BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 04/06 pp 41–55
Volume 3, Number 1, March 2020
DOI: 10.26599/BDMA.2019.9020016

On Quantum Methods for Machine Learning Problems Part I: Quantum Tools

Farid Ablayev, Marat Ablayev, Joshua Zhexue Huang, Kamil Khadiev, Nailya Salikhova, and Dingming Wu*

Abstract: This is a review of quantum methods for machine learning problems that consists of two parts. The first part, "quantum tools", presents the fundamentals of qubits, quantum registers, and quantum states, introduces important quantum tools based on known quantum search algorithms and SWAP-test, and discusses the basic quantum procedures used for quantum search methods. The second part, "quantum classification algorithms", introduces several classification problems that can be accelerated by using quantum subroutines and discusses the quantum methods used for classification.

Key words: quantum algorithm; quantum programming; machine learning

1 Introduction

Machine Learning (ML) is a rapidly developing area of computer science, motivated by a sharp growth in the volume of data that is being transferred, stored, and processed on a daily basis. Quantum methods bring new ideas and approaches to machine learning problems. Over the last decade, several survey and tutorial papers have been published by mathematicians with a variety of backgrounds in computer science, placing emphasis on different machine learning problems. These papers present several aspects of the application of quantum methods in machine learning[1–4].

In this review, we focus on two main problems of machine learning, namely the classification problem and the clustering problem. Classification is associated with learning algorithms that group data according to certain criteria, while clustering aims to find inherent patterns in the data. In this paper, we consider classification as the process by which algorithms group data based on predefined characteristics, which is known as supervised learning. Clustering, on the other hand, is the process of grouping data without predefined characteristics, which is known as unsupervised learning.

We present the basic quantum methods, such as the Grover quantum search and its variants as fundamental tools. We then demonstrate how these quantum tools can be useful for accelerating the computations of classification and clustering problems. Rather than attempting to cover all known classification and clustering problems for which quantum subroutines can be useful, we have chosen what we consider to be typical tasks as references to demonstrate the power of quantum subroutines.

The first part of the review is organized as follows. First, we present the basic notions and formalizations of qubits and quantum registers, and the quantum states of qubits and quantum registers. We show the difference between the classical and quantum registers and their states, and define the transformations of the quantum states of quantum registers and methods for the extraction of information. We present the quantum circuit, quantum query algorithm model, and quantum branching program as models for the realizations of quantum algorithms. In the main section, we present

• Joshua Zhexue Huang and Dingming Wu are with the College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518000, China. E-mail: zx.huang@szu.edu.cn; dingming@szu.edu.cn.
• Farid Ablayev, Marat Ablayev, Kamil Khadiev, and Nailya Salikhova are with the Kazan Federal University, Kazan 42008, Russia. E-mail: fablayev@gmail.com; mablayev@gmail.com; kamilhadi@gmail.com; nailyasalikhova66@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2019-09-10; accepted: 2019-09-25

the basic quantum procedures used in quantum search algorithms. We not only give a formal description of the procedures, but also present them in a form that can be used for constructing classification problems and present recent sources for readers to find realizations of such procedures.

2 Quantum Computing Basics

In this section, we present the basic notions of quantum computations as found in Ref. [5]. Jozsa[6] also provided some excellent notes on quantum computations.

2.1 Qubit

The notion of quantum bit (qubit) is the basis of quantum computations. Qubit is the quantum version of the classical binary bit physically realized with a two-state device. Just like a binary digit, there are two possible outcomes for the measurement of a qubit: the value 0 or 1. Whereas the state of a classical bit can only be either 0 or 1, the general state of a qubit according to quantum mechanics can be any coherent superposition of both, which allows computation on 0 and 1 simultaneously. Such a phenomenon is known as the quantum parallelism.

Formally, a qubit's state is a column vector $|\psi\rangle$ from the two-dimensional Hilbert space $\mathcal{H}^2$, i.e.,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

where vectors $|0\rangle$ and $|1\rangle$ are the orthonormal bases of $\mathcal{H}^2$, and $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. Numbers α and β are called amplitudes.

The Bloch sphere is a representation of a qubit's state as a point on a three-dimensional unit sphere (see Fig. 1). Consider a qubit in state $|\psi\rangle$ such that $|\alpha|^2 + |\beta|^2 = 1$, we can then represent amplitudes as

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + \mathrm{e}^{\mathrm{i}\phi}\sin\frac{\theta}{2}\,|1\rangle\,, 0 \leqslant \phi < 2\pi, 0 \leqslant \theta \leqslant \pi \tag{2}$$

where i is the imaginary unit, ϕ is the azimuthal angle, and θ is a polar angle of a point on the bloch spere in Fig. 1.

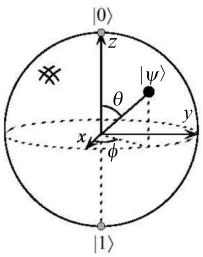


Fig. 1 Bloch sphere.

If we consider only real values for α and β, then the state of a qubit is a point on a unit circle (see Fig. 2). Thus, in the case of real-valued amplitudes, the state of a qubit is

$$|\psi\rangle = \cos\theta\,|0\rangle + \sin\theta\,|1\rangle\,, \theta \in [0, 2\pi) \tag{3}$$

2.2 States of quantum registers

A quantum register is an isolated quantum mechanical system composed of n qubits (a quantum n-register). A quantum n-register can represent the superposition of 2^n states. This allows us to compute on 2^n states simultaneously. This phenomenon of quantum parallelism is a potential advantage of quantum computational models.

Formally, a quantum state $|\psi\rangle$ of a quantum n-register is described as follows. Let $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ be a binary sequence. Then the tensor product $|\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \cdots \otimes |\sigma_n\rangle$ is denoted by $|\sigma\rangle$. Let $Basis = \{|00\cdots0\rangle, |00\cdots1\rangle, \ldots, |bin(i-1)\rangle, \ldots, |11\cdots1\rangle\}$ be a set of orthonormal vectors, where $bin(i)$ is a binary representation of i. *Basis* forms a basis for 2^n dimensional Hilbert space $\mathcal{H}^{2^n}$. The basis vectors from *Basis* can also be represented in brief as $|0\rangle, \ldots, |2^n - 1\rangle$. Usually, *Basis* is usually referred to as computational basis.

A quantum state $|\psi\rangle$ of a quantum n-register is a complex valued unit vector in 2^n-dimensional Hilbert space $\mathcal{H}^{2^n}$ that is described as a linear combination of basis vectors $|i\rangle, i \in \{0, \ldots, 2^n - 1\}$:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\,, \quad \text{with} \quad \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1 \tag{4}$$

where $|\alpha_i|^2$ expresses the probability to find the n-register in state $|i\rangle$ when state $|\psi\rangle$ is measured with regard to *Basis*. We say that state $|\psi\rangle$ is in the superposition of basis vector $|i\rangle$ with amplitude α_i. We will also use the notation $(\mathcal{H}^2)^{\otimes n}$ for Hilbert space $\mathcal{H}^{2^n}$ to outline the fact that vectors are the states of a quantum n-register.

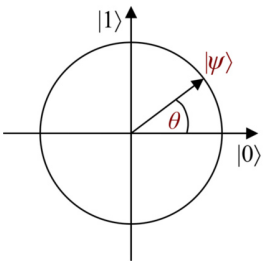


Fig. 2 Qubit's state with real-value amplitudes.

If a state $|\psi\rangle \in (\mathcal{H}^2)^{\otimes n}$ can be decomposed to the tensor product of several single qubits, i.e., $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$, $|\psi\rangle$, it is considered "not entangled". Otherwise, if the state can not be decomposed to the tenser product of several single qubits, it is called "entangled". An example of entangled states are Einstein, Podolsky, and Rosen (EPR)-pairs.

2.3 Transformations of quantum states

Quantum mechanics postulates that transformations of quantum states $|\psi\rangle \in (\mathcal{H}^2)^{\otimes n}$ (of a quantum n-register) are mathematically determined by unitary operators

$$\left|\psi'\right\rangle = U\,|\psi\rangle \tag{5}$$

where U is a $2^n \times 2^n$ unitary matrix. Such a unitary transformation is acting on the quantum n-register. A unitary matrix U can be written in the exponential form

$$U = \mathrm{e}^{\mathrm{i}W} \tag{6}$$

where W is a Hermitian matrix[7].

2.4 Basic transformations of quantum states

Qubit transformations are rotations of θ near $\hat{x}$, $\hat{y}$, and $\hat{z}$ axes of the Bloch sphere:

$$\begin{cases} R_{\hat{x}}(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\mathrm{i}\sin\frac{\theta}{2} \\ -\mathrm{i}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}; \\ R_{\hat{y}}(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}; \\ R_{\hat{z}}(\theta) = \begin{pmatrix} \mathrm{e}^{-\frac{\mathrm{i}\theta}{2}} & 0 \\ 0 & \mathrm{e}^{\frac{\mathrm{i}\theta}{2}} \end{pmatrix} \end{cases} \tag{7}$$

Below are several basic transformations (unitary matrices).

- I is an identity operator. That is, $I = R_{\hat{x}}(0) = R_{\hat{y}}(0) = R_{\hat{z}}(0)$.

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{8}$$

- X is a *NOT* operator. *NOT* flips the state of a qubit. It is a special case of $R_{\hat{x}}(\theta)$—a rotation around the X-axis of the Bloch sphere by π.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{9}$$

- S and T are phase transformation operators.

$$\begin{cases} S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}; \\ T = \pi/8 = \mathrm{e}^{\mathrm{i}\pi/8} \begin{pmatrix} \mathrm{e}^{-\mathrm{i}\pi/8} & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\pi/8} \end{pmatrix} \end{cases} \tag{10}$$

They are special cases of the rotation around the Z-axis of the Bloch sphere: S is rotation by $\frac{\pi}{2}$ and T is rotation by $\frac{\pi}{4}$.

- σ_z is the Pauli-Z gate. It is a special case of $R_{\hat{z}}(\theta)$—a rotation around the Z-axis of the Bloch sphere by π.

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{11}$$

- H is a Hadamard operator. It is the combination of two rotations: around the Z-axis of the Bloch sphere by π and around the Y-axis of the Bloch sphere by $\frac{\pi}{2}$.

$$\begin{cases} H = R_{\hat{y}}\left(\frac{\pi}{2}\right) R_{\hat{z}}(\pi); \\ H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{cases} \tag{12}$$

The Hadamard operator has several useful properties. Firstly, since $H = H^*$, for any state of a qubit $|\psi\rangle$, we have $H(H|\psi\rangle) = |\psi\rangle$. Secondly, the Hadamard operator creates a superposition of the basis states of a qubit with equal probabilities. The Hadamard operator maps the $|0\rangle$ to $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$. A measurement of this qubit has equal probabilities to obtain basis states $|0\rangle$ or $|1\rangle$. The Hadamard gate is usually used to initialize the state of a qubit. Thirdly, if a one-qubit state is in the form $|\psi_0\rangle = \frac{\mathrm{e}^{\mathrm{i}\phi}}{\sqrt{2}}|0\rangle + \frac{\mathrm{e}^{-\mathrm{i}\phi}}{\sqrt{2}}|1\rangle$, applying of the Hadamard operator to this qubit allows for the transfer of phase information into amplitudes. After applying the Hadamard gate to $|\psi_0\rangle$, we obtain $|\psi_1\rangle = H\,|\psi_0\rangle = \cos\phi\,|0\rangle + \sin\phi\,|1\rangle$.

2.5 Quantum circuits

Circuit is a simple and visual way of representing a sequence of transformations of register states. A classical Boolean circuit is a finite directed acyclic graph with *AND*, *OR*, and *NOT* gates. It has n input nodes, which contain n input bits (a state of n-register). The internal nodes are *AND*, *OR*, and *NOT* gates, and there are one or more designated output nodes. The initial input bits are fed into *AND*, *OR*, and *NOT* gates according to the circuit, and eventually the output nodes assume some values. A circuit computes a Boolean function $f : \{0,1\}^n \to \{0,1\}^m$, if the output nodes get the value $f(\sigma)$ for every input $\sigma \in \{0,1\}^n$.

A quantum circuit (also called a quantum network or quantum gate array) acting on the quantum register is a realization of the transformation of quantum states

(states of a quantum register). A quantum circuit generalizes the idea of the classical circuit, replacing the *AND*, *OR*, and *NOT* gates with elementary operators (quantum gates). A quantum gate is a unitary operator on a small (usually 1, 2, or 3) number of qubits. Mathematically, if gates are applied in a parallel way to different parts of the quantum register, then they can be composed by taking tensor products. If gates are applied sequentially, then they can be composed by taking the ordinary product.

Figure 3 illustrates the Hadamard operator (or gate) and Fig. 4 illustrates the *NOT* or X gate. If the rectangle of a gate crosses a line of a qubit then the gate is applied to this qubit. If the rectangle crosses several lines of qubits, then it is applied to all of these qubits.

Controlled quantum transformation. Controlled operator $C^q(U)$ (the quantum "if then else" operator) is implemented by a unitary transformation U that is applied only if some condition for q qubits is fulfilled. Supposing that qubits $|c_1\rangle$, $|c_2\rangle, \ldots, |c_q\rangle$ are in one of basis states $\{|0\rangle, |1\rangle\}$, Fig. 5 shows a quantum circuit that implements operator $C^q(U)$, which applies a unitary transformation U controlled by $|c_1\rangle$, $|c_2\rangle, \ldots, |c_q\rangle$ qubits with the condition that all controlling qubits are in state $|1\rangle$. This condition is represented on the circuit diagram in Fig. 5 by black points. This operator can be represented as

$$C^q(U)\,|c_1\rangle\,|c_2\rangle \cdots |c_q\rangle\,|\psi\rangle = |c_1\rangle\,|c_2\rangle \cdots |c_q\rangle\,U\,|\psi\rangle \tag{13}$$

If all the control qubits are in state $|1\rangle$, the (unitary) transformation U is applied. If one qubit is in state $|0\rangle$, the identical transformation is applied.

The quantum circuit shown in Fig. 6 generalizes the above controlled quantum transformation. Supposing

Fig. 3 Hadamard gate.

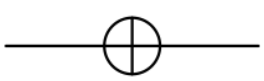


Fig. 4 *NOT* or *X* gate.

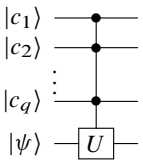


Fig. 5 Controlled quantum transformation.

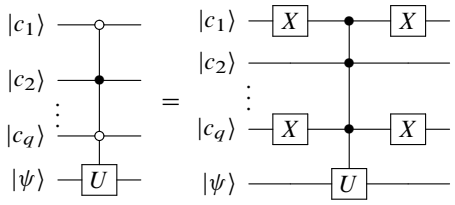


Fig. 6 General form of controlled quantum circuit.

that we need to apply transformation U for $|\psi\rangle$ with the condition that $c_1 = 0, c_2 = 1, \ldots, c_q = 0$, such a circuit is shown on the left side of Fig. 6. The white and black points describe the conditions of the controlling qubits (0 and 1) for applying transformation U for $|\psi\rangle$.

The technical construction of such a circuit is shown on the right side of Fig. 6. The inversion operator X is added to those controlling qubits that should be 0. Operator U will then be applied if all controlling qubits are 1. After the operator U has been applied, we "return" the controlling qubits in their initial states.

Using this mechanism, an arbitrary control condition can be implemented. Such controlled transformations are key transformations for quantum algorithms. The controlled operator $C^q(U)$ itself can be expressed as a unitary matrix. Two basic controlled operators are *CNOT* and *CCNOT* (or *2CNOT*).

The $CNOT(a, b)$ gate flips the second entry b (the target qubit) if and only if the first entry a (the control qubit) is $|1\rangle$ (see Fig. 7).

The *CCNOT* (or 2*CNOT*) is a Toffoli gate. The $2CNOT(a, b, c)$ gate flips the third entry c (the target qubit) if and only if the first entry a and the second entry b (the control qubits) are $|1\rangle$ (see Fig. 8).

At the abstract level, quantum state transformations are described in terms of linear transformations of U. Such transformations are implemented by affecting the quantum register, which means that linear transformations of U are implemented by "controlled" actions of type O on the quantum register:

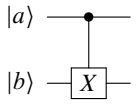


Fig. 7 *CNOT* gate.

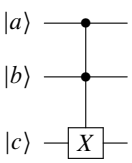


Fig. 8 *CCNOT* gate.

$$O : |i\rangle \to \beta\,|j\rangle\,.$$

Note that the quantum register can be prepared in some specific initial quantum state. A quantum algorithm acts on the quantum register to obtain the necessary quantum state.

2.6 Information extraction (measurement)

There is only one way to extract information from a state of a quantum n-register for the "macro" world, which is to measure the state of the quantum register. Measurement can also be considered as the second type of quantum operators, with unitary transformations being the first.

Different measurements are considered in quantum computation theory. In this review, we will use only a measurement in the respect of "computational basis". This measurement is described as follows. If we measure the quantum state $\sum_i \alpha_i\,|i\rangle$, then we get one of the basis states $|i\rangle$ with a probability $|\alpha_i|^2$.

We can also perform partial measurement of the state of a quantum register. Considering the case of a quantum 2-register, let $|\psi\rangle$ be a state of such a quantum 2-register:

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle \quad (14)$$

Imagining that we measure the first qubit of the 2-register, the probability of getting $|0\rangle$ is the same as if we measure both qubits and sum over all of the probabilities of the $|0\rangle$ outcome on the first qubit:

$$Pr(|0\rangle) = Pr(|00\rangle) + Pr(|01\rangle) = |\alpha_{00}|^2 + |\alpha_{01}|^2 \quad (15)$$

The state of the second qubit after the measurement is

$$|\psi_0\rangle = \frac{\alpha_{00}\,|0\rangle + \alpha_{01}\,|1\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \quad (16)$$

In a similar way, we get the probability of getting 1.

$$Pr(|1\rangle) = Pr(|10\rangle) + Pr(|11\rangle) = |\alpha_{10}|^2 + |\alpha_{11}|^2 \quad (17)$$

The state of the second qubit after the measurement is

$$|\psi_1\rangle = \frac{\alpha_{10}\,|0\rangle + \alpha_{11}\,|1\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}} \quad (18)$$

Partial measurement of a state $|\psi\rangle$ of a quantum n-register for $n \geqslant 2$ is a direct generalization of the case of partially measuring state $|\psi\rangle$ of the quantum 2-register to the quantum n-register case.

3 Computational Models and Complexity Measurements

The computational complexity is based on the formalization of computation models. This section defines the main computational models that are used for quantum search problems.

The computational model oriented for Boolean functions is defined as

$$f : \{0,1\}^N \to \{0.1\} \quad (19)$$

Let $X = \{x_1, \ldots, x_N\}$ be a set of variables of function f. Note that the above computational model can be easily generalized for other general functions.

3.1 Decision trees and branching programs

We consider a deterministic version of the closely related Decision Tree (DT) and Branching Program (BP) models of computation[8]. They are closely related. Following are two versions of DTs and BPs for quantum generalization.

3.1.1 Graph representation

DT A is a directed leveled binary tree with a selected starting node (root node). All the nodes V of A are partitioned into levels $V_1, \ldots, V_\ell$. Level V_1 contains the starting node. Nodes at level V_i are connected to nodes at level V_{i+1}. At each node in the tree, a Boolean variable $x \in X$ is tested. Different from the DT model, the BP model is not a tree but a leveled acyclic directed graph in which there could be more than one in-going edges to a vertex at level V_i from the vertices at level V_{i-1}.

The computation of a DT and a BP starts from the root node. At each node in the graph, a Boolean variable $x \in X$ is tested. Depending on the outcome of the query, the algorithm proceeds to the $x = 0$ child or the $x = 1$ child of the node. Leaf nodes are marked by "0" or "1". If for an input $\sigma = \sigma_1, \ldots, \sigma_N$, the computation reaches a leaf node in the tree, it outputs the value of the function listed at this leaf node.

Usually, the DT model is used when seeking to minimize computational time (number of queries). The DT model can be arbitrarily wide. The BP model is typically used when seeking to minimize both computational time (number of queries) and memory (number of nodes).

3.1.2 Linear representation

Let $dim(A) = \max\limits_{1 \leqslant i \leqslant \ell} |V_i|$ and $d = dim(A)$. Let $S = \{s^1, \ldots, s^d\}$ be a set of d-dimensional column-vectors, where s^i is a vector with all components "0" except one "1" in the i-th position. We call vector $s \in S$ a state of A. State s at each level i, $1 \leqslant i \leqslant \ell$ represents a node (by "1" component) where A can be found at that level.

Next, let $X_i \subseteq X$ be a set of variables tested at level i. A transformation of state s at level i, $1 \leqslant i \leqslant \ell - 1$, is described by a set $\mathcal{Q}^i(X_i)$ of matrices that depend on the values of variables X_i tested at that level. State $s^{(0)} \in S$ is an initial state.

DT A can be formalized as

$$A = \langle S, \mathcal{Q}^1(X_1), \ldots, \mathcal{Q}^{\ell-1}(X_{\ell-1}), s^{(0)} \rangle \quad (20)$$

Computation on an input σ by A is presented as a sequence $s^{(0)} \to s^{(1)} \to \cdots \to s^{(\ell)}$ of state transformations determined by the structure of A and input σ. Let $s^{(i)}$ be the current state at level i, then the next state will be $s^{(i+1)} = Q^i(X_i, \sigma)s^{(i)}$, where matrix $Q^i(X_i, \sigma) \in \mathcal{Q}^i(X_i)$ is determined by σ for X_i variables. In the BP model, if x is a variable tested at level i, $\mathcal{Q}^i(X_i) = \{Q^i(x, 0), Q^i(x, 1)\}$.

3.1.3 Complexity measures

Time and space (memory) are the two main complexity measures used for computational models. For decision models of computation, the analogs of these complexity measures are query complexity and size complexity. Query complexity is the maximum number of queries that the algorithm can perform during computation, which is equal to the depth $D(A)$ that is the length of the longest path from the root to a leaf in the decision tree A. Size complexity is the width of A, denoted by $dim(A)$. In addition, bits are used to encode a state at levels of A, such that $size(A) = \lceil \log_2 dim(A) \rceil$. For function f, the complexity measure $D(f)$ refers to the minimum time needed for computing f, and both of $dim(f)$ and $size(f)$ refer to the space.

3.2 Quantum query algorithm

The Quantum Query Algorithm (QQA) is a generalization of the DT model. A QQA A for computing Boolean function $f(X)$ is based on a quantum $size(A)$-register (on a quantum system composed of $size(A)$ qubits). With $|\psi_{start}\rangle$ as an initial state, the procedure of computation is determined by the sequence $U_0, Q, U_1, \ldots, Q, U_\ell$ of operators that are $dim(A) \times dim(A)$ unitary matrices.

Algorithm A contains of two types of operators. Operators U_i are independent of input X, while Q is the query-operator of a fixed form that depends on the tested input X. The algorithm consists of performing $U_0, Q, U_1, \ldots, Q, U_\ell$ on $|\psi_{start}\rangle$ and measuring the result, as shown in Fig. 9.

The algorithm computes $f(X)$ on an input σ. The initial state $|\psi_{start}\rangle$ is transformed to a final quantum state $|\psi\rangle$ that can be measured to get value $f(\sigma)$. The final state $|\psi\rangle$ depends on the input σ, so $|\psi\rangle$ can be represented as $|\psi_\sigma\rangle$; the final state also depends on $f(\sigma)$, so $|\psi\rangle$ can be represented as $|\psi(f(\sigma))\rangle$ or $|\psi_{f(\sigma)}\rangle$. These are the notations used in the paper.

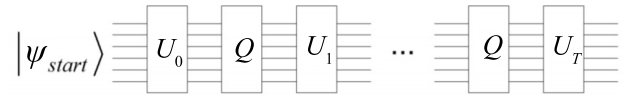


Fig. 9 Quantum query algorithm as a quantum circuit.

3.3 Quantum branching program

Quantum Branching Programs (QBPs)[9–11] are another known model of computations, and a generalization of the classical BP model. QBPs and QQAs are closely related, and each could be considered a variant of the other depending on the point of view. For example, in this paper, QBP can be considered as a variant of the QQA model, which can test only one input variable at a level of computation.

Following Ref. [11], a QBP A over the Hilbert space $\mathcal{H}^d$ is defined as

$$A = \langle \mathbb{TA}, |\psi_0\rangle \rangle \quad (21)$$

where $\mathbb{TA}$ is a sequence of l instructions, $\mathbb{TA}_j = \left(x_{i_j}, U_j(0), U_j(1)\right)$ is determined by the variable x_{i_j} tested on step j, and $U_j(0)$, $U_j(1)$ are unitary transformations in $\mathcal{H}^d$, $d = dim(A)$. Vectors $|\psi\rangle \in \mathcal{H}^d$ are called states (state vectors) of A, and $|\psi_0\rangle \in \mathcal{H}^d$ is the initial state of A. Quantum branching program as a quantum circuit is shown in Fig. 10. A computation of A on an input $\sigma = \sigma_1 \cdots \sigma_n \in \{0, 1\}^n$ is defined as follows.

(1) A computation of A starts from the initial state $|\psi_0\rangle$.

(2) The j-th instruction of A reads the input symbol σ_{i_j} (the value of x_{i_j}) and applies the transition matrix $U_j = U_j(\sigma_{i_j})$ to the current state $|\psi\rangle$ for obtaining the state $|\psi'\rangle = U_j(\sigma_{i_j})|\psi\rangle$.

(3) The final state is

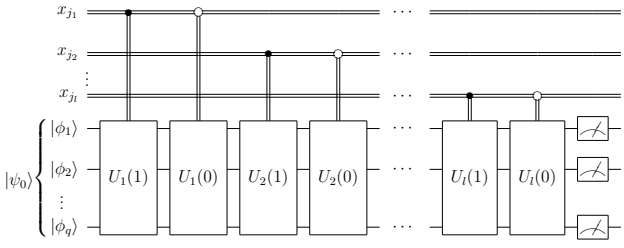


Fig. 10 Quantum branching program as a quantum circuit.

$$|\psi_\sigma\rangle = \left(\prod_{j=l}^{1} U_j(\sigma_{i_j})\right)|\psi_0\rangle \qquad (22)$$

BPs and QBPs are convenient computational models in complexity theory. It is easy and natural to define various restricted models for this computational model. The one that we use here is a read-once model, which has the restriction that each input variable can be tested at least once. In this case, we have $\ell = N$, meaning that the number of computational steps equal the number of input variables. Note that QQA and QBP in the above forms present a quantum algorithm in terms of linear transformations of quantum states by applying unitary $d \times d$ matrices U. Such an abstract linear presentation is important for the complexity analysis of a quantum algorithm. However, the programming of the quantum algorithm uses a different kind of presentation.

3.4 Programming-oriented presentation

3.4.1 QQA

One of the tools for constructing the algorithm is to change a quantum register. We present here as an example of important tool for inverting the sign of amplitude that is one of the main steps in Grover's search algorithm.

A w-register can be changed as follows. Let $S = \{|1\rangle, \ldots, |w\rangle\}$ be the basis states. Any $|a\rangle \in S$ is represented as $|i, j\rangle$, where i denotes variable x_i tested in state $|a\rangle$ and j corresponds to the j-th x_i testing at the level where $|a\rangle$ is located. The query Q performs the transformation (inverting the sign of amplitude):

$$Q : |i, j\rangle \mapsto (-1)^{x_i} |i, j\rangle .$$

Inverting the sign of amplitude for a state $|i\rangle$ where x_i is tested can be implemented using an additional qubit $|\varphi\rangle$. From $|\varphi\rangle = \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$, the query makes the following transformation:

$$Q : |i, j\rangle |\varphi\rangle \mapsto |i, j\rangle |\varphi \oplus x_i\rangle .$$

Property 1. Supposing $|\varphi\rangle = \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$, the operator Q performs the "inverting the sign of amplitude" procedure:

$$Q : |i, j\rangle |\varphi\rangle \mapsto (-1)^{x_i} |i, j\rangle |\varphi\rangle .$$

Proof. If $x_i = 0$, then $|i, j\rangle |\varphi \oplus x_i\rangle = |i, j\rangle |\varphi\rangle = (-1)^0 |i, j\rangle |\varphi\rangle$. If $x_i = 1$, then $|i, j\rangle |\varphi \oplus x_i\rangle = |i, j\rangle |\varphi \oplus 1\rangle = |i, j\rangle \frac{|0\oplus 1\rangle - |1 \oplus 1\rangle}{\sqrt{2}} = |i, j\rangle \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -|i, j\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = (-1)^1 |i, j\rangle |\varphi\rangle$. Therefore, $|i, j\rangle |\varphi \oplus x_i\rangle = (-1)^{x_i} |i, j\rangle |\varphi\rangle$.

After the last transformation, the state of the algorithm is measured with regard to $|0\rangle, \ldots, |w-1\rangle$ and the result is transformed into the answer of the algorithm according to a predefined rule.

Two most frequently considered types of quantum query algorithms are exact and bounded error algorithms. In this work, we will consider only bounded error algorithms. A quantum query algorithm A computes f with bounded error, if for every $(x_1, \ldots, x_N)$, for which $f(x_1, \ldots, x_N)$ is defined, the probability that A outputs $f(x_1, \ldots, x_N)$ as the answer is at least $2/3$.

3.4.2 QBP

We change the read-once QBP P to the following QBP A by modifying its register. Basis states $S = \{|1\rangle, \ldots, |d\rangle\}$ are equipped with ancillary qubits as follows. State $|a\rangle \in S$ will be modified by adding state $|k\rangle$ and qubit $|\phi\rangle$, where k is the index of variable $x \in X$ tested in the state $|a\rangle$ and $|\phi\rangle$, and presents a Boolean value of the input x tested. The new basis states are that S' will be $S' = \{|k\rangle |a\rangle |\phi\rangle : k \in \{1, \ldots, N\}, a \in \{1, \ldots, d\}, \phi \in \{0, 1\}\}$.

From the initial state of $|j_1\rangle |\psi_0\rangle |0\rangle$, where $|\psi_0\rangle$ is a starting state of Q, the transformation of A moves through the following sequence:

$$Q, U_1, Q, T, Q, U_2, Q, T, Q, \ldots, Q, U_N .$$

The matrix T defines a transition that changes the testing variable's index on the current level to the variable that is tested on the next level:

$$T : |j_z\rangle |a\rangle |\phi\rangle \to |j_{z+1}\rangle |a\rangle |\phi\rangle .$$

The matrix Q is a query, such that

$$Q : |k\rangle |a\rangle |\phi\rangle \to |k\rangle |a\rangle |\phi \oplus x_k\rangle .$$

U_i is an operator from QBP P that is oriented for acting on the basis states of the quantum state. U_i applies U_i^0 to $|a\rangle$ if $|\phi\rangle = |0\rangle$ and applies U_i^1 to $|a\rangle$ if $|\phi\rangle = |1\rangle$. We have to apply the query Q twice for one step of the algorithm in order to get state $|\phi\rangle = |0\rangle$ before testing the next input. More precisely, before applying Q, the qubit $|\phi\rangle$ is in the state of $|0\rangle$. The first application of Q converts $|k\rangle |a\rangle |0\rangle$ to state $|k\rangle |a\rangle |x_k\rangle$; the second application of Q converts $|k\rangle |a'\rangle |x_k\rangle$ to $|k\rangle |a'\rangle |0\rangle$.

3.5 Complexity measures

As the main complexity measures, we use the following:

• $size(A)$ is a number of qubits of the underlying quantum register.

• $dim(A)$ is the size of the quantum system; that is, the dimension of the quantum states of the register ($dim(A) = 2^{size(A)}$).

• $height(A)$ is the number of queries for a query model algorithm; $height(A) = N$ for a quantum branching program. The standard notation for a number of queries in the quantum computation literature for the quantum model is $Q(A)$ and for the deterministic model is $D(A)$.

• $time(A)$ is a number of basis gates in the quantum circuit that implements A.

4 Quantum Tools for Quantum Classification

This section presents some quantum subroutines which can be used both independently and as the part of other algorithms. In particular, the controlled normalization operator is used in the algorithm for constructing a superposition of the training set, and the operators for changing the phase of the quantum state by a value that is proportional to the amount of zeroes and for transferring quantum state phase information into the amplitudes subroutine are used in the first quantum k-nearest neighbor algorithm. The number incremenntation operation, *OR* operation, and overflow check subroutine are used in the second quantum k-nearest neighbor algorithm.

4.1 Quantum operators

4.1.1 Controlled normalization operator

This gate[12] is designed to divide one part of the quantum register into two parts, so that the coefficients of the untouched parts remain unchanged and the two new parts have the correct normalized coefficients.

Problem. Given a quantum $(n+2)$-register and its quantum state is in the form

$$|\psi_0\rangle = \frac{1}{\sqrt{p}}\sum_{i=1}^{k-1}|x_0^i,\ldots,x_{n-1}^i;00\rangle + \sqrt{\frac{p-k+1}{p}}|x_0^k,\ldots,x_{n-1}^k;11\rangle \tag{23}$$

How to achieve the following quantum state:

$$|\psi_1\rangle = \frac{1}{\sqrt{p}}\sum_{i=1}^{k-1}|x_0^i,\ldots,x_{n-1}^i;00\rangle + \frac{1}{\sqrt{p}}|x_0^k,\ldots,x_{n-1}^k;10\rangle + \sqrt{\frac{p-k}{p}}|x_0^k,\ldots,x_{n-1}^k;11\rangle \tag{24}$$

Solution. Given an n-register and a utility register of two qubits.

$$|\psi_0\rangle = \frac{1}{\sqrt{p}}|x_0^1,\ldots,x_{n-1}^1;00\rangle + \sqrt{\frac{p-1}{p}}|x_0^2,\ldots,x_{n-1}^2;11\rangle \tag{25}$$

where $0 \leqslant p \leqslant 1$, $(x_0^1,\ldots,x_{n-1}^1)$ and $(x_0^2,\ldots,x_{n-1}^2) \in \{0,1\}^n$, and the number p is given.

To separate the second term of the above state (the first utility qubit state is $|1\rangle$) into two parts, one where the state of the second utility qubit is $|0\rangle$ with coefficient $\frac{1}{\sqrt{p}}$ and the other where the state of the second utility qubit is $|1\rangle$ with coefficient $\sqrt{\frac{p-2}{p}}$; that is, to achieve the following quantum state:

$$|\psi_1\rangle = \frac{1}{\sqrt{p}}\left|x_0^1,\ldots,x_{n-1}^1;00\right\rangle + \frac{1}{\sqrt{p}}\left|x_0^2,\ldots,x_{n-1}^2;10\right\rangle + \sqrt{\frac{p-2}{p}}\left|x_0^2,\ldots,x_{n-1}^2;11\right\rangle \tag{26}$$

The same operator should convert $\left|\psi_0'\right\rangle$ to $\left|\psi_1'\right\rangle$, where

$$\left|\psi_0'\right\rangle = \frac{1}{\sqrt{p}}\left|x_0^1,\ldots,x_{n-1}^1;01\right\rangle + \sqrt{\frac{p-1}{p}}\left|x_0^2,\ldots,x_{n-1}^2;10\right\rangle \tag{27}$$

$$\left|\psi_1'\right\rangle = \frac{1}{\sqrt{p}}\left|x_0^1,\ldots,x_{n-1}^1;01\right\rangle + \sqrt{\frac{p-2}{p}}\left|x_0^2,\ldots,x_{n-1}^2;10\right\rangle - \frac{1}{\sqrt{p}}\left|x_0^2,\ldots,x_{n-1}^2;11\right\rangle \tag{28}$$

This can be achieved by applying the following controlled gate to the utility register $|u_1\rangle\,|u_2\rangle$:

$$CS = \begin{pmatrix} I & 0 \\ 0 & S \end{pmatrix}, S = \begin{pmatrix} \sqrt{\frac{p-2}{p-1}} & \frac{1}{\sqrt{p-1}} \\ \frac{-1}{\sqrt{p-1}} & \sqrt{\frac{p-2}{p-1}} \end{pmatrix} \tag{29}$$

where the first utility qubit $|u_1\rangle$ acts as a control qubit

and the second utility qubit $|u_2\rangle$ acts as a target qubit. Operator CS provides for the adjustment of amplitudes to the desired values.

Similarly, given the following quantum state:

$$|\psi_0\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^{k-1} |x_0^i, \ldots, x_{n-1}^i; 00\rangle + \sqrt{\frac{p-k+1}{p}} |x_0^k, \ldots, x_{n-1}^k; 11\rangle \quad (30)$$

The following quantum state

$$|\psi_1\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^{k-1} |x_0^i, \ldots, x_{n-1}^i; 00\rangle + \frac{1}{\sqrt{p}} |x_0^k, \ldots, x_{n-1}^k; 10\rangle + \sqrt{\frac{p-k}{p}} |x_0^k, \ldots, x_{n-1}^k; 11\rangle \quad (31)$$

can be obtained by applying the following controlled gate to the utility register $|u_1\rangle\,|u_2\rangle$:

$$CS = \begin{pmatrix} I & 0 \\ 0 & S^{p+1-k} \end{pmatrix},$$

$$S^{p+1-k} = \begin{pmatrix} \sqrt{\frac{p-k}{p-k+1}} & \frac{1}{\sqrt{p-k+1}} \\ \frac{-1}{\sqrt{p-k+1}} & \sqrt{\frac{p-k}{p-k+1}} \end{pmatrix} \quad (32)$$

4.1.2 Changing the phase

This operator[12] changes the phase of the state of the quantum register by a value that is proportional to the number of zeroes.

Problem. Given a state $|\phi_0\rangle = |m_1, \ldots, m_n\rangle$ of a quantum n-register, how to change the phase of the state by a value that is proportional to the number of qubits that is equal 0.

Solution. Apply the unitary operator $U = \mathrm{e}^{\mathrm{i}\frac{\pi}{2n}D_m}$, where $D_m = \sum_{k=1}^{n} \left(\frac{\sigma_z + I}{2}\right)$ is a linear combination of $\left(\frac{\sigma_z + I}{2}\right)$ matrices and σ_z is the Pauli-Z gate. The result quantum state is

$$|\phi_1\rangle = U\,|\phi_0\rangle = \mathrm{e}^{\mathrm{i}\frac{\pi}{2n}D_m}\,|m_1, \ldots, m_n\rangle \quad (33)$$

The result is the change of the phase by value $\frac{\pi}{2n}k$, where k is the number of zeroes.

4.1.3 Number incrementation operation

Problem. Given an n-bit number a, how to increment this number ($a = a + 1$).

Solution[13]. One utility qubit is initially set to $|1\rangle$. Let $a[0], \ldots, a[n-1]$ be the binary representation of number a. Increment by 1 flips the least significant bit. If it was flipped from 0 to 1, the addition should be stopped; otherwise, proceed to flip the next least significant bit. Repeat in this way until the bit is flipped from 0 to 1. The utility qubit can be viewed as a flag which signals the first time when a bit was flipped from 0 to 1. The utility qubit is $|1\rangle$ as long as it is required to continue flipping bits and becomes to $|0\rangle$ when bits can stop being flipped.

The time cost of the number incrementation operation is measured by the number of elementary gates (*NOT*, 2*CNOT*):

$$time = \begin{cases} 1, & \text{if} \quad n = 1; \\ 10, & \text{if} \quad n = 2; \\ 2n^2 + n - 5, & \text{if} \quad n \geqslant 3 \end{cases} \quad (34)$$

Algorithm 1 shows the procedure, where a is an n-bits binary number.

4.1.4 *OR* operation

Problem. Given t bits, how to apply an *OR* operation to them.

Solution. Algorithm 2 shows the procedure, where v is an n-qubit register. The algorithm requires $n-1$ utility qubits.

4.1.5 Overflow check

Problem. Given three integers b, t, and n, such that $t \leqslant n$, how to know if b is greater than or equal to $n-t$.

Solution. Consider integer k such that $2^{k-1} \leqslant n \leqslant$

Algorithm 1 **Increment (a, n)**

$|a\rangle \leftarrow a$ ▷ the number is described with n bits: $a[0], \ldots, a[n-1]$, where $a[n-1]$ is the most significant bit.
$|a\rangle \leftarrow |a\rangle \otimes |1\rangle$ ▷ initial state
for $i \in \{0, \ldots, n-2\}$ **do**
  $|a\rangle \leftarrow CNOT(a_n, a_i)|a\rangle$
  $|a\rangle \leftarrow CNOT(a_i, a_n)|a\rangle$
end for
$|a\rangle \leftarrow CNOT(a_n, a_{n-1})|a\rangle$
$|a\rangle \leftarrow NOT(a_n)|a\rangle$

Algorithm 2 ***OR*** **(v)**

$|v\rangle \leftarrow |v\rangle \otimes |0\rangle^{\otimes(n-1)}$ ▷ initial state
for $i \in \{1, \ldots, n\}$ **do**
  $|v\rangle \leftarrow NOT(v_i)|v\rangle$
end for
$|v\rangle \leftarrow 2CNOT(v_0, v_1, v_{n+1})|v\rangle$
for $i \in \{3, \ldots, n\}$ **do**
  $|v\rangle \leftarrow 2CNOT(v_i, v_{n+i-2}, v_{n+i-1})|v\rangle$
end for
$|v\rangle \leftarrow NOT(v_{2n-1})|v\rangle$
return $|v_{2n-1}\rangle$ ▷ result is the state of the last qubit

2^k, and let integer $l = 2^k - n$. Condition $b \geqslant n - t$ can then be rewritten as

$$\begin{cases} b + l \geqslant n + l - t, \\ b + l + t \geqslant 2^k \end{cases} \tag{35}$$

Let $a = l + t$. Condition $b \geqslant n - t$ is true if the sum $b + a$ overflows. Choose $\log_2 t$ as the most significant qubits of the result of this sum and use the previously described *OR* operation on them to check for overflow. If the result of this operation is 0 then the condition $b \geqslant n - t$ is met.

Algorithm 3 shows the procedure, where b is an n-bit number, and n and t are integers.

4.2 Quantum search

This section presents a quantum solutions for search problems that are faster than classical solutions.

4.2.1 Search problem

Given a function $f : [N] \to \{0, 1\}$, find the $x \in \{L, \ldots, R\} \subset [N]$ such that $f(x) = 1$. The function f is computed by Oracle.

4.2.2 Classical algorithm and complexity

If we consider deterministic algorithms, the only way to find the required x is the brute-force linear search algorithm. If we consider randomized (probabilistic) algorithms, for set $f^{-1}(1) = \{x : f(x) = 1\}$, there exist efficient algorithms. Let $t = |f^{-1}(1)|$, if one of $x \in [N]$ is randomly chosen, the probability of getting the right x is t/N. Therefore, in N/t repetitions, we can find $x \in f^{-1}(1)$ with high probability.

4.2.3 Quantum grover algorithm and complexity

Below, we present a basic "maximum/minimum search" problem that explores the Grover search algorithm. These quantum algorithms play essential roles as subroutines in different quantum algorithms for classification problems. Originally, the Grover algorithm was developed for the "single one search" problem[14].

Single one search problem. Given a function $f : [N] \to \{0, 1\}$, find the $x \in \{L, \ldots, R\} \subset [N]$, such that $f(x) = 1$. It is known that there is only one $x \in \{L, \ldots, R\}$ having $f(x) = 1$. The function f is computed by Oracle.

Consider a quantum register $|\psi\rangle$ with N quantum states or $\log_2 N$ qubits. The state $x \in [N]$ is corresponding to solution x. We start from the state of equal amplitudes for all states, then we apply query to Oracle operator $\mathcal{Q}$ that inverts the amplitude of $\hat{x} \in f^{-1}(1)$; i.e., $\mathcal{Q} : |i\rangle \to (-1)^{f(i)} |i\rangle$. After that we apply diffusion operator $\mathcal{D}$:

$$\mathcal{D} = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \cdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix} = H^{\otimes \log_2 N} R H^{\otimes \log_2 N} \tag{36}$$

$$R = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{37}$$

The diffusion operator rotates each amplitude near the mean of all amplitudes. The repetition of query and diffusion operators increase the amplitude of $\hat{x} \in f^{-1}(1)$. In $\frac{\pi}{4}\sqrt{N}$ steps, we will get the maximal amplitude of the solution $\hat{x}$. After measurement, $\hat{x}$ will be obtained with high probability. Algorithm 4 shows the procedure.

Property 2[14]**.** Grover's search algorithm A has query complexity of $height(A) = O(\sqrt{R - L}) = O(\sqrt{N})$ and error probability of $O\left(\frac{1}{N}\right)$ for the single one search problem.

In a case of multiple solutions for $t = |f^{-1}(1)| > 1$, the property below holds.

Property 3[15, 16]**.** Grover's search algorithm

Algorithm 3 Overflow_check (b, n, t)

$|b\rangle \leftarrow b$
$k \leftarrow$ choose k from condition $2^{k-1} \leqslant n \leqslant 2^k$
$l \leftarrow 2^k - n$
$a \leftarrow l + t$
$|b\rangle \leftarrow |b\rangle + a$
$m \leftarrow$ choose $\log_2 t$ as the most significant qubits of $|b\rangle$
return $OR|m\rangle$

Algorithm 4 Grover (L, R, f)

$|\psi\rangle \leftarrow |0\rangle$ ▷ The initial value
$|\psi\rangle \leftarrow H^{\otimes n}|\psi\rangle$ ▷ We obtain equal amplitudes
for $i \in \{1, \ldots, \frac{\pi}{4}\sqrt{N}\}$ **do**
  $|\psi\rangle \leftarrow \mathcal{O}|\psi\rangle$ ▷ Query.
  $|\psi\rangle \leftarrow \mathcal{D}|\psi\rangle$ ▷ Diffusion.
end for
$\hat{x} \leftarrow MEASUREMENT(|\psi\rangle)$
if $f(\hat{x}) = 0$ **then** ▷ If $\hat{x}$ is not solution, then there is no solution.
  $\hat{x} \leftarrow -1$
end if
return $\hat{x}$

A' has a query complexity of $height(A') = O\left(\sqrt{(R-L)/t}\right) = O\left(\sqrt{N/t}\right)$ and an error probability of $O\left(\frac{1}{N}\right)$ for the multiple ones search problem, where $t = |f^{-1}(1)|$.

In a case of an unknown number of solutions, the property below holds.

Property 4[17]**.** In a case of unknown number of solutions t, there is a quantum algorithm A'' with an expected query complexity of $hight(A'') = O\left(\sqrt{(R-L)/t}\right) = O\left(\sqrt{N/t}\right)$ and an error probability of $O\left(\frac{1}{2}\right)$ for the multiple ones search problem, where $t = |f^{-1}(1)|$.

It is known that if the Grover's search algorithm A or its variant has a query complexity of $height(A) = Q(N)$ then $time(A) = O(Q(N) \cdot \log_2 N)$, because the implementation of diffusion operator $\mathcal{D}$ requires $O(\log_2 N)$ gates from a standard basis.

4.2.4 Amplitude amplification

The amplitude amplification algorithm is a generalization of Grover's search algorithm[15]. There are two generalizations: changing the diffusion operator and changing the query operator. For the first generalization, assuming that we have a quantum algorithm with measurement in the end for the single one search or multiple ones search problems. This quantum algorithm can be represented by a unitary matrix $\mathcal{A}$. If the algorithm has the success probability p, then we can replace the diffusion operator by $\mathcal{A}R\mathcal{A}^{-1}$ and do $1/\sqrt{p}$ steps. We then obtain an algorithm that finds a solution with high probability.

Property 5[15]**.** Assuming that there is a quantum algorithm $\mathcal{A}$ for the single one search problem with measurement only in the end and a small success probability p. The amplitude amplification algorithm which extends Grover's search algorithm by replacing $\mathcal{D}$ by $\mathcal{A}R\mathcal{A}^{-1}$ has a query complexity of $height(A) = O\left(height(\mathcal{A}) \cdot \frac{1}{\sqrt{p}}\right)$ and an error probability of $O(p)$.

A similar result can be obtained for the multiple ones search problem.

For the second generalization, supposing that we have a complex function f with a classical or a quantum algorithm with measurement only in the end, such that its running time is $T(N)$. We can then implement it in Oracle and the running time of the one's search will be $Q(N) \cdot T(N)$, where $Q(N)$ is the query complexity of the corresponding version of Grover's search:

- $O(\sqrt{N} \cdot T(N))$ in a case of single one search;
- $O(\sqrt{N/t} \cdot T(N))$ in a case of t ones search problem;
- $O\left(height(\mathcal{A}) \cdot \frac{1}{\sqrt{p}}\right)$ in a case of amplitude amplification.

4.2.5 Equal amplitudes preparation

We can prepare $\frac{1}{m}\sum_{j=1}^{m}|j\rangle$ using the following subroutine. Supposing that we have a quantum register $|\psi\rangle$ of m quantum states ($\lceil \log_2(m) \rceil$ qubits), a quantum register $|\phi\rangle$ of m quantum states ($\lceil \log_2(m) \rceil$ qubits), a qubit $|\xi\rangle$, and a qubit $|\zeta\rangle$. Setting $|\phi\rangle = |m\rangle$, we then have the following quantum transformation *CMP*:

$$\begin{cases} CMP|i\rangle|m\rangle|0\rangle \to |i\rangle|m\rangle|0\rangle, \text{ for } i \leqslant m; \\ CMP|i\rangle|m\rangle|0\rangle \to |i\rangle|m\rangle|1\rangle, \text{ for } i > m. \end{cases}$$

Algorithm 5 shows the procedure.

4.2.6 Amplitude estimation algorithm

If we do not know the number of ones for function f, we can estimate the probability of success via measurement after several iterations; this approach is called amplitude estimation.

Property 6. For any positive integers k and S, the amplitude estimation algorithm[15] outputs $\tilde{a}$ ($0 \leqslant \tilde{a} \leqslant 1$) such that

$$|\tilde{a} - a| \leqslant 2\pi k \frac{\sqrt{a(1-a)}}{S} + \left(\frac{\pi k}{S}\right)^2$$

with probability of at least $\frac{8}{\pi^2}$ when $k = 1$ and with probability greater than $1 - \frac{1}{2(k-1)}$ for $k \geqslant 2$. This uses exactly S iterations of Grover's algorithm. If $a = 0$ then $\tilde{a} = 0$ with certainty, and if $a = 1$ and S is even, then $\tilde{a} = 1$ with certainty.

The algorithm is based on Fourier analysis[18], like the Shor algorithm[19,20]. This approach can also be

Algorithm 5 **Equal_ampls_base** (m, $|\psi\rangle$)

$|\psi\rangle \leftarrow |0^{\otimes \lceil \log_2(m) \rceil}\rangle$ ▷ Initial zero value
$|\psi\rangle \leftarrow H^{\otimes \lceil \log_2(m) \rceil}|\psi\rangle$
$|\phi\rangle \leftarrow |m\rangle$
$|\zeta\rangle \leftarrow |0\rangle$
$|\xi\rangle \leftarrow |0\rangle$
$|\zeta\rangle|\psi\rangle \leftarrow CNOT|\zeta\rangle|\psi\rangle$
$|\psi\rangle|\phi\rangle|\xi\rangle \leftarrow CMP|\psi\rangle|\phi\rangle|\xi\rangle$
$MEASUREMENT(|\xi\rangle, |\zeta\rangle)$
return $|\psi\rangle$

used for eigenvalue estimation[21–23].

As the first and the last step of the algorithm, we apply the quantum Fourier transform operator F_S:

$$F_S : |x\rangle \to \frac{1}{\sqrt{S}} \sum_{y=0}^{S-1} \mathrm{e}^{2\pi \mathrm{i} xy/S} |y\rangle, \quad 0 \leqslant x < S.$$

In the algorithm, we use two quantum registers: the register $|\psi\rangle$ with N quantum states ($\lceil \log_2 N \rceil$ qubits) and the register $|\phi\rangle$ with S quantum states ($\lceil \log_2 S \rceil$ qubits).

Supposing we have an integer $i \in \{1, \ldots, S\}$ and any quantum operator U that can be applied to the register $|\psi\rangle$. Consider the following operator Λ_U^i that can be applied to the register $|\phi\rangle|\psi\rangle$:

$$\Lambda_U^i : |j\rangle|\psi\rangle \to |j\rangle(U|\psi\rangle), \text{ if } j \geqslant i-1;$$
$$\Lambda_U^i : |j\rangle|\psi\rangle \to |j\rangle|\psi\rangle, \text{ if } j < i-1.$$

We will use transformation $\Lambda_{\mathcal{O}}^i$ for a query and transformation $\Lambda_{\mathcal{D}}^i$ for a diffusion. Algorithm 6 shows the procedure.

One of the applications of the amplitude estimation algorithm is the estimation of $|f^{-1}(1)|$[15]; i.e., the number of arguments for a function f that have a 1-result. The idea is to find the probability of $\tilde{a}$ and then $|f^{-1}(1)| \approx \tilde{a}/N$.

The traditional approach for amplitude estimation is not reversible. However, sometimes we need reversiblity; for example, applying the amplitude amplification algorithm and getting a result with high probability. A reversible algorithm[24] uses a coherent form of majority voting to obtain a reversible analog for algorithms like amplitude estimation. The property of this algorithm is presented in Property 7.

Property 7. Let $\mathcal{A}$ be a unitary operation that maps $|0^{\otimes n}\rangle \to \sqrt{a}|y\rangle + \sqrt{1-|a|}|y^{\perp}\rangle$ for $1/2 < |a_0| \leqslant |a| \leqslant 1$ using Q queries. There exists an algorithm, such that for any $\Delta > 0$, there exists an integer k and a state $|\psi\rangle$ that can be produced and obeys $\||\psi\rangle - |0^{\otimes nk}\rangle|y\rangle\|_2 \leqslant \sqrt{2\Delta}$ using a number of queries bounded above by

$$2Q \left\lceil \frac{\ln(1/\Delta)}{2\left(|a_0| - \frac{1}{2}\right)^2} \right\rceil.$$

The basic idea of the algorithm is to prepare k copies of the state $\sqrt{a}|y\rangle + \sqrt{1-|a|}|y^{\perp}\rangle$, coherently compute the median via a reversible circuit, and uncompute the k resource states to find the median of the values of y.

Algorithm 6 Amplest (N, S, f)

$|\psi\rangle \leftarrow |0\rangle$ ▷ The initial value
$|\phi\rangle \leftarrow |0\rangle$ ▷ The initial value
$|\psi\rangle \leftarrow H^{\otimes \log_2 N}|\psi\rangle$ ▷ Initialization of the Grover's search
$|\phi\rangle \leftarrow F_S|\phi\rangle$ ▷ quantum Fourier transform
for $i \in \{1, \ldots, S\}$ **do**
  $|\phi\rangle|\psi\rangle \leftarrow \Lambda_{\mathcal{O}}^i|\phi\rangle|\psi\rangle$ ▷ Query.
  $|\phi\rangle|\psi\rangle \leftarrow \Lambda_{\mathcal{D}}^i|\phi\rangle|\psi\rangle$ ▷ Diffusion.
end for
$|\phi\rangle \leftarrow F_S^{-1}|\phi\rangle$ ▷ quantum Fourier transform
$x \leftarrow MEASUREMENT(|\phi\rangle)$
$\tilde{a} \leftarrow \left(\sin\left(\frac{\pi x}{S}\right)\right)^2$
return $\tilde{a}$

4.2.7 Quantum "maximum search" algorithm

"Maximum" problem. Given a function $f : [N] \to [D]$, find the $x \in \{L, \ldots, R\} \subset [N]$, such that $f(x) = \max_{y \in \{L, \cdots, R\}} f(y)$, for some integer $D > 0$. The function f is computed by Oracle.

There is no better classical algorithm than brute force. The time and query complexity of the classical algorithm is $O(N)$.

Durr and Høyer[25] developed a quantum algorithm based on the Grover search algorithm from Section 4.2.3. The expected query complexity of the algorithm is $O(\sqrt{N})$. Grover_max (L, R, f) searches the index of $x \in \{L, \ldots, R\}$, which is the maximal element. Algorithm 7 shows the procedure.

Let Grover (l, r, f) be a quantum subroutine that returns one of $x \in \{l, \ldots, r\}$, such that $f(x) = 1$ with equal probability where $l, r \in [N]$. If there is no such x, then the subroutine returns -1. This subroutine was discussed in Section 4.2.3. Let $h_m : [N] \to \{0, 1\}$ be the function such that $h_m(x) = f(x) > f(m)$, for $m \in [N]$.

Durr and Høyer[25] proved that the expected query complexity of the algorithm is $O(\sqrt{R-L})$ and the error probability is 0.5 at most. The "minimum" problem can be defined similarly and the algorithm is the same, but with $h_m(x) = f(x) < f(m)$.

Algorithm 7 Grover_max (L, R, f)

$m \leftarrow L$ ▷ We assume that m is the answer
$m' \leftarrow m$
while $m \neq -1$ **do**
  $m' \leftarrow m$ ▷ We store m in m' and try to improve the answer.
  $m \leftarrow$ Grover$(L, R, h_{m'})$ ▷ We search index of the element that is smaller than $f(m')$
end while
return m' ▷ No element is greater than $f(m')$.

4.3 SWAP-test

The SWAP-test is the known quantum procedure for the equality test of two unknown quantum states $|\psi\rangle$ and $|\psi'\rangle$. The SWAP-test is used in different areas, including quantum cryptography[26, 27] and the quantum nearest neighbors algorithm.

"Standard" SWAP-test. The SWAP-test procedure is described by the following quantum circuit (see Fig. 11).

$Pr_{\text{swap}}[|\psi\rangle = |\psi'\rangle]$ is the probability that the SWAP-test has quantum states $|\psi\rangle$ and $|\psi'\rangle$ output the result "$|\psi\rangle = |\psi'\rangle$".

Property 8. For any two different states $|\psi\rangle$ and $|\psi'\rangle$, it is true that

$$Pr_{\text{swap}}[|\psi\rangle = |\psi'\rangle] = \frac{1}{2}\left(1 + |\langle\psi|\psi'\rangle|^2\right).$$

4.4 Distance computing algorithms

Different quantum subroutines can be used for computing the Euclidean distance between two n-dimensional vectors. These are used, for example, in the nearest-neighbor algorithm. Below are some assumptions.

(1) The input vectors $x^1, \ldots, x^m$ and x are d-sparse for some $1 \leqslant d \leqslant n$.

(2) Quantum oracles are provided in the form

$$\mathcal{O}|i\rangle|j\rangle|0\rangle \to |i\rangle|j\rangle|x_j^i\rangle,$$
$$\mathcal{F}|i\rangle|l\rangle \to |i\rangle|pos(i,l)\rangle,$$

where $pos(i,l)$ is the position of l-th non-zero element of x^i.

(3) There is r_{max} such that $0 \leqslant x_j^i \leqslant r_{max}$ for any $i \in \{1,\ldots,m\}$, $j \in \{1,\ldots,n\}$.

(4) Each vector is normalized to 1, for convenience.

(5) The running time of the algorithm is dominated by the number of queries made to oracles $\mathcal{O}$ and $\mathcal{F}$.

4.4.1 Inner product method

Property 9[24]. Let $v_1, \ldots, v_m \in \mathbb{C}^n$ be d-sparse unit vectors such that

$$\max_{j\in\{1,\ldots,m\},\, i\in\{1,\ldots,n\}} |v_{j,i}| \leqslant r_{max}, u \in \mathbb{C}^n.$$

Then the task of finding $\max_{j\in\{1,\ldots,m\}} |\langle u|v_j\rangle|^2$ with error of at most ε and with success probability at least $1-\delta_0$ requires an expected number of combined queries to $\mathcal{O}$ and $\mathcal{F}$ that is bounded above by

$$O\left(\frac{(\sqrt{m}\log m)\cdot d^2\cdot r_{max}^4}{\varepsilon}\right).$$

If d and r_{max} are small enough, we have quadratic speed-up comparing to the classical algorithm. Additionally, if $r_{max} = \Theta(1/\sqrt{d})$, the running time does not depend on d or r_{max}.

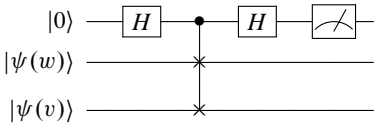


Fig. 11 Quantum circuit.

Before presenting the algorithm, we introduce some unitary transformations and subroutines. We combine the results of majority voting for the amplitude estimation and maximum finding algorithms to obtain the following result.

Property 10. Assuming that for any $j = 1, \ldots, m$ and integers $y_1, \ldots, y_m$, a unitary transformation $|j\rangle|0^{\otimes n}\rangle \to |j\rangle\left(\sqrt{a}|y_j\rangle + \sqrt{1-|a|}|y_j^{\perp}\rangle\right)$ for $1/2 < |a_0| \leqslant |a| \leqslant 1$ can be performed using Q queries, then the expected number of queries made to find $\min_j y_j$ with failure probability of at most δ_0 is bounded above by

$$O\left(\sqrt{m}Q\frac{\log M - \log\delta_0}{|a_0|^2}\right)$$

We can efficiently prepare the sates that store the inner products for vectors and then apply the algorithm from Property 10.

Property 11. For any fixed $\varepsilon > 0$ and any pair of d-sparse unit vectors $u \in \mathbb{C}^n$ and $v_j \in \mathbb{C}^n$ (where $j \in \{1,\ldots,m\}$, $r_{0,max} \geqslant \max_{i\in\{1,\cdots,n\}} u_i$ and $r_{j,max} \geqslant \max_{i\in\{1,\cdots,n\}} v_{j,i}$), a state of the form $\sqrt{A}|\psi\rangle|y\rangle + \sqrt{1-|A|}|\psi\rangle|y^{\perp}\rangle$ can be efficiently prepared where y encodes $|\langle u|v_j\rangle|^2$ within error ε and $|A| \geqslant 8/\pi^2$ using a number of queries

$$Q \leqslant 12\left\lceil\frac{4\pi(\pi+1)d^2 r_{0,max}^2 r_{j,max}^2}{\varepsilon}\right\rceil.$$

4.4.2 Other methods

There is another way to compute the distances between vectors[24], based on computing Euclidean distances rather than inner products. This alternative method gives approximately the same speed-up.

4.5 Implementation of the algorithms

There are a range of quantum Software Development Kits (SDKs) and languages, some of which already contain implementations of the algorithms that we have covered in this paper (see Table 1).

Languages and SDKs:

Table 1 Algorithms implemented in SDKs.

Library	GS	AA	AE	MAX
Project Q	+	−	−	−
Qiskit (Aqua)	+	−	+	−
Rigetti forest SDK	+	+	+	−
Microsoft Q#	+	+	+	−
Quipper	−	−	−	−

- Project Q[28];
- Qiskit[29];
- Rigetti forest SDK[30];
- Microsoft Q#[31];
- Quipper[32].

Algorithms:

- Grover's Search (GS)—In some languages, Grover's search is considered as a partial case of amplitude amplification;
- Amplitude Amplification (AA);
- Amplitude Estimation (AE);
- Maximum Search Algorithm (MAX).

5 Conclusion

This paper presents preliminary knowledge of quantum computing, including qubits, quantum registers, quantum states, basic transformations, quantum circuits, and information extraction. The fundamental quantum tools for quantum classification algorithms are also discussed.

Acknowledgment

This work was supported in part by the Russian Science Foundation (No. 19-19-00656) and Natural Science Foundation of Guangdong Province, China (No. 2019A1515011721).

References

[1] S. Arunachalam and R. de Wolf, Guest column: A survey of quantum learning theory, *ACM SIGACT News*, vol. 48, no. 2, pp. 41–67, 2017.

[2] D. Kopczyk, Quantum machine learning for data scientists, arXiv preprint arXiv: 1804.10068, 2018.

[3] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[4] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, *Quantum Information Processing*, vol. 13, no. 11, pp. 2567–2586, 2014.

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge University Press, 2000.

[6] R. Jozsa. Quantum computation prerequisite material, https:// www.semanticscholar.org / paper / QUANTUM-COMPUTATION-PREREQUISITE-MATERIAL-Jozsa/ 6dc618f0041a20d8c722ed2cdd8c4057d6e92a0b, 2013.

[7] A. Eremenko, Spectral theorems for hermitian and unitary matrices, Technical report, Purdue University, USA, 2017.

[8] I. Wegener, *Branching Programs and Binary Decision Diagrams: Theory and Applications*. Philadelphia, PA, USA: SIAM, 2000.

[9] F. Ablayev, A. Gainutdinova, and M. Karpinski, On computational power of quantum branching programs, in *Fundamentals of Computation Theory*. *FCT 2001*, R. Freivalds, ed. Berlin, Germany: Springer, 2001, pp. 59–70.

[10] F. M. Ablayev and A. V. Vasilyev, On quantum realisation of Boolean functions by the fingerprinting technique, *Discrete Mathematics and Applications*, vol. 19, no. 6, pp. 555–572, 2009.

[11] F. Ablayev, M. Ablayev, K. Khadiev, and A. Vasiliev, Classical and quantum computations with restricted memory, in *Adventures Between Lower Bounds and Higher Altitudes*, 2018, pp. 129–155.

[12] C. A. Trugenberger, Probabilistic quantum memories, *Physical Review Letters*, vol. 87, no. 6, p. 067901, 2001.

[13] P. Kaye, Reversible addition circuit using one ancillary bit with application to quantum computing, arXiv preprint arXiv: quant-ph/0408173, 2004.

[14] L. K. Grover, A fast quantum mechanical algorithm for database search, in *ACM Symp. on Theory of Computing*, Philadelphia, PA, USA, 1996, pp. 212–219.

[15] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemporary Mathematics*, vol. 305, pp. 53–74, 2002.

[16] L. K. Grover and J. Radhakrishnan, Is partial quantum search of a database any easier, in *ACM Symp. on Parallelism in Algorithms and Architectures*, Las Vegas, NV, USA, 2005, pp. 186–194.

[17] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, *Fortschritte der Physik*, vol. 46, nos. 4&5, pp. 493–505, 1998.

[18] G. Brassard, P. Høyer, and A. Tapp, Quantum counting, in *International Colloquium on Automata, Languages, and Programming*, K. G. Larsen, S. Skyum, and G. Winskel, eds. Berlin, Germany: Springer, 1998, pp. 820–831.

[19] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[20] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[21] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem, arXiv preprint arXiv: quant-ph/ 9511026, 1995.

[22] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Quantum algorithms revisited, in *Proceedings of the Royal Society of London. Series A*: *Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 339–354, 1998.

[23] M. Mosca, Counting by quantum eigenvalue estimation, *Theoretical Computer Science*, vol. 264, no. 1, pp. 139–153, 2001.

[24] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning, *Quantum Information & Computation*, vol. 15, nos. 3&4, pp. 316–356, 2015.

[25] C. Durr and P. Høyer, A quantum algorithm for finding the minimum, arXiv preprint arXiv: quant-ph/9607014, 1996.

[26] D. Gottesman and I. Chuang, Quantum digital signatures, arXiv preprint arXiv: quant-ph/0105032, 2001.

[27] F. M. Ablayev and A. V. Vasiliev, Cryptographic quantum hashing, *Laser Physics Letters*, vol. 11, no. 2, p. 025202, 2014.

[28] Project Q, https://projectq.ch/, 2019.

[29] Qiskit, https://qiskit.org/, 2019.

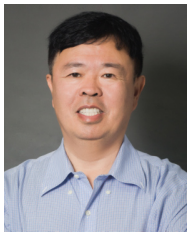
[30] Rigetti forest sdk, https://pyquil.readthedocs.io/en/stable/start.html, 2019.

[31] Microsoft Q#, https://docs.microsoft.com/enus/quantum/?view=qsharp-preview, 2019.

[32] The quipper language, https://www.mathstat.dal.ca/selinger/quipper/, 2019.

Farid Ablayev received the habilitation degree (doctor of physics and mathematics-second level after the PhD level) at Moscow State University. He is a professor in Kazan Federal University and Kazan E. K. Zavoisky Physical-Technical Institute. His current research interests include complexity theory, quantum computing, automata theory, machine learning, and data stream processing algorithms.

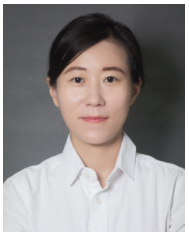
Marat Ablayev received the master degree from Kazan Federal University in 2005. He is a researcher in Kazan Federal University and Kazan E. K. Zavoisky Physical-Technical Institute. His current research interests include complexity theory, quantum computing, automata theory, machine learning, and data stream processing algorithms.



Joshua Zhexue Huang received the PhD degree from the Royal Institute of Technology, Sweden. He is now a professor in the College of Computer Science and Software, Shenzhen University, and professor and chief scientist in the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, and honorary professor in the Department of Mathematics, the University of Hong Kong. His research interests include data mining, machine learning, and clustering algorithms.

Kamil Khadiev received the PhD degree from Kazan Federal University in 2015. He worked in Institute of Informatics of Tatarstan Academy of Science, University of Latvia, Smart Quantum Technologies Ltd, Kazan E. K. Zavoisky Physical-Technical Institute, and Kazan Federal University. His current research interests include quantum computing, quantum algorithms, communicational complexity, automata theory, branching programs, machine learning, and data stream processing algorithms.

Nailya Salikhova is a PhD student in Kazan Federal University. Her current research interests include quantum computing, machine learning, and data stream processing algorithms.



Dingming Wu received the PhD degree in computer science in 2011 from Aalborg University, Denmark. She is an assistant professor with College of Computer Science & Software Engineering, Shenzhen University, China. Her general research area is data management and mining, including data modeling, database design, and query languages, efficient query and update processing, indexing, and mining algorithms.