", see, e.g. [7, Chapter 4]. We prove that for any p and q and for any of the models of access described above, query complexity of this problem is $\Theta(1/\operatorname{d_H}(p,q))$. This constitutes quadratic improvement over classical algorithm for *any* pair of distributions p and q. Moreover, our algorithm also admits a simple low-level implementation, which is efficient assuming the distributions p and q can be efficiently processed.

1.3 Techniques

Our main technical tool for proving the upper bound is the version of the adversary bound for state-generating oracles, which is a special case of the adversary bound for general input oracles [11]. It is stated in the form of a relative γ_2-norm and generalises the dual formulation of the general adversary bound [28, 29] for function evaluation, as well as for other problems [5, 24]. The dual adversary bound has been used rather successfully in construction of quantum algorithms, as in terms of span programs and learning graphs [9, 23, 13, 8, 22], as in an unrelated fashion [10, 4]. Our work gives yet another application of these techniques for construction of quantum algorithms.

Our upper bound naturally follows from the analysis of the γ_2-norm optimisation problem associated with the task. We also compare our techniques with more standard ones involving quantum rejection sampling and amplitude amplification in the spirit of [20] and show that our techniques give a slightly better result.

As for the lower bound, we make use of the version of the adversary bound from [12]. This is a simple generalisation of the primal version of the general adversary bound [21] for function evaluation, and it is tailored for the task we are interested in: distinguishing two probability distributions on input strings. Our lower bound is surprisingly simple and gives a very intuitive justification of the significance of Hellinger distance for this problem.

2 Preliminaries

We mostly use standard linear-algebraic notation. We use ket-notation for vectors representing quantum states, but generally avoid it. We use A^* to denote conjugate operators (transposed and complex-conjugated matrices). For P a predicate, we use 1_P to denote 1 if P is true, and 0 if P is false. We use $[n]$ to denote the set $\{1, 2, \dots, n\}$.

It is unfortunate that the same piece of notation, $\oplus$, is used both for direct sum of matrices and direct sum of vectors, which is in conflict with each other if a vector, as it often does, gets interpreted as a column-matrix. Since we will extensively use both these operations in this paper, let us agree that $\boxplus$ denotes direct sum of vectors, and $\oplus$ always denotes direct sum of matrices. Thus, in particular, for $u, v \in \mathbb{R}^m$, we have

$$u \boxplus v = \begin{pmatrix} u_1 \\ \vdots \\ u_m \\ v_1 \\ \vdots \\ v_m \end{pmatrix} \quad \text{and} \quad u \oplus v = \begin{pmatrix} u_1 & 0 \\ \vdots & \vdots \\ u_m & 0 \\ 0 & v_1 \\ \vdots & \vdots \\ 0 & v_m \end{pmatrix}.$$

We often treat scalars as 1×1-matrices which may be also thought as vectors.

2.1 Relative γ_2-norm

In this section, we state the relative γ_2-norm and formulate some of its basic properties. All the results are from [11].

▶ **Definition 1** (Relative γ_2-norm). *Let $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be vector spaces, and D_1 and D_2 be some sets of labels. Let $A = \{A_{xy}\}$ and $\Delta = \{\Delta_{xy}\}$, where $x \in D_1$ and $y \in D_2$, be two families of linear operators: $A_{xy} \colon \mathcal{Z}_2 \to \mathcal{Z}_1$ and $\Delta_{xy} \colon \mathcal{X}_2 \to \mathcal{X}_1$. The* relative γ_2-norm,

$$\gamma_2(A|\Delta) = \gamma_2(A_{xy} \mid \Delta_{xy})_{x \in D_1,\, y \in D_2},$$

is defined as the optimal value of the following optimisation problem, where Υ_x and Φ_y are linear operators,

$$\text{minimise} \quad \max\Big\{\max_{x \in D_1} \|\Upsilon_x\|^2, \max_{y \in D_2} \|\Phi_y\|^2\Big\} \tag{1a}$$

$$\text{subject to} \quad A_{xy} = \Upsilon_x^*(\Delta_{xy} \otimes I_{\mathcal{W}})\Phi_y \quad \text{for all } x \in D_1 \text{ and } y \in D_2; \tag{1b}$$

$$\mathcal{W} \text{ is a vector space}, \quad \Upsilon_x \colon \mathcal{Z}_1 \to \mathcal{X}_1 \otimes \mathcal{W}, \quad \Phi_y \colon \mathcal{Z}_2 \to \mathcal{X}_2 \otimes \mathcal{W}. \tag{1c}$$

This is a generalisation of the usual γ_2-norm, also known as Schur (Hadamard) product operator norm [14].

In a quantum algorithm with general input oracles, the input oracle performs some unitary operation O on some fixed Hilbert space. The algorithm can execute either O or its inverse O^{-1} on some register. Each execution counts as one query. It is known that O is equal to one O_x out of a set of possible input unitaries, where x ranges over some set D of labels. If $O = O_x$, the algorithm has to perform a unitary V_x on some specified part of its work-space. The algorithm knows in advance all possible O_x and which V_x corresponds to each O_x, but it does not know which O_x it is given in a specific execution. The adversary bound corresponding to this problem is $\gamma_2\big(V_x - V_y \mid O_x - O_y\big)_{x,y \in D}$. This bound is *semi-tight*: it is a lower bound on the exact version of the problem and an upper bound on the approximate version.

The γ_2-norm formalism is modular in the sense that the general task of implementing a unitary can be replaced by something more specific. For instance, assume that our task is to evaluate a function $f(x)$. Then the adversary bound reads as $\gamma_2\big(1_{f(x)\neq f(y)} \mid O_x - O_y\big)_{x,y\in D}$. In this case, the bound is tight: it is also a lower bound on the approximate version of the problem.

As another example, consider the standard input oracle O_x encoding a string $x \in [q]^n$. It works as $O_x\colon |i\rangle|0\rangle \mapsto |i\rangle|x_i\rangle$, which can be seen as a direct sum of oracles performing transformation $|0\rangle \mapsto |x_i\rangle$. Using the modular approach, the corresponding adversary bound becomes $\gamma_2\big(1_{f(x)\neq f(y)} \mid \bigoplus_j 1_{x_j\neq y_j}\big)_{x,y\in D}$, where $\bigoplus$ stands for direct sum of matrices (resulting in a diagonal matrix). This is equivalent to the usual version of dual adversary for function evaluation (up to a constant factor).

Now we consider state-generating input oracles[1]. In this case, the input to the algorithm is given by a state $\psi \in \mathbb{C}^m$, and the algorithm should work equally well for any unitary performing the transformation $O\colon |0\rangle \mapsto |\psi\rangle$. Without loss of generality, we may assume that $e_0 = |0\rangle$ is orthogonal to $\mathbb{C}^m$, thus the operator O above works in $\mathbb{C}^{m+1}$.

The corresponding γ_2-object can be defined in two alternative ways:

$$L_\psi = \psi e_0^* + e_0\psi^* \qquad \text{or} \qquad L_\psi = \psi \oplus \psi^*.$$

In the second expression, ψ is an $m \times 1$-matrix and ψ^* is a $1 \times m$-matrix, the resulting matrix being of size $(m+1)\times(m+1)$. In the case of a function-evaluation problem, the corresponding adversary bound is $\gamma_2\big(1_{f(x)\neq f(y)} \mid L_{\psi_x} - L_{\psi_y}\big)_{x,y\in D}$.

Let us also state the version of the adversary bound for the decision problem with state-generating input oracles. This is the version we will use further in the paper. Assume we have a collection of states $\psi_x \in \mathcal{X}$ for $x \in D_0$, and a collection of states $\psi_y \in \mathcal{X}$ for $y \in D_1$. The task is to distinguish the two classes of states. Let $D = D_0 \cup D_1$. Using the general case, we obtain the following version of the adversary bound.

▶ **Theorem 2.** *The quantum query complexity of the decision problem with state-generating oracles as above is equal to* $\gamma_2\big(1 \mid L_{\psi_x} - L_{\psi_y}\big)_{x\in D_0, y\in D_1}$ *up to a constant factor.*

An explicit optimisation problem for $\gamma_2\big(1 \mid L_{\psi_x} - L_{\psi_y}\big)_{x\in D_0, y\in D_1}$ is given by

$$\begin{array}{lll} \text{minimise} & \max_{z\in D}\big(\|u_z\|^2 + \|v_z\|^2\big) & \\ \text{subject to} & \big\langle v_x, (\psi_x - \psi_y)\otimes u_y\big\rangle + \big\langle (\psi_x - \psi_y)\otimes u_x, v_y\big\rangle = 1 & \forall x \in D_0, y \in D_1; \\ & u_z \in \mathcal{W}, \quad v_z \in \mathcal{X}\otimes\mathcal{W} & \forall z \in D. \end{array} \tag{2}$$

This result follows from general results of [11], see the full version of the paper, where we also give a stand-alone implementation and analysis of the corresponding quantum algorithm.

3 Models

In this section we formally define four different models how a quantum algorithm can access a classical probability distribution $p = (p_a)_{a\in A}$. These models were briefly explained in the introduction. We would like to understand relations between them, and, ideally, prove some equivalences between them.

[1] The results below will appear in an updated version of [11] (to appear). Alternatively, refer to the full version of the paper.

(i) A standard input oracle encoding a string $x \in A^n$ for some relatively large n, where p_a is given as the frequency of a in x:

$$p_a = \frac{1}{n}\Big|\{i \mid x_i = a\}\Big|.$$

(ii) A standard input oracle encoding a string $x \in A^n$ for some relatively large n, where each x_i is drawn independently at random from p.

(iii) A quantum procedure that generates the state

$$\mu_p = \sum_a \sqrt{p_a}|a\rangle = \boxplus_a \sqrt{p_a}. \tag{3}$$

(iv) A quantum procedure that generates a state of the form

$$\sum_a \sqrt{p_a}|a\rangle|\psi_a\rangle = \boxplus_a \sqrt{p_a}\psi_a, \tag{4}$$

where ψ_a are arbitrary unit vectors.

As mentioned in the introduction, model (i) is used in [16, 18, 27, 25]. It has a downside that the probabilities p_a must be multiples of $1/n$. All other models are free from this assumption.

Model (ii) seems like the most obvious way to encode probability distribution as a classical string, which a quantum algorithm can later gain access to. Up to our knowledge, this model has not been previously used. It has a downside that the distribution p is encoded as a probability distribution over possible input strings, which is not usual for quantum algorithms. The acceptance probability of the quantum algorithm depends both on the randomness introduced by the algorithm and the randomness in the input.

Model (iii) is the one used in [17, 2, 6]. And model (iv) is used in [26, 20, 19]. Both of these two models assume that the input oracle prepares a quantum state, which again is not very common for quantum algorithms.

▶ **Proposition 3.** *We have the following relations between these models.*

(a) *Models (i) and (ii) are equivalent assuming n is large enough. More precisely, no quantum algorithm can distinguish models (i) and (ii) encoding the same probability distribution unless it makes $\Omega(n^{1/3})$ queries.*

(b) *Model (iv) is more general than model (i). This means that any algorithm working in model (iv) can be turned into an algorithm working in model (i) with the same query complexity.*

(c) *Model (iv) is strictly more general than model (iii). This means there exist problems where model (iii) allows substantially smaller query complexity than model (iv).*

Proof. We leave (a) for the end of the proof, and let us start with (b). Note that using one query to the input oracle of model (i), it is possible to prepare that state

$$\frac{1}{\sqrt{n}}\sum_i |i\rangle|x_i\rangle = \sum_{a\in A}\left[\frac{1}{\sqrt{n}}\sum_{i:x_i=a}|i\rangle\right]\otimes|a\rangle,$$

which is a legitimate input state in model (iv) if one swaps the registers.

Now let us prove (c). It is obvious that model (iv) is more general than model (iii). To prove that (iii) cannot simulate (iv), consider the collision problem [15]. In this problem, a function $f\colon [n] \to [n]$ is given, and one has to distinguish whether f is 1-to-1 or 2-to-1. In terms of model (i), this boils down to distinguishing a probability distribution p which is uniform on $[n]$ from a probability distribution q which is uniform on half of $[n]$.

In model (iii), this problem can be solved in $O(1)$ queries because the state μ_p as in (3) has inner product $1/\sqrt{2}$ with all μ_q. On the other hand, by [1, 3], quantum query complexity of this problem in model (i) is $\Omega(n^{1/3})$. As model (iv) is more general than model (i), this gives the required lower bound.

To prove (a), we show that if one can distinguish models (i) and (ii), one can distinguish a random function from a random permutation, and the result follows from the lower bound of $\Omega(n^{1/3})$ for this task from [30]. Indeed, let p be a probability distribution and let y be a fixed string encoding p as in model (i). Let $\sigma\colon [n] \to [n]$ be a function, and consider the input string x given by $x_i = y_{\sigma(i)}$, which can be simulated given oracle access to σ (as the string y is fixed). If σ is a random permutation, then x is a uniformly random input string from model (i). If σ is a random function, then x is distributed as in model (ii). ◀

4 Distinguishing Two Probability Distributions

Recall the definition of Hellinger distance between two probability distributions p and q on the same space A:

$$\mathrm{d_H}(p,q) = \sqrt{\frac{1}{2}\sum_{a\in A}\left(\sqrt{p_a}-\sqrt{q_a}\right)^2}.$$

Up to a constant factor, it equals $\|\mu_p - \mu_q\|$ and $1 - \langle \mu_p, \mu_q\rangle$, where μ_p and μ_q are as in (3).

In this section, we prove the following result:

▶ **Theorem 4.** *For any two probability distributions p and q on the same space A, and any model of accessing them from Section 3, the quantum query complexity of distinguishing p and q is*

$$\Theta\left(\frac{1}{\mathrm{d_H}(p,q)}\right).$$

Note that this is quadratically better than complexity of the best classical algorithm *for every choice of p and q*. Note also that for this problem model (iii) is equal in strength to the remaining models.

The proof of main involves proving lower and upper bounds in all four models, but, luckily, we can use relations from Proposition 3. The outline of the proof is as follows. We prove upper bound in model (iv), which implies upper bounds in all other models as model (iv) is the most general of them. As for the lower bounds, we prove it for model (ii), which implies lower bounds in models (i) and (iv). For model (iii), we prove the lower bound independently. As a bonus, we prove an upper bound in model (iii) as a warm-up for the upper bound in model (iv).

In most of the proofs, we will use α for the angle between the vectors μ_p and μ_q. Note that

$$\alpha = \Theta(\|\mu_p - \mu_q\|) = \Theta(\mathrm{d_H}(p,q)).$$

4.1 Analysis in Model (iii)

In this section, we analyse the problem in model (iii).

▷ Claim 5. Quantum query complexity of distinguishing probability distributions p and q in model (iii) is $\Theta\big(1/\operatorname{d_H}(p,q)\big)$.

Proof. Let us start with the upper bound. Let O be the input oracle, and let U be a unitary that maps $|\mu_p\rangle$ into $|0\rangle$ and $|\mu_q\rangle$ into $\cos\alpha|0\rangle+\sin\alpha|1\rangle$. Now use quantum amplitude amplification on the unitary UO amplifying for the value $|1\rangle$. The algorithm can be also made exact using exact quantum amplitude amplification.

Now let us prove the lower bound. Let O_p be the input oracle exchanging $|0\rangle$ and $|\mu_p\rangle$ and leaving the vectors orthogonal to them intact. Similarly, let O_q exchange $|0\rangle$ and $|\mu_q\rangle$. Simple linear algebra shows $\|O_p-O_q\|=O(\alpha)$. Let $\mathcal{A}^O$ be a query algorithm making t queries to O and distinguishing O_p from O_q. Then,

$$\big\|\mathcal{A}^{O_p}-\mathcal{A}^{O_q}\big\|\le t\|O_p-O_q\|=O(t\alpha).$$

As this must be $\Omega(1)$, we get that $t=\Omega(1/\alpha)$. ◁

4.2 Upper Bound in Model (iv)

The aim of this section is to prove the following claim.

▷ Claim 6. Quantum query complexity of distinguishing probability distributions p and q in model (iv) is $O\big(1/\operatorname{d_H}(p,q)\big)$.

We prove this claim by constructing a feasible solution to (2). In the full version of the paper, we explain how to implement this algorithm time-efficiently and give a comparison to an algorithm using more typical techniques.

Let ψ and ϕ be some vectors encoding p and q, respectively, as in model (iv). That is,

$$\psi=\bigboxplus_a\sqrt{p_a}\psi_a,\qquad\text{and}\qquad\phi=\bigboxplus_a\sqrt{q_a}\phi_a,$$

where ψ_a and ϕ_a are some normalised vectors. Our goal is to come up with a feasible solution to (2) with ψ_x and ψ_y replaced by ψ and ϕ.

We first analyse a pair of vectors $\sqrt{p_a}\psi_a$ and $\sqrt{q_a}\phi_a$ for a fixed a. We would like to get a construction in the spirit of (2) that "erases" directions ψ_a and ϕ_a, and only depends on the norms $\sqrt{p_a}$ and $\sqrt{q_a}$. One way is to use the following identity:

$$\Big\langle\sqrt{p_a}\psi_a,\sqrt{p_a}\psi_a-\sqrt{q_a}\phi_a\Big\rangle+\Big\langle\sqrt{p_a}\psi_a-\sqrt{q_a}\phi_a,\sqrt{q_a}\phi_a\Big\rangle=p_a-q_a. \tag{5}$$

We combine this identity over all a, add weights c_a, and re-normalise:

$$\Big\langle\frac{\bigboxplus_a c_a\sqrt{p_a}\psi_a}{\sqrt[4]{\sum_a c_a^2p_a}},\ (\psi-\phi)\cdot\sqrt[4]{\sum c_a^2p_a}\Big\rangle + \Big\langle(\psi-\phi)\cdot\sqrt[4]{\sum c_a^2q_a},\ \frac{\bigboxplus_a c_a\sqrt{q_a}\phi_a}{\sqrt[4]{\sum_a c_a^2q_a}}\Big\rangle=\sum_a c_a(p_a-q_a),$$

which gives

$$\gamma_2\Big(\sum_a c_a(p_a-q_a)\ \Big|\ L_\psi-L_\phi\Big)_{\psi,\phi}\le\sqrt{\sum_a c_a^2p_a}+\sqrt{\sum_a c_a^2q_a}.$$

Dividing by $\sum_a c_a(p_a - q_a)$, we get that complexity of distinguishing p from q is at most

$$O\left(\frac{\sqrt{\sum_a c_a^2 p_a} + \sqrt{\sum_a c_a^2 q_a}}{\sum_a c_a(p_a - q_a)}\right). \tag{6}$$

Using triangle inequality

$$\sqrt{\sum_a c_a^2(\sqrt{p_a} + \sqrt{q_a})^2} \le \sqrt{\sum_a c_a^2 p_a} + \sqrt{\sum_a c_a^2 q_a} \le 2\sqrt{\sum_a c_a^2(\sqrt{p_a} + \sqrt{q_a})^2},$$

so (6) is equivalent to

$$O\left(\frac{\sqrt{\sum_a c_a^2(\sqrt{p_a} + \sqrt{q_a})^2}}{\sum_a c_a(p_a - q_a)}\right).$$

Now it is easy to see that it is minimised to

$$O\left(\frac{1}{\sqrt{\sum_a(\sqrt{p_a} - \sqrt{q_a})^2}}\right) = O\left(\frac{1}{\mathrm{d_H}(p,q)}\right)$$

when $c_a = (\sqrt{p_a} - \sqrt{q_a})/(\sqrt{p_a} + \sqrt{q_a})$.

4.3 Lower Bound in Model (ii)

We use the following version of the adversary lower bound from [12].

▶ **Theorem 7.** *Assume $\mathcal{A}$ is a quantum algorithm that makes T queries to the input string $x = (x_1, \ldots, x_n) \in D$, with $D = A^n$, and then either accepts or rejects. Let P and Q be two probability distributions on D, and p_x and q_y denote probabilities of x and y in P and Q, respectively. Let s_P and s_Q be acceptance probability of $\mathcal{A}$ when x is sampled from P and Q, respectively. Then,*

$$T = \Omega\left(\min_{j\in[n]} \frac{\delta_P^* \Gamma \delta_Q - \tau(s_P, s_Q)\|\Gamma\|}{\|\Gamma \circ \Delta_j\|}\right), \tag{7}$$

for any $D \times D$ matrix Γ with real entries. Here, $\delta_P[\![x]\!] = \sqrt{p_x}$ and $\delta_Q[\![y]\!] = \sqrt{q_y}$ are unit vectors in $\mathbb{R}^D$; for $j \in [n]$, the $D \times D$ matrix Δ_j is defined by $\Delta_j[\![x, y]\!] = 1_{x_j \neq y_j}$; and

$$\tau(s_P, s_Q) = \sqrt{s_P s_Q} + \sqrt{(1 - s_P)(1 - s_Q)} \le 1 - \frac{|s_P - s_Q|^2}{8}. \tag{8}$$

In our case, $\delta_P = \mu_p^{\otimes n}$ and $\delta_Q = \mu_q^{\otimes n}$. We construct Γ as a tensor power $G^{\otimes n}$, where G is an $A \times A$ matrix satisfying

$$G\mu_q = \mu_p, \qquad \|G\| = 1, \qquad \text{and} \qquad \|G \circ \Delta\| \text{ is as small as possible,}$$

where Δ is the $A \times A$ matrix given by $A[\![a, b]\!] = 1_{a \neq b}$. Then,

$$\delta_P^* \Gamma \delta_Q = \|\Gamma\| = 1, \qquad \text{and} \qquad \|\Gamma \circ \Delta_j\| = \|G \circ \Delta\|,$$

and adv gives the lower bound of $\Omega\big(1/\|G \circ \Delta\|\big)$.

We construct G as follows. Recall that α is the angle between μ_q and μ_p. Then, G is rotation by the angle α in the plane spanned by μ_q and μ_p and homothety with coefficient $\cos\alpha$ on its orthogonal complement. That is, in an orthonormal basis where the first two vectors span the plane of μ_q and μ_p, we have

$$G = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & \cdots & 0 \\ \sin\alpha & \cos\alpha & 0 & \cdots & 0 \\ 0 & 0 & \cos\alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \cos\alpha \end{pmatrix}.$$

Clearly, $G\mu_q = \mu_p$ and $\|G\| = 1$. Let $G' = G - \cos\alpha\, I$. We have

$$\|G \circ \Delta\| = \|G' \circ \Delta\| \leq 2\|G'\| = 2\sin\alpha = O\big(\mathrm{d_H}(p,q)\big).$$

For the inequality we used that $\gamma_2(\Delta) \leq 2$, see [24, Theorem 3.4]. This gives the required lower bound.

5 Summary and Future Work

In this paper we considered quantum algorithms dealing with classical probability distributions. We identified four different models, and proved various relations between them. We conjecture that models (i), (ii) and (iv) are equivalent.

Also, we considered the problem of distinguishing two probability distributions and obtained precise characterisation of its quantum query complexity in all four models in terms of Hellinger distance between the probability distributions. The complexity turned out to be exactly quadratically smaller than the classical complexity of this problem for all pairs of distributions.

We showed that the corresponding algorithm can be implemented efficiently given that the probability distributions p and q can be handled efficiently. We also compared our algorithm with a more standard approach using rejection sampling and amplitude estimation.

This raises a number of interesting open problems. The first one is to prove or disprove the conjecture that models (i) and (iv) are equivalent. Another interesting problem is to come up with a nice γ_2-characterisation of probability distribution oracles like gamma2StatePreparing characterises state-generating oracles. Unfortunately, we do not have any hypothesis of how this characterisation might look like. Finally, we would be interested in further quantum algorithms based on techniques of Section 4.2.

References

1. Scott Aaronson and Yaoyun Shi. Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *Journal of the ACM*, 51(4):595–605, 2004. doi:10.1145/1008731.1008735.
2. Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47–82, 2007. doi:doi.org/10.1137/060648829.
3. Andris Ambainis. Polynomial Degree and Lower Bounds in Quantum Complexity: Collision and Element Distinctness with Small Range. *Theory of Computing*, 1:37–46, 2005.
4. Andris Ambainis, Aleksandrs Belovs, Oded Regev, and Ronald de Wolf. Efficient Quantum Algorithms for (Gapped) Group Testing and Junta Testing. In *Proc. of 27th ACM-SIAM SODA*, pages 903–922, 2016. doi:10.1137/1.9781611974331.ch65.

5 Andris Ambainis, Loïck Magnin, Martin Rötteler, and Jérémie Roland. Symmetry-assisted adversaries for quantum state generation. In *Proc. of 26th IEEE CCC*, pages 167–177, 2011. doi:10.1109/CCC.2011.24.

6 Alp Atıcı and Rocco A. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005. doi:10.1007/s11128-005-0001-2.

7 Ziv Bar-Yossef. *The Complexity of Massive Data Set Computations*. PhD thesis, UC Berkeley, 2002.

8 Aleksandrs Belovs. Learning-graph-based Quantum Algorithm for k-distinctness. In *Proc. of 53rd IEEE FOCS*, pages 207–216, 2012. doi:10.1109/FOCS.2012.18.

9 Aleksandrs Belovs. Span programs for functions with constant-sized 1-certificates. In *Proc. of 44th ACM STOC*, pages 77–84, 2012. doi:10.1145/2213977.2213985.

10 Aleksandrs Belovs. Quantum Algorithms for Learning Symmetric Juntas via the Adversary Bound. *Computational Complexity*, 24(2):255–293, 2015. doi:10.1007/s00037-015-0099-2.

11 Aleksandrs Belovs. Variations on Quantum Adversary, 2015. arXiv:1504.06943.

12 Aleksandrs Belovs, Gilles Brassard, Peter Høyer, Marc Kaplan, Sophie Laplante, and Louis Salvail. Provably secure key establishment against quantum adversaries. In *Proc. of 12th TQC*, volume 73 of *LIPIcs*, pages 3:1–3:17. Dagstuhl, 2018.

13 Aleksandrs Belovs and Ben W. Reichardt. Span programs and quantum algorithms for st-connectivity and claw detection. In *Proc. of 20th ESA*, volume 7501 of *LNCS*, pages 193–204. Springer, 2012. doi:10.1007/978-3-642-33090-2_18.

14 Rajendra Bhatia. *Positive definite matrices*. Princeton University Press, 2009.

15 Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *Proc. of 3rd LATIN*, volume 1380 of *LNCS*, pages 163–169. Springer, 1998. doi:10.1007/BFb0054319.

16 Sergey Bravyi, Aram W. Harrow, and Avinatan Hassidim. Quantum algorithms for testing properties of distributions. *IEEE Transactions on Information Theory*, 57:3971–3981, 2011. doi:10.1109/TIT.2011.2134250.

17 Nader H. Bshouty and Jeffrey C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136–1153, 1998. doi:10.1137/S0097539795293123.

18 Sourav Chakraborty, Eldar Fischer, Arie Matsliah, and Ronald de Wolf. New Results on Quantum Property Testing. In *Proc. of 30th FSTTCS*, volume 8 of *LIPIcs*, pages 145–156. Dagstuhl, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.145.

19 András Gilyén and Tongyang Li. Distributional property testing in a quantum world, 2019. arXiv:1902.00814.

20 Yassine Hamoudi and Frédéric Magniez. Quantum Chebyshev's Inequality and Applications, 2018. arXiv:1807.06456.

21 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proc. of 39th ACM STOC*, pages 526–535, 2007. doi:10.1145/1250790.1250867.

22 Michael Jarret, Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum Algorithms for Connectivity and Related Problems. In *Proc. of 26th ESA*, volume 112 of *LIPIcs*, pages 49:1–49:13. Dagstuhl, 2018. doi:10.4230/LIPIcs.ESA.2018.49.

23 Troy Lee, Frédéric Magniez, and Miklos Santha. A learning graph based quantum query algorithm for finding constant-size subgraphs. *Chicago Journal of Theoretical Computer Science*, 2012(10), 2012.

24 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proc. of 52nd IEEE FOCS*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

25 Tongyang Li and Xiaodi Wu. Quantum query complexity of entropy estimation. *IEEE Transactions on Information Theory*, page 1–1, 2018. doi:10.1109/TIT.2018.2883306.

26 Ashley Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A*, 471(2181), 2015. doi:10.1098/rspa.2015.0301.

27 Ashley Montanaro. The quantum complexity of approximating the frequency moments. *Quantum Information & Computation*, 16:1169–1190, 2016.

28 Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proc. of 50th IEEE FOCS*, pages 544–551, 2009. doi:10.1109/FOCS.2009.55.

29 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proc. of 22nd ACM-SIAM SODA*, pages 560–569, 2011. doi:10.1137/1.9781611973082.44.

30 Mark Zhandry. A Note on the Quantum Collision and Set Equality Problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.