Learning-Graph-Based Quantum Algorithm for k-distinctness

Aleksandrs Belovs[*]

Abstract

We present a quantum algorithm solving the k-distinctness problem in $O\left(n^{1-2^{k-2}/(2^k-1)}\right)$ queries with a bounded error. This improves the previous $O(n^{k/(k+1)})$-query algorithm by Ambainis. The construction uses a modified learning graph approach. Compared to the recent paper by Belovs and Lee [6], the algorithm doesn't require any prior information on the input, and the complexity analysis is much simpler.

The complete version of the paper is available as [4].

1 Introduction

The *element distinctness* problem consists of computing function $f\colon [m]^n \to \{0,1\}$ that evaluates to 1 iff there is a pair of equal elements in the input, i.e., $f(x_1,\ldots,x_n)=1$ iff $\exists i \neq j : x_i = x_j$. (Here we use notation $[n]=\{1,2,\ldots,n\}$.) The quantum query complexity of the element distinctness problem is well understood. It is known to be $\Theta(n^{2/3})$, with the algorithm given by Ambainis [3], and the lower bound shown by Aaronson and Shi [1] and Kutin [11] for the case of large alphabet size $\Omega(n^2)$, and by Ambainis [2] in the general case.

Ambainis' algorithm for the element distinctness problem was the first application of the quantum random walk framework to a "natural" problem (i.e., one seemingly having little relation to random walks), and it had significantly changed the way quantum algorithms have been developed since then. The core of the algorithm is quantum walk on the Johnson graph. This primitive has been reused in many other algorithms: triangle detection in a graph given by its adjacency matrix [14], matrix product verification [8], restricted range associativity [10], and others. Given that the behaviour of quantum walk is well-understood for arbitrary graphs [15, 13], it is even surprising that the applications have been mostly limited to the Johnson graph.

The *k-distinctness problem* is a direct generalization of the element distinctness problem. Given the same input, the function evaluates to 1 iff there is a set of k input elements that are all equal, i.e., a set of indices $a_1,\ldots,a_k \in [n]$ with $a_i \neq a_j$ and $x_{a_i} = x_{a_j}$ for all $i \neq j$.

The situation with the quantum query complexity of the k-distinctness problem is not so clear. (In this paper we assume $k = O(1)$, and consider the complexity of k-distinctness as $n \to \infty$.) As element distinctness reduces to k-distinctness by repeating each element $k-1$ times, the lower bound of $\Omega(n^{2/3})$ carries over to the k-distinctness problem (this argument is attributed to Aaronson in Ref. [3]). This simple lower bound is the best one known so far.

In the same paper [3] with the element distinctness algorithm, Ambainis applied quantum walk on the Johnson graph in order to solve the k-distinctness problem. This resulted in a quantum algorithm with query complexity $O(n^{k/(k+1)})$. This was the best known algorithm for this problem prior to this paper.

The aforementioned algorithms work by searching for a small subset of input variables such that the value of the function is completely determined by the values therein. For instance,

[*] Faculty of Computing, University of Latvia, stiboh@gmail.com.

the values of two input variables are sufficient to claim the value of the element distinctness function is 1, provided their values are equal. This is formalized by the notion of certificate complexity as follows.

An *assignment* for a function $f\colon \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subseteq [m]^n$ is a function $\alpha\colon S \to [m]$ with $S \subseteq [n]$. The *size* of α is $|S|$. An input $x = (x_i) \in [m]^n$ *satisfies* assignment α if $\alpha(i) = x_i$ for all $i \in S$. An assignment α is called a *b-certificate* for f, with $b \in \{0,1\}$, if $f(x) = b$ for any $x \in \mathcal{D}$ satisfying α. The *certificate complexity* $C_x(f)$ of f on x is defined as the minimal size of a certificate for f that x satisfies. The b-certificate complexity $C^{(b)}(f)$ is defined as $\max_{x\in f^{-1}(b)} C_x(f)$. Thus, for instance, 1-certificate complexity of element distinctness is 2, and 1-certificate complexity of triangle detection is 3.

Soon after the Ambainis' paper, it was realized [9] that the algorithm developed for k-distinctness can be used to evaluate, in the same number of queries, any function with 1-certificate complexity equal to k. Now we know that for some functions this algorithm is tight, due to the lower bound for the k-sum problem [7]. The goal of the k-sum problem is to detect, given n elements of an Abelian group as input, whether there are k of them that sum up to a prescribed element of the group. The k-sum problem is noticeable in the sense that, given any $(k-1)$-tuple of input elements, one has absolutely no information on whether they form a part of an (inclusion-wise minimal) 1-certificate, or not.

The aforementioned applications of the quantum walk on the Johnson graph (triangle finding, etc.) went beyond the $O(n^{k/(k+1)})$ upper bound by utilizing additional relations between the input variables: the adjacency relation of the edges for the triangle problem, row-column relations for the matrix products, and so on. For instance, two edges in a graph cannot be a part of a 1-certificate for the triangle problem, if they are not adjacent.

The k-distinctness problem is different in the sense that it does not possess any structure of the variables. But it does possess a relation between the *values* of the variables: two elements cannot be a part of a 1-certificate if their values are different. However, it seems that quantum walk on the Johnson graph fails to utilize this structure efficiently.

In this paper, we construct a quantum algorithm that solves the k-distinctness problem in $O\left(n^{1-2^{k-2}/(2^k-1)}\right)$ queries. Note that $O\left(n^{1-2^{k-2}/(2^k-1)}\right) = o(n^{3/4})$. Thus, our algorithm solves k-distinctness, for arbitrary k, in asymptotically less queries than the best previously known algorithm solves 3-distinctness.

2 Techniques used

The algorithm is developed in the learning graph model. The learning graph is a novel way of construction quantum query algorithms. Somehow, it may be thought as a way of designing a more flexible quantum walk than just on the Johnson graph. And compared to the quantum walk design paradigms from Ref. [15, 13], it is easier to deal with. In particular, it does not require any spectral analysis of the underlying graph.

A learning graph is a randomized procedure for loading values of the variables with the goal of convincing someone the value of the function is 1. For each input $x \in f^{-1}(1)$, the designer of the learning graph builds its own procedure. The goal is to load a 1-certificate for x. The value of the complexity of the learning graph arises from the interplay between the procedures for different inputs.

Compared to the previous applications of learning graphs [5, 16, 12], the model is changed significantly. The previous approaches used the so-called non-adaptive model of learning graphs. In this model, the learning graph does not depend on the values of the variables loaded so far. The values of the variables come into consideration only at the very end, when it is claimed that a complete 1-certificate is loaded. This model is easy to analyse, that explains its relative popularity, but it has strong limitations. In particular, this model fails in developing an algorithm for k-distinctness with complexity better than $O(n^{k/(k+1)})$ for the reason the same algorithm would solve the k-sum problem as well.

The learning graph developed in this paper uses the values of the loaded variables along the way. But, in order to get the claimed complexity, a number of other alterations are introduced. Not the complete list of the values of loaded variables is stored, but only a partial one, with irrelevant values replaced by a placeholder. The distinction between relevant and irrelevant values is determined by the values of already loaded variables. The introduction of irrelevant values makes it possible to reduce the cost of their loading akin a repeated application of the Grover search for relevant values.

In order to reduce the complexity, the elements of the 1-certificate are loaded at the very end. The addition of these elements may spoil the distinction of relevant and irrelevant values, and the algorithm may go astray. We show that this may happen at a limited rate only, and develop a fault-tolerant version of the learning graph, capable of dealing with this problem.

References

[1] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[2] A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1:37–46, 2005, `arXiv:quant-ph/0305179`.

[3] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37:210–239, 2007, `arXiv:quant-ph/0311001`.

[4] A. Belovs. Learning-graph-based quantum algorithm for k-distinctness. 2012, `arXiv:1205.1534`.

[5] A. Belovs. Span programs for functions with constant-sized 1-certificates. In *Proc. of 44th ACM STOC*, pages 77–84, 2012, `arXiv:1105.4024`.

[6] A. Belovs and T. Lee. Quantum algorithm for k-distinctness with prior knowledge on the input. 2011, `arXiv:1108.3022`.

[7] A. Belovs and R. Špalek. Adversary lower bound for the k-sum problem. 2012, `arXiv:1206.6528`.

[8] H. Buhrman and R. Špalek. Quantum verification of matrix products. In *Proc. of 17th ACM-SIAM SODA*, pages 880–889, 2006, `arXiv:quant-ph/0409035`.

[9] A. Childs and J. Eisenberg. Quantum algorithms for subset finding. *Quantum Information & Computation*, 5(7):593–604, 2005, `arXiv:quant-ph/0311038`.

[10] S. Dörn and T. Thierauf. The quantum query complexity of algebraic properties. In *Proc. of 16th FCT*, volume 4639, pages 250–260. Springer-Verlag, 2007, `arXiv:0705.1446`.

[11] S. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1(1):29–36, 2005.

[12] T. Lee, F. Magniez, and M. Santha. A learning graph based quantum query algorithm for finding constant-size subgraphs. 2011, arXiv:1109.5135.

[13] F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. In *Proc. of 39th ACM STOC*, pages 575–584, 2007, arXiv:quant-ph/0608026.

[14] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007, arXiv:quant-ph/0310134.

[15] M. Szegedy. Quantum speed-up of markov chain based algorithms. In *Proc. of 45th IEEE FOCS*, pages 32–41, 2004.

[16] Y. Zhu. Quantum query complexity of subgraph containment with constant-sized certificates. 2011, arXiv:1109.4165.