Int. J. Appl. Math. Comput. Sci., 2019, Vol. 29, No. 1, 139–150
DOI: 10.2478/amcs-2019-0011

RECOMMENDATION SYSTEMS WITH THE QUANTUM K–NN AND GROVER ALGORITHMS FOR DATA PROCESSING

MAREK SAWERWAIN [a,*], MAREK WRÓBLEWSKI [a]

[a] Institute of Control and Computation Engineering
University of Zielona Góra, Licealna 9, 65-417 Zielona Góra, Poland
e-mail: {M.Sawerwain,M.Wroblewski}@issi.uz.zgora.pl

In this article, we discuss the implementation of a quantum recommendation system that uses a quantum variant of the k-nearest neighbours algorithm and the Grover algorithm to search for a specific element in an unstructured database. In addition to the presentation of the recommendation system as an algorithm, the article also shows the main steps in construction of a suitable quantum circuit for realisation of a given recommendation system. The computational complexity of individual calculation steps in the recommendation system is also indicated. The verification of the correctness of the proposed system is analysed as well, indicating an algebraic equation describing the probability of success of the recommendation. The article also shows numerical examples presenting the behaviour of the recommendation system for two selected cases.

Keywords: quantum k-NN algorithm, recommendation systems, Grover algorithm, big data.

1. Introduction

The development of quantum computational methods (Nielsen and Chuang, 2010; Galindo and Martin-Delgado, 2002) allows their use in areas such as machine learning (Wiebe *et al.*, 2015), decisions models (Busemeyer and Bruza, 2012) or big data (Nielsen, 2016), as well as in classification tasks (Sergioli *et al.*, 2017; Santucci, 2017; Montanaro, 2017). The classical methods of analysing data sets of big data are widely used, but the use of quantum computers that allow the processing of exponential amounts of data seems to be extremely important in this area (Stefanowski *et al.*, 2017; Veloso *et al.*, 2015). In addition, due to the features of quantum information, the lack of the possibility of cloning information and direct comparison of two quantum registers, the construction of such a system requires a slightly different approach. In this article we show that the recommendation process can be based on the method of k-nearest neighbours classification (Hechenbichler and Schliep, 2004). Such an approach allows us to create a system where the effectiveness of the recommendation can be very high.

In the article, we discuss the construction of a recommendation system based on two quantum computing algorithms. The first one is the quantum algorithm of k-nearest neighbours (termed quantum k-NN) (Pinkse *et al.*, 2013; Schuld *et al.*, 2014; Wiebe *et al.*, 2015), based on the method presented by Trugenberger (2002). In this algorithm, we use the Hamming distance to classify the elements to be recommended (Wiśniewska and Sawerwain, 2018). Grover's algorithm (Grover, 1996; Arikan, 2003; Galindo and Martin-Delgado, 2000) is used to improve the quality of recommendations. The use of both the methods allows the construction of a recommendation algorithm which is characterized in the recommendation process by better computational complexity than classical approaches (Alpaydin, 2004; Armbrust *et al.*, 2010). However, it should be emphasized that the described solution, like other quantum algorithms (Shor, 1999), is probabilistic, although the probability of indicating recommended elements is very high.

The solution proposed in this work can also be based on the classifiers proposed by Santucci (2017), Sergioli *et al.* (2018) and Chakrabarty *et al.* (2017). However, the use of the k-nearest neighbours algorithm and the Grover algorithm, due to the relative simplicity of circuit design for these algorithms allows us to expect that their physical

[*] Corresponding author

implementation of Q Experience (IBM, 2018) is possible at the moment. In addition, the proposed method, due to its direct translation into the language of quantum gates, allows providing its exact computational complexity in the process of recommendation. Additionally, owing to the use of the Grover algorithm, it is possible to achieve a high probability of indicating the right elements. In the further part of the work, the value of this probability is described by the value $P_{\max}$.

It should be noted that this article is an extension of a previous publication (Sawerwain and Wróblewski, 2019). The current paper presents more precisely the structure of the quantum register and numerical examples showing the behaviour of the recommending system. The computational complexity of individual steps implemented in the recommendation process was also indicated.

The organization of the article is as follows: In Section 2, selected issues of quantum computing, such as quantum register and superpositions, are presented. These concepts characteristic for quantum computing are directly applied in the discussed recommendation system. In Section 3, we discuss the construction of the recommendation system, present the algorithm of conduct and discuss the construction of the quantum circuit which implements the discussed solution. The correctness of the recommendation system, i.e., the description of the probability of indicating the correct element (or elements), is presented in Section 3.4. Numerical experiments are also implemented and discussed in Section 4. The experiment shows the behaviour of the system in two cases, when one and two elements are recommended. The summary of the article is given in Section 5, where the final conclusions and some comments about possible further work on the method described in this article are given.

2. Brief introduction to quantum information processing

The concept of a classic bit, which is the basic unit of information, can be extended to the qubit for quantum systems (Walther *et al.*, 2005; Steane, 1998). For this purpose, we consider a two-dimensional Hilbert space $\mathcal{H}_2$ and indicate its orthonormal basis:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1}$$

In the last equation, Dirac's notation is used. The vector $|0\rangle$ usually means zero. The natural thing is that $|1\rangle$ represents one. The given 1 base is also called a standard computation base.

In quantum information, the notion of a qubit is the equivalent of the classical bit. The qubit state is represented by a vector in the two-dimensional Hilbert space $\mathcal{H}$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{2}$$

This vector is normalized, so that $|\alpha|^2 + |\beta|^2 = 1$, where $\alpha, \beta \in \mathbb{C}$, the set of complex numbers. The presented state $|\psi\rangle$ is called a vector state or a pure state.

Following the classic computer science, concatenation of multiple classical bits creates a register. The $|\psi\rangle$ register with n qubits is built using the tensor product

$$|\psi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_{n-1}\rangle, \tag{3}$$

where $\otimes$ is the tensor product operation.

Remark 1. (*Quantum entanglement*) It should be added that there are cases where the register cannot be written in the form of a tensor product. This state of the register is called an entangled state.

It should also be noted that, in addition to the description in the form of pure states, the so-called unknown density matrices (the pure state of qubit $|\psi\rangle$) taking the following form are also used:

$$\rho = |\psi\rangle\langle\psi| = \begin{bmatrix} \alpha^2 & \alpha\beta \\ \alpha\beta & \beta^2 \end{bmatrix}, \tag{4}$$

where the vector $\langle\psi|$ signifies the transpose of $|\psi\rangle$.

In general, the representation for a mixed state for which only pure states are included, $|\psi_i\rangle$, takes the following form:

$$\rho = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|, \tag{5}$$

where λ_i means the probability of the state $|\psi_i\rangle$ and $\sum_i \lambda_i = 1$.

Remark 2. (*Exponential capacity of a quantum register*) One of the main differences between the classic and quantum registers is the exponential capacity of the latter. The amount of classical information contained in the quantum register described by the state vector can be described by 2^n, where n is the number of qubits, which makes classical simulation of the quantum register ineffectual using classical computations methods. There are, of course, special cases such as the so-called CHP circuits (Aaronson and Gottesman, 2004), but in general the quantum register requires exponential computational resources to simulate the operation of the quantum register using a classical machine. This situation is much worse when we using a density matrix, as their size is $d^n \times d^n$.

The state vector for 16 qubits needs 512 kB of memory, and 256 MB for a density matrix, assuming that we use double precision floating point numbers describing individual amplitudes of probability. However, doubling the number of qubits to 32 due to the exponential capacity of the quantum register already requires 32 GB of available memory, and 1 ZB (zetta bytes) for the density matrix.

Several additional operations can be performed on the quantum register. As part of this brief introduction, we will only quote the most important examples of operations, i.e., unitary and measurement ones.

The realization of a unitary operation of U for the quantum state given by the vector is represented by the equation

$$U|\psi_0\rangle = |\psi_1\rangle. \tag{6}$$

For a density matrix, the unitary operation U is described by the following relationship:

$$U\rho_0 U^\dagger = \rho_1. \tag{7}$$

A very important thing is the method of creating a unitary operation for the quantum register. In the case of state modifications for the first and the third qubit, the unitary unit construction takes the following form:

$$U = u_1 \otimes I \otimes u_2. \tag{8}$$

The unitary operation is reversible (also called an uncompute one), i.e., for the operation U you can always enter the operation $U^\dagger$, where the symbol $\dagger$ represents the Hermitian adjoint operation. If $U = U^\dagger$, then U is called as a self-adjoint operation.

The basic set of unit operations includes the so-called Pauli gates (operators): X, Y, Z. Their matrix representation is

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{9}$$

The Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{10}$$

is also very important because it is used to introduce the so-called superposition for quantum states. For example, let $|\Xi\rangle$ be an n-qubit state

$$|\Xi\rangle = |\xi_0\rangle \otimes |\xi_1\rangle \otimes \cdots \otimes |\xi_{n-2}\rangle \otimes |\xi_{n-1}\rangle. \tag{11}$$

Using the $|\Xi\rangle$ register, you can easily show how the use of a single Hadamard gate works:

$$\begin{aligned} H|\Xi\rangle &= H(|\xi_0\rangle \otimes |\xi_1\rangle \otimes \cdots \otimes |\xi_{n-2}\rangle \otimes |\xi_{n-1}\rangle) \\ &= H|\xi_0\rangle \otimes H|\xi_1\rangle \otimes \cdots \otimes H|\xi_{n-2}\rangle \otimes H|\xi_{n-1}\rangle \\ &= \bigotimes_{i=0}^{n-1} H|\xi_i\rangle = \frac{1}{\sqrt{2^n}} \left[\bigotimes_{i=0}^{n-1} \big(|0\rangle + (-1)^{\xi_i}|1\rangle\big) \right]. \end{aligned} \tag{12}$$

The use of the Hadamard gateway results in the amplitude values being equal to the absolute value $1/\sqrt{2^n}$ for all the states taken by the specified quantum register.

We should be also pay attention to the construction of controlled gates, whose construction requires, in addition to the tensor product, the use of a direct sum of matrices and also uses the projection matrix.

An example may be one of the possible implementations of the CNOT gate (controlled negation gate) for qubits:

$$U = |000\rangle\langle 000|I + |010\rangle\langle 010|X + |111\rangle\langle 111|I. \tag{13}$$

This gate performs a qubit negation if the so-called control qubit takes the state one.

The second type of the basic operation is the so-called measurement operation. We will present only one example of this type of operation: a von Neumann type measurement. It begins with the preparation of observables. The operator is presented in the following way:

$$M = \sum_i \lambda_i P_i, \tag{14}$$

where P_i is a projector onto the operator's eigenspace M with the eigenvalue λ_i.

The results of the measurement performed represent the eigenvalues λ_i, where the individual results occur with the probability

$$P(\lambda_i) = \langle\psi|P_i|\psi\rangle. \tag{15}$$

The obtained result λ_i indicates that the register $|\psi\rangle$ has been transformed to

$$|\psi^{'}\rangle = \frac{P_i|\psi\rangle}{\sqrt{\lambda_i}}, \tag{16}$$

with the probability determined by Eqn. (15).

3. Quantum recommendation system

This section presents the construction of a quantum recommendation system. An example of a database is indicated from which the elements chosen by users will be recommended. The quantum register is also described in Section 3.1. Section 3.2 describes the necessary calculation steps for the implementation of the quantum recommendation algorithm. The construction of a quantum circuit is also shown in Section 3.3. Analysis of the correctness of the proposed algorithm is given in Section 3.4. An analytical description of the probability of success of indicating the recommended element (or many elements) is presented.

3.1. Database and quantum register structure. The proposal for building a quantum recommendation system (QRS) presented in this article is hybrid. The database from which items are recommended based on the user's suggestions and needs is naturally stored in the classical system. An example of a classical database on the basis of which a quantum recommendation system can be built is the IMDB movie database (OMDb, 2018). Figure 1 shows several records selected from the database, with a field describing the feature of the particular movie.

It should be emphasized that there is no need to store the entire database into the quantum system. Usually only two columns of data are relevant: the identifier of recommended elements and the element's feature saved as a binary word. The length of words that are used to describe the identifier and characteristics is important. It is assumed that these will be binary numbers: the identifier will be described with q bits, while the feature with l bits. Usually the entire classic database can be divided into sections referring to elements with common features. Based on this, you can create a quantum register that contains information about the database. A representation of this register and its possible division are shown in Fig. 1. If we use the entire database in this case, we have only one register, or we can divide the whole register into sub-registers with common features, e.g., all historical films would be collected in one register.

In view of the remarks about the form of the database and the structure of the quantum register, we distinguish four main parts:

$$|\Psi\rangle = |\psi_q\rangle \otimes |\psi_{f_f}\rangle \otimes |\psi_{f_u}\rangle \otimes |\psi_k\rangle. \tag{17}$$

The first q qubits of the registry $|\Psi\rangle$, i.e., $|\psi_q\rangle$ are used to encode the identifier of the recommended element. The next part of the $|\psi_{f_f}\rangle$ register describes the features of individual database elements. They refer naturally to the identifier of the individuals items. The l qubits are used to describe the feature, because thanks to the superposition properties we shall create 2^l various features in the database containing 2^q classical elements. The third part of $|\psi_{f_u}\rangle$ is a description of the feature desired by the user. Once again, l qubits are used, although we describe only one feature desired by the user. Again by the superposition principle, applying only l operations (the feature is described by a binary number with a width of l bits), we connect information about user features with the rest of the database. The last part, denoted by the $|\psi_k\rangle$ register, represents additional/auxiliary qubits. More precisely we only need one qubit for the k-NN quantum part and one additional qubit for the Grover amplification. We point out where there qubits are used later in Section 3.3. Section 3.4 presents an analytical analysis, and the success probability of the recommendation is described by $P_{\max}$.

Table 1. Sample structure of the classic database. It should be noted that in the quantum part of the database only two columns are used: identifier ("Id") and description of the feature ("Feature.") The indicated fields are also the main keys within the classical base.

Id	Year	Title	Time	Feature	Language	Rating	Votes	Type	Feature in bits
831	2004	Vendetta: No Conscience, No Mercy	112 min	Crime, Drama, Action	English	4.5	239	movie	110100110000101000
1090	1976	The Big Bus	88 min	Comedy, Action	English	5.6	2165	movie	011100101100000000
1597	1940	Flash Gordon Conquers the Universe	220 min	Action, Sci-Fi, Adventure	English	7.0	819	movie	101100101110100000
2533	1999	Batman Beyond: The Movie	132 min	Animation, Sci-Fi, Action	English	7.8	3201	movie	010110101110101100
2624	1958	Kings Go Forth	109 min	Drama, War, Action	English, French, German	6.6	909	movie	110000110110101000
2977	1987	Innerspace	120 min	Action, Adventure, Comedy	English	6.7	38517	movie	101100100110011000
4540	2012	Aftershock	89 min	Action, Adventure, Horror	Spanish, English	4.8	7531	movie	101100100110111000
5267	2005	The Prophecy: Uprising	88 min	Thriller, Action, Fantasy	English, Romanian	5.4	1603	movie	110010101100000000
5898	2000	Epicenter	102 min	Action, Adventure, Crime	English	3.0	1059	movie	101100100110110000
6334	2002	Escape from Afghanistan	90 min	Action, War	English	3.0	141	movie	101100110110000000
7775	2000	U-571	116 min	Action, War	English, German	6.6	62229	movie	101100110110000000
8364	1973	Yaadon Ki Baaraat	164 min	Drama, Romance, Action	Hindi	7.4	1267	movie	110000100100101000
8430	1994	The Kung Fu Master	45 min	Action	Cantonese	7.5	45	series	101100000000000000
8643	1974	The Arena	83 min	Action, Adventure	English	5.4	650	movie	101100100110000000
9110	2003	The Foreigner	96 min	Action, Thriller	English, Danish, German, Polish	3.2	5272	movie	101100110010000000
9299	2001	American Outlaws	94 min	Action, Western	English	6.0	11659	movie	101100101010000000
12510	1994	I Spy Returns	95 min	Adventure, Comedy, Action	English	5.2	236	movie	100110011100101000

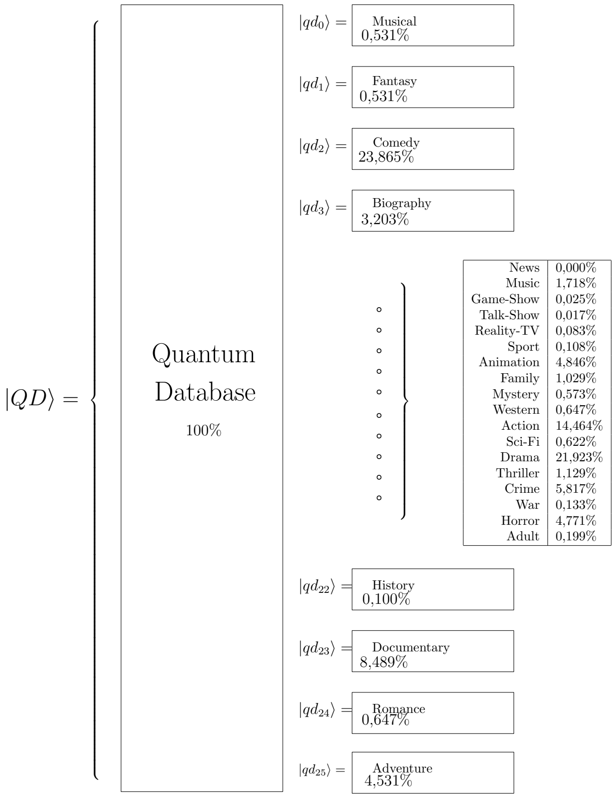


Fig. 1. Partition of the main quantum registry into subregisters in the proposed recommendation system. It is assumed that individual sub-registers contain entries with the same leading feature.

Remark 3. (*Quantity of recommendations*) In the description of the registry structure, we have provided only one feature for the user $|\psi_{f_u}\rangle$; of course, you can add additional user features, e.g., $|\psi_{f_{u_1}}\rangle, |\psi_{f_{u_2}}\rangle, \dots, |\psi_{f_{u_K}}\rangle$, to compute a K-th recommendation. However, in the remainder of this article, for simplicity, we limit our analysis to a single feature.

3.2. Main QRS algorithm.

The proposed approach to the recommendation system is based on two main stages. The first one points the database elements whose features are closest to those indicated by the user. For this purpose, the quantum version of the k-nearest neighbours algorithm is used. Then, to amplify the effectiveness of the recommendation, Grover's algorithm is used. The individual computational steps are presented as Algorithm 1.

The algorithm can also be represented as a control flow in the proposed approach to the recommendation system. The corresponding diagram is shown in Fig. 2. Because the algorithm is based on measurement operations, an additional qubit called c_0 has been introduced in the quantum register. Performing the measurement operation on the qubit c_0 allows us to determine whether the registry has successively been converted to the correct probability distribution. The measurement result c_0 assuming the state $|0\rangle$ means that the quantum register has been transformed into a proper state, and its probability distribution for the next measurement will indicate a recommended element.

Algorithm 1. Quantum recommendation system.

Successive steps in the quantum recommendation system:

(I) Creation of a database; in this step we use $|\psi_q\rangle$ and $|\psi_{f_f}\rangle$ registers.

(II) The user determines which features recommended elements should have; information is represented by the register $|\psi_{f_u}\rangle$.

(III) For a given feature, the appropriate sub-register (or a whole quantum register) is selected representing the relevant part of the database; data are encoded in the state of register $|\psi_{f_f}\rangle$.

(IV) A recommendation process is performed using a quantum algorithm of k-nearest neighbours; two registers are used, $|\psi_{f_u}\rangle$ and $|\psi_{f_f}\rangle$, and we also use one auxiliary qubit from $|\psi_k\rangle$ called $|c_0\rangle$.

(Va) If the state of qubit $|c_0\rangle$ after measurement is $|0\rangle$, then the obtained probability distribution of the recommended elements from Step (IV) can be amplified by Grover's algorithm to improve the probabilistic properties of the best recommended elements; only the state of register $|\psi_{f_u}\rangle$ is amplified. Additionally, we use one auxiliary qubit from $|\psi_k\rangle$ called $|q_A\rangle$. We can jump to Step (VI).

(Vb) If the state of qubit $|c_0\rangle$ after measurement is $|1\rangle$, then we uncompute the q-kNN part and Step (IV) must be repeated.

(VI) Performing the second measurement on the quantum register representing the database. Finally, the recommended element will be obtained (i.e., the element that will have the highest probability of measurement).

3.3. Quantum circuit for QRS and computational complexity.

Figure 3(a) shows the general structure of the quantum circuit implementing the quantum recommendation system. Three sample circuits implementing the most important stages of the recommendation system are also given. The examples are based on a 16-element database shown in Fig. 3(b).

Figure 3(d) shows the database initialization process,

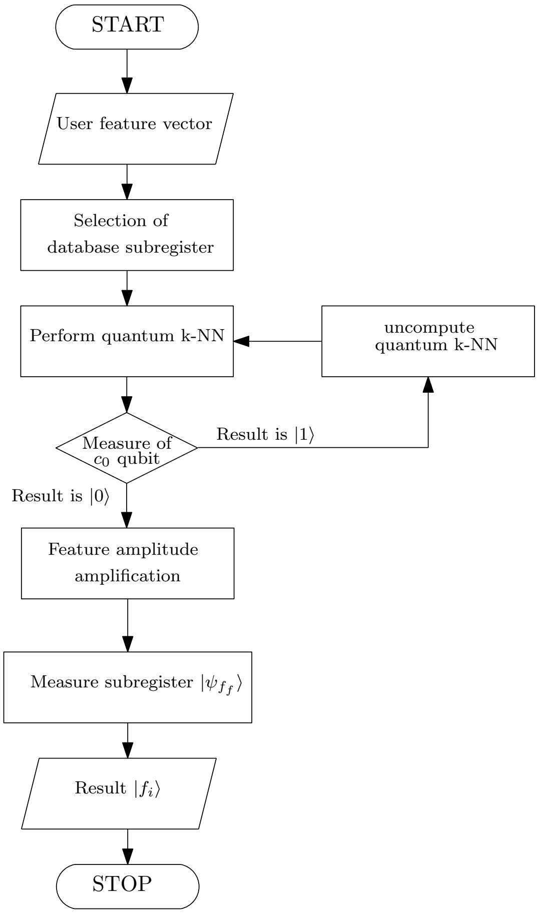


Fig. 2. Control flow of the proposed the quantum recommendation system.

which consists of three stages. The first one creates database identifiers, and for this purpose it applies exactly l Hadamard gates. The next stage is the encoding of features: unfortunately, it requires a larger amount of operation, which depends on the number of elements in the database.

In general, after creating identifiers in the $|\psi_q\rangle$ register, a permutation of elements must be performed in the register $|\psi_{f_f}\rangle$. Permutation generally requires no more than $N(N-1)/2$ single qubits and controlled gates (Li *et al.*, 2013), with $N = 2^q$. Unfortunately, the process of creating a database will not be significantly better than the creation of a classic database, although the structure of feature codes may be helpful in the process of creating a database. This can also be seen in the presented example, where six operators of the controlled negation are enough (the given relation defines the upper limit, therefore for 16 elements, the maximum number of gates is 120) to generate 16 database entries. The third stage, due to the superposition principle, will require only l negation gates to build the user's feature. Computational complexity in the $\mathcal{O}$ notation of this stage can be written as

$$\mathcal{O}_1(l, N) = 2l + N(N-1)/2. \tag{18}$$

However, as shown in the correctness analysis in Section 3.4, you do not need to repeat the process of building the entire database. Circuit (d) presents the process of calculating the Hamming distance and the so-called sum of distances to make a classification. The recommendation process is completed after the measurement has been made by auxiliary qubits c_0 if the result measurement is $|0\rangle$; further details are presented in Section 3.4. The number of operations to be performed when calculating the Hamming distance (d) is linear and depends on the width of the feature, i.e., the number of classic bits l describing the feature:

$$\mathcal{O}_2(l) = 3l + 2. \tag{19}$$

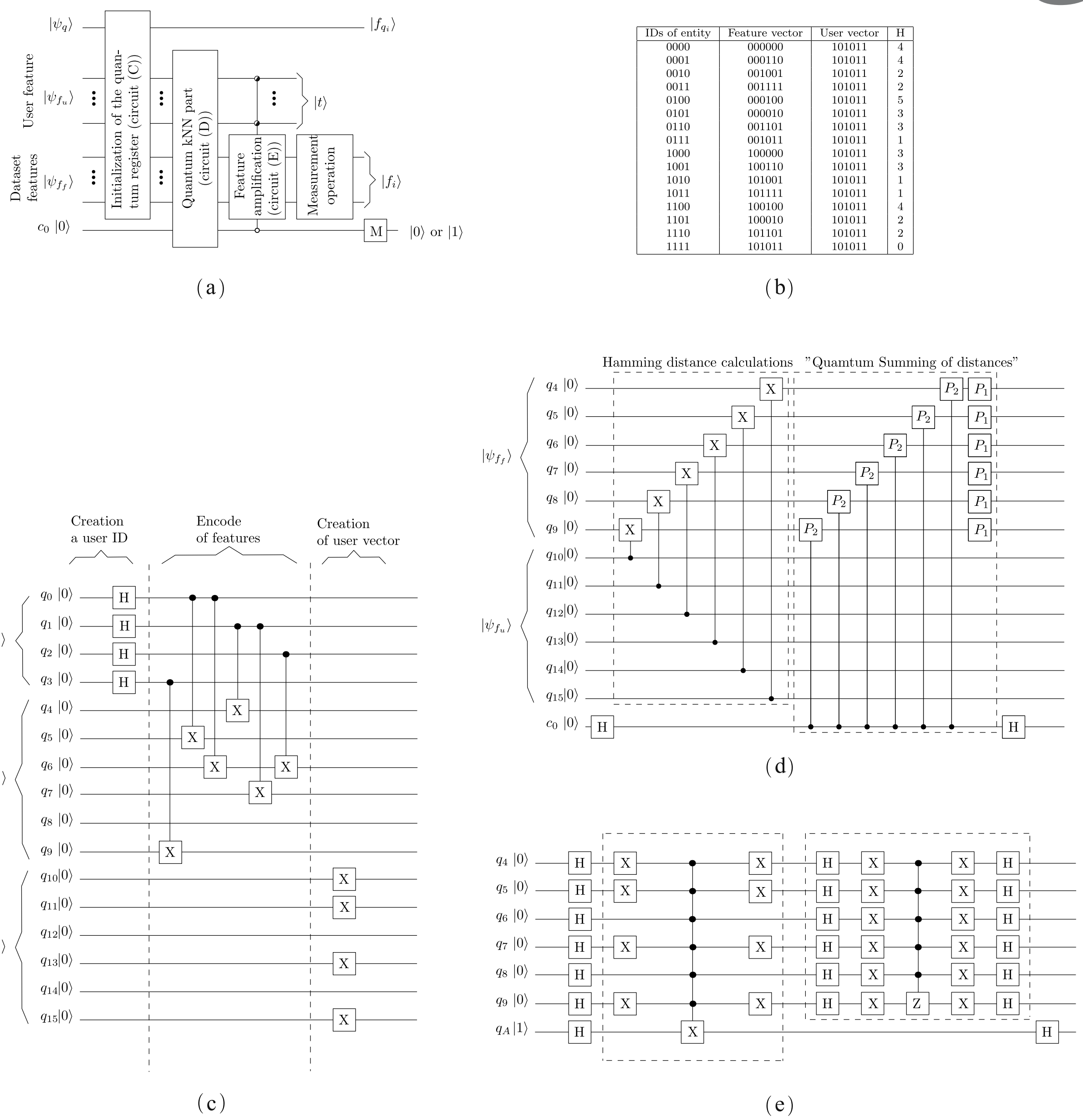
Remark 4. (*Complexity of the quantum k-NN part*) Particular attention should be paid to the fact that the computational complexity of this step depends only on the width of the pattern, unlike in the classical methods, where the number of rows should be taken into account for the analysis of computational complexity.

The construction of the amplifying amplitude circuit for one or several amplitudes with the Grover algorithm depends on the form of the feature described by the user. Figure 3(e) shows the amplification of one amplitude for the sample database shown in Fig. 3(b). The number and arrangement of the negation gates in the oracle section are an exact mapping of the feature specified by the user. The definition of an oracle can be simplified by using only negation on the bits that are encoded, which encode the main feature, e.g., a historical movie can be encoded with the oldest or the youngest bit. However, it can be assumed that the feature requires the use of the largest number of gates, defined as

$$\mathcal{O}_3(l) = 7l + 2c + 3, \tag{20}$$

where l is the width of the feature, and c means the number of gates after decomposing the negation gate in the oracle stage of the Grover algorithm and in the controlled gate Z in the part performing the rotation around the average. Decomposition of controlled gates can be done according to Barenco *et al.* (1995) or Shende and Markov (2009). However, because we have l wide feature then finally we can obtain polynomial complexity (in addition, we still operate on 2^q elements of the classical data).

All computational complexity is naturally dominated by the database creation process, but the calculation process of the recommendation depends only on the width of the feature, and not on the amount of data in the database.



IDs of entity	Feature vector	User vector	H
0000	000000	101011	4
0001	000110	101011	4
0010	001001	101011	2
0011	001111	101011	2
0100	000100	101011	5
0101	000010	101011	3
0110	001101	101011	3
0111	001011	101011	1
1000	100000	101011	3
1001	100110	101011	3
1010	101001	101011	1
1011	101111	101011	1
1100	100100	101011	4
1101	100010	101011	2
1110	101101	101011	2
1111	101011	101011	0

Fig. 3. Quantum circuit for the discussed recommendation system. Part (a) presents three main elements of the recommendation process: preparation of the database, implementation of a quantum algorithm and amplifying the recommended element (with indicated circuits representing individual stages of the recommendation algorithm). Part (b) presents a system of recommended items that were used to construct sample circuits (column H shows the Hamming distance between the element and the example feature 101011 defined by the user). Part (c) represents the initialization part of the database. Part (d) shows an example of implementation of the main part of the quantum k-NN algorithm: Hamming distance calculation and the so-called quantum summing of the distance. Part (e) represents the operation of amplifying amplitudes for the user's feature, i.e., 101011.

3.4. Correctness of the quantum recommendation system. We will begin the analysis of the correctness of the quantum recommendation system by defining the initial state of a quantum register during the first stage, i.e., preparation of the database (for the clarity of our analysis, the qubits describing the record identifier $|\psi_q\rangle$ are omitted):

$$|\psi_0\rangle = |0\rangle^{\otimes 2l+1}. \tag{21}$$

The first l qubits represent features from the database while the next l qubits represent the user feature vector. Initializing the sub-register with the feature database gives the following state:

$$|\Psi_1\rangle = \frac{1}{\sqrt{L}} \sum_{p=1}^{L} |r_1^p, \dots, r_l^p\rangle, \tag{22}$$

where $L = 2^l$, and r_i^p, $i = 1, \ldots, l$, is a bit description of the features of a selected row in the database. After entering the form of the user's feature vector, the state of the quantum register is given as

$$|\Psi_2\rangle = \frac{1}{\sqrt{L}}\left(\sum_{p=1}^{L}|r_1^p, \ldots, r_l^p\rangle\right) \otimes |t_1, \ldots, t_l\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{23}$$

where $|r_i^p\rangle$, $i = 1, \ldots, l$, represents features from the database and $|t_i\rangle$, $i = 1, \ldots, l$, represents the user features.

In the next step, we can calculate the Hamming distance between the user's feature and those stored in the database. Responsible for distance calculation is the quantum circuit in Fig. 3(d), which is described by the following unitary operation U:

$$U = e^{-\mathbf{i}\frac{\pi}{2l}\hat{H}}, \tag{24}$$
$$\hat{H} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}^{\otimes l} \otimes I_{l\times l} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The operation U can be decomposed as the set of single qubit gates

$$P_1 = \begin{bmatrix} e^{-\mathbf{i}\frac{\pi}{2l}} & 0 \\ 0 & 1 \end{bmatrix}, \tag{25}$$

and the controlled gate

$$P_2 = |0\rangle\langle 0| + |1\rangle\langle 1| \otimes {P_1}^{-2}. \tag{26}$$

A further discussion about decomposition of U can be found in the work of Trugenberger (2002). A suitable quantum circuit which implements operation U is depicted as the circuit in Fig. 3(d) as "Quantum summing of distance."

The use of the U operation described by Eqn. (24) results in the state

$$|\psi_4\rangle = \frac{1}{\sqrt{L}}\sum_{p=1}^{L}\Big(e^{\mathbf{i}\frac{\pi}{2l}d(t,r^p)}|d_1^p, \ldots, d_l^p\rangle \otimes|t_1, \ldots, t_l\rangle \otimes |0\rangle + e^{-\mathbf{i}\frac{\pi}{2l}d(t,r^p)}|d_1^p, \ldots, d_l^p\rangle \otimes|t_1, \ldots, t_l\rangle \otimes |1\rangle\Big), \tag{27}$$

where $|d_i^p\rangle$, $i = 1, \ldots, l$, represents the quantum register which possesses information about the distance between a feature from the database and the user feature, and $d(t, r^p)$ represents the value of the Hamming distance between the features. The performing of a Hadamard operation on the additional qubit c_0 allows us to obtain the description of the final state for the stage related to the quantum k-NN algorithm:

$$|\psi_5\rangle = \frac{1}{\sqrt{L}}\sum_{p=1}^{L}\Big(\cos\left(\frac{\pi}{2l}d(t,r^p)\right)|d_1^p, \ldots, d_l^p\rangle \otimes|t_1, \ldots, t_l\rangle \otimes |0\rangle + \sin\left(\frac{\pi}{2l}d(t,r^p)\right)|d_1^p, \ldots, d_l^p\rangle \otimes|t_1, \ldots, t_l\rangle \otimes |1\rangle\Big). \tag{28}$$

The probability of measuring state zero on qubit c_0, i.e., of success in obtaining the desired probability with the correct indication of the recommended elements, is

$$P(c_0) = \frac{1}{L}\sum_{p}\cos^2\left(\frac{\pi}{2l}d(t,r^p)\right). \tag{29}$$

If we have measured the state $|1\rangle$ on qubits c_0, then restoring the state $|\psi_4\rangle$ is necessary. We can do this by performing an inverse operation to U given by Eqn. (24) i.e., $U^\dagger$. It will require the same amount of work as the operation U. This operation restores the state of the registry to the state before the Hamming distance calculation and summing. In this way, we avoid the costly process of rebuilding the database.

If, by simplifying, we denote by P the probability of receiving a particular recommendation, then obtaining a probability distribution with the accuracy of ε requires generally $\mathcal{O}(P \times (1 - P) \times 1/\varepsilon^2)$ repetitions of the implementation of a quantum algorithm of k-nearest neighbours. It should be emphasized, however, that using classical multiprocessor solutions we can use many quantum machines to solve the same task, for example, to obtain linear calculation time of the probability distribution for the exponential amount of data L, described by Eqn. (22).

Remark 5. (*Improving the distribution of the probability of success*) The distribution of the probability obtained is not satisfactory if the user's designated feature determines the choice of one compatible item or a few items from a very close neighbourhood. Therefore, using the Grover algorithm significantly improves the final probability distribution of the obtained recommendation (Brassard and Hoyer, 1997).

Therefore, assuming that zero was obtained by measuring the state of c_0, we get the state of

$$|\psi_6\rangle = \frac{1}{\sqrt{L}}\sum_{p=1}^{L} m_p|\psi_{\mathrm{rcmd}}\rangle, \tag{30}$$

where $|\psi_{\mathrm{rcmd}}\rangle = |d_1^p, \ldots, d_l^p\rangle \otimes |t_1, \ldots, t_l\rangle \otimes |0\rangle$ and m_p is represented by the probability amplitudes obtained after the measurement. It is possible to emphasize g amplitudes for recommended elements denoted as m_p^r and also $L - g$ amplitudes with lower compatibility denoted by m_p^{nr}:

$$|\psi_7\rangle = \frac{1}{\sqrt{L}} \left(\sum_{p=1}^{g} m_p^r |\psi_{\text{rcmd}}\rangle + \sum_{p=g+1}^{L} m_p^{nr} |\psi_{\text{rcmd}}\rangle \right). \tag{31}$$

The average for amplitudes and variance for probability amplitudes for state $|\psi_7\rangle$ take the following form based on the work of (Biham *et al.*, 1999):

$$\begin{aligned} \overline{m^r}(t) &= \frac{1}{g} \sum_{p=1}^{g} m_p^r(t), \\ \overline{m^{nr}}(t) &= \frac{1}{g} \sum_{p=g+1}^{L} m_p^{nr}(t), \\ \sigma_{nr}^2(t) &= \frac{1}{L-g} \sum_{p=g+1}^{L} |m_p^{nr}(t) - \overline{m^{nr}}(t)|^2, \end{aligned} \tag{32}$$

where t is the iteration number commonly referred to as the execution time of the Grover algorithm, for $t = 0$; of course, the initial values of the distribution of probability amplitudes are known (Biham *et al.*, 1999). The highest probability of measuring the existing recommended elements from the database is described as

$$\begin{aligned} P_{\max} = 1 &- (L-g)\sigma_{nr}^2 \\ &- \frac{1}{2}\left((L-g)|\overline{m^{nr}}(0)|^2 + g|\overline{m^r}(0)|^2 \right) \\ &+ \left(\frac{1}{2} |(L-g)\overline{m^{nr}}(0)^2 + g\overline{m^r}(0)^2| \right), \end{aligned} \tag{33}$$

with the $\mathcal{O}(\sqrt{L/g})$ iteration.

4. Numerical experiments

The given algebraic relations describing the states of the quantum register after particular stages of the implementation of Algorithm 1 allow us to give numerical examples presenting the behaviour of the recommending system. Figure 4 presents probability distributions showing the state of the registry for a database of 16 elements. There have been cases where one element f_{15} or two elements f_6 and f_{13} are amplified in the database that match the user's expectations

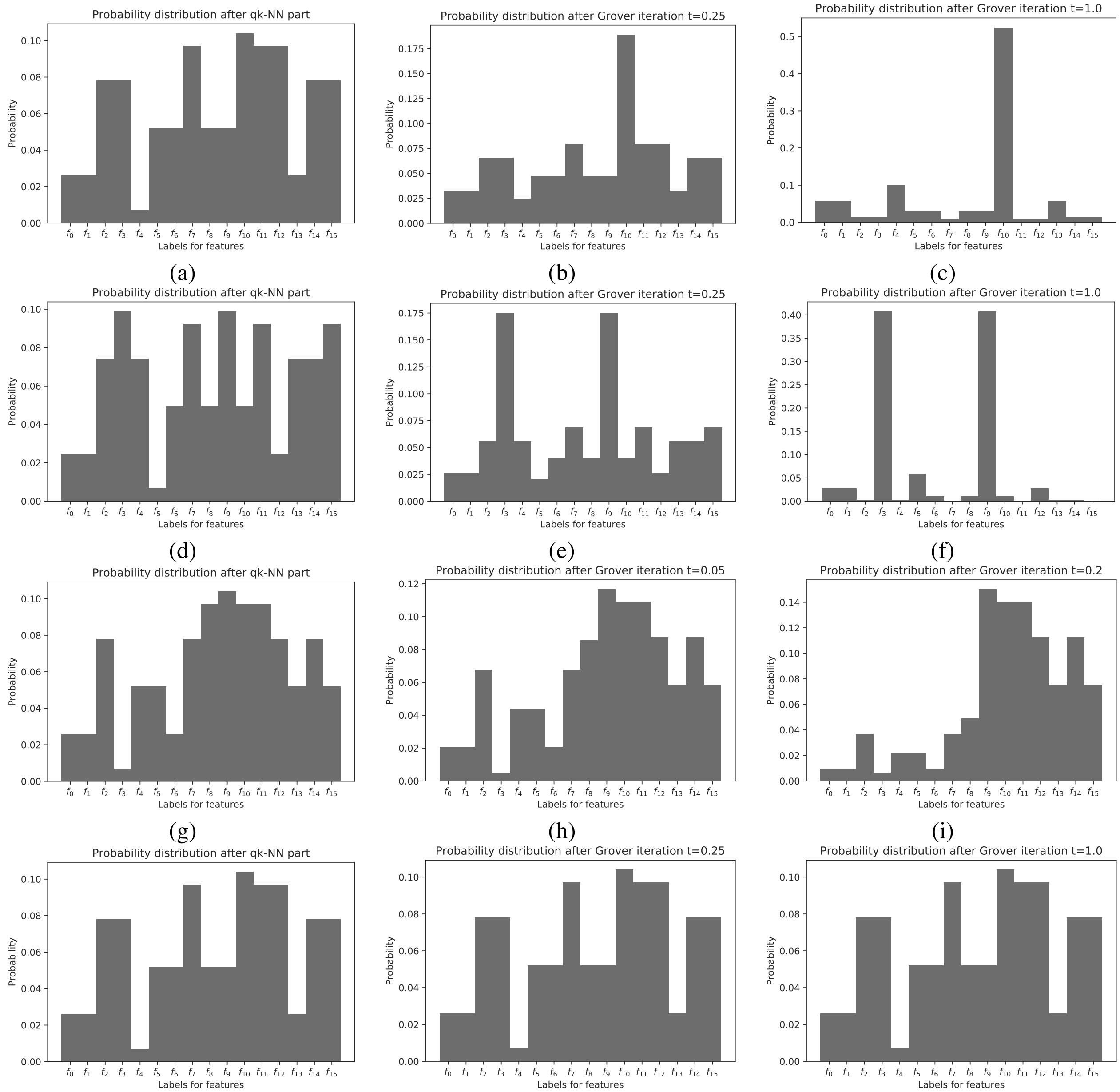
Regardless of the case, Fig. 4(a) shows the probability distribution for the database after the execution of the k-NN part. In both the cases we have no explicit indication of the element nearest to the user feature, although, for the eligible feature, for example, f_{10}, we have the highest probability of measuring this state. However, since the difference in the Hamming distance for the other features was not significantly large, the other elements also have a high probability of measurement. If we stop the process of recommendation at this stage, we should repeat the implementation of Steps (I)–(IV) from Algorithm 1 and the measurement operation to collect the results with the probability described by Eqn. (29).

It should also be emphasized that in Figs. 4(b) and (c) we describe the case of a recommendation of only one element. However, the quality of the amplification will be good for two or more elements that will be indicated by a user-specified characteristic, which is presented in Figs. 4(e) and (f). It should be emphasized that the sum of probabilities of recommended elements is determined by Eqn. (33).

Remark 6. (*Use of Grover's algorithm*) At this moment, one can ask whether instead of using k-NN we can employ only the Grover algorithm to amplify the recommended element or more elements. The answer is affirmative only if there is an element or elements with the same vector of features as the user's requirements. If, however, there is no such element, then the Grover algorithm will not be able to amplify elements similar to those defined by the user. It should be added that you do not have to specify that we are interested in amplify only one element that fully complies with the user-defined characteristic. You can specify that we amplify the k-NN result for a leading user-specified characteristic (e.g., historical films, and the idea of such a gain is shown in Figs. 4(f)–(h)). Therefore, the combination of two quantum k-NN algorithms and amplitude amplification using the Grover algorithm allows building a recommendation system.

According to Remark 6, it is reasonable to use the Grover algorithm to amplify the probability for the element (elements) indicated by k-NN. For $t > 1$ we already have a very high probability (close to the maximum theoretical value) to measure the recommended element. However, even for smaller values of t it amplifies the desired effect, while maintaining the probability distribution obtained after the quantum part k-NN. This also means shortening the circuit's operational time, because you do not have to do additional iterations during the amplification with the help of the Grover algorithm.

The legitimacy of the application of the Grover algorithm is also shown the cases described by Fig. 4 (g)–(i) where the user's feature does not precisely indicate a single element or two, but the whole group. The amplification for the group also retains the differences in probabilities for individual elements. Additionally, the bigger group of amplified amplitudes makes the value of time smaller for other cases where only one or two elements were recommended. In the case when there is no recommendation, the initial probability distribution obtained after executing the k-NN part will be preserved, which is presented in the example described Fig. 4(j)–(l).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 4. Distribution of probability in the quantum register (for a database with 16 elements) at three main stages in the proposed recommendation system. The probability after the stage of database initialization is not shown because of an even probability distribution for each case. In this case each feature f_i has a probability of 1/16. Cases (a), (d), (g) and (j) represent the quantum state after the execution of the part realisations of the k-NN algorithm. Cases (b), (c) and (e), (f) show the process of amplifying the amplitude of one (feature f_{10}) and two recommended elements (f_3, f_9) in the database using the Grover algorithm. Cases (h), (i) describe a situation when the feature described by the user indicates a larger group of searched elements and Grover amplification affects a group of matching elements (labels from f_9 to f_{15} and labels are grouped for better clarity of plots). Cases (k), (l) present a situation when in a database there are no elements compatible with the user's feature; in this case the initial probability distribution is preserved and the Grover algorithm does not amplify any elements.

5. Conclusions

The article presented the structure of the recommendation system based on the quantum k-NN and Grover algorithms. The discussed approach is characterized by better computational complexity in the recommendation process. However, the construction of the database can be done only once, at the beginning of the mentioned stage. The process of its construction depends only on the amount of data. The probability value of the recommendation's success was also indicated, showing that it depends directly on the Hamming distance and amplitude amplification using the Grover algorithm.

One of the next tasks may be a different approach to verifying the correctness operation of the system using the quantum predicate system (D'Hondt and Panangaden, 2006; Gielerak and Sawerwain, 2010). An attempt to implement the presented system is also planned to check the quality of recommendations on existing systems of experimental installations of quantum computing systems (IBM, 2018). They offer access to a quantum register of 20 qubits. This is not a sufficient number to build a larger system. However, technological advances seem to soon allow the construction of a system containing over 50 qubits, and this size is already available in the system (IBM, 2018).

The sample full database includes over 12,000 records. However, in the case of quantum calculations, the quantum register stores an exponential amount of data. Therefore, in the case of the OMDB movie database and indexes of individual films, we only need 15 qubits: $2^{15} > 12,000$. Therefore, if nearby quantum processing systems offer access to 50 qubits, this number will allow full implementation of the proposed solution.

Acknowledgment

We appreciate the useful discussions within the *Q-INFO* group at the Institute of Control and Computation Engineering of the University of Zielona Góra, Poland. We would like also to thank the anonymous referees for their comments on the preliminary version of this paper. The numerical results were obtained using the hardware and software available at the GPU μ-Lab located at the Institute of Control and Computation Engineering of the University of Zielona Góra.

References

Aaronson, S. and Gottesman, D. (2004). Improved simulation of stabilizer circuits, *Physical Review A* **70**(5): 052328, DOI: 10.1103/PhysRevA.70.052328.

Alpaydin, E. (2004). *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*, Massachusetts Institute of Technology Press, Cambridge, MA.

Arikan, E. (2003). An information-theoretic analysis of Grover's algorithm, *in* A.S. Shumovsky and V.I. Rupasov (Eds.), *Quantum Communication and Information Technologies*, Springer Netherlands, Dordrecht, pp. 339–347.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2010). A view of cloud computing, *Communications of the Association for Computing Machinery* **53**(4): 50–58, DOI: 10.1145/1721654.1721672.

Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A. and Weinfurter, H. (1995). Elementary gates for quantum computation, *Physical Review A* **52**(5): 3457–3467, DOI: 10.1103/PhysRevA.52.3457.

Biham, E., Biham, O., Biron, D., Grassl, M. and Lidar, D. (1999). Grover's quantum search algorithm for an arbitrary initial amplitude distribution, *Physical Review* **60**(4): 2742–2745, DOI: 10.1103/PhysRevA.60.2742.

Brassard, G. and Hoyer, P. (1997). An exact quantum polynomial-time algorithm for Simon's problem, *Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems, Ramat Gan, Israel*, DOI: 10.1109/ISTCS.1997.595153.

Busemeyer, J. and Bruza, P. (2012). *Quantum Models of Cognition and Decision*, Cambridge University Press, Cambridge.

Chakrabarty, I., Khan, S. and Singh, V. (2017). Dynamic grover search: Applications in recommendation systems and optimization problems, *Quantum Information Processing* **16**(6): 153, DOI: 10.1007/s11128-017-1600-4.

D'Hondt, E. and Panangaden, P. (2006). Quantum weakest preconditions, *Mathematical Structures in Computer Science* **16**(3): 429–451.

Galindo, A. and Martin-Delgado, M. A. (2000). A family of Grover's quantum searching algorithms, *Physical Review A* **62**(6): 062303, DOI: 10.1103/PhysRevA.62.062303.

Galindo, A. and Martin-Delgado, M.A. (2002). Information and computation: Classical and quantum aspects, *Reviews of Modern Physics* **74**(2): 347–423, DOI: 10.1103/RevModPhys.74.347.

Gielerak, R. and Sawerwain, M. (2010). Generalised quantum weakest preconditions, *Quantum Information Processing* **9**(4): 441–449, DOI: 10.1007/s11128-009-0151-8.

Grover, L.K. (1996). A fast quantum mechanical algorithm for database search, *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, STOC'96, Philadelphia, PA, USA*, pp. 212–219, DOI: 10.1145/237814.237866.

Hechenbichler, K. and Schliep, K. (2004). Weighted k-nearest-neighbor techniques and ordinal classification, *Technical report*, Ludwig-Maximilians-Universität München, München, `https://epub.ub.uni-muenchen.de/1769/1/paper_399.pdf`.

IBM (2018). Q Experience, `https://quantumexperience.ng.bluemix.net/`.

Li, C.-K., Roberts, R. and Yin, X. (2013). Decomposition of unitary matrices and quantum gates, *International Journal of Quantum Information* **11**(1): 1350015, DOI: 10.1142/S0219749913500159.

Montanaro, A. (2017). Quantum pattern matching fast on average, *Alghoritmica: An International Journal in Computer Science* **77**(1): 16–39, DOI: 10.1007/s00453-015-0060-4.

Nielsen, M. and Chuang, I. (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, Cambridge.

Nielsen, P. (2016). Big data analytics—a brief research synthesis, *in* L. Borzemski *et al.* (Eds.), *Information Systems Architecture and Technology*, Springer International Publishing, Cham, pp. 3–9.

OMDb (2018). Homepage, `http://www.omdbapi.com/`.

Pinkse, P., Goorden, S., Horstmann, M., Skoric, B. and Mosk, A. (2013). Quantum pattern recognition, *Conference on Lasers and Electro-Optics Europe (CLEO EUROPE/IQEC) and International Quantum Electronics Conference, Munich, Germany*, p. 1–1.

Santucci, E. (2017). Quantum minimum distance classifier, *Entropy* **19**(12): 659, DOI: 10.3390/e19120659.

Sawerwain, M. and Wróblewski, M. (2019). Application of quantum k-nn and Grover's algorithms for recommendation big-data system, *in* L. Borzemski *et al.* (Eds.), *Information Systems Architecture and Technology*, Springer International Publishing, Cham, pp. 235–244.

Schuld, M., Sinayskiy, I. and Petruccione, F. (2014). Quantum computing for pattern classification, *in* D.-N. Pham and S.-B. Park (Eds.), *PRICAI 2014: Trends in Artificial Intelligence*, Springer International Publishing, Cham, pp. 208–220.

Sergioli, G., Bosyk, G.M., Santucci, E. and Giuntini, R. (2017). A quantum-inspired version of the classification problem, *International Journal of Theoretical Physics* **56**(12): 3880–3888, DOI: 10.1007/s10773-017-3371-1.

Sergioli, G., Santucci, E., Didaci, L., Miszczak, J.A. and Giuntini, R. (2018). A quantum-inspired version of the nearest mean classifier, *Soft Computing* **22**(3): 691–705, DOI: 10.1007/s00500-016-2478-2.

Shende, V. and Markov, I.L. (2009). On the CNOT-cost of TOFFOLI gates, *Quantum Information & Computation* **9**(5): 461–486.

Shor, P. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review* **41**(2): 303–332, DOI: 10.1137/S0036144598347011.

Steane, A. (1998). Quantum computing, *Reports on Progress in Physics* **61**(2): 117–173, DOI: 10.1088/0034-4885/61/2/002.

Stefanowski, J., Krawiec, K. and Wrembel, R. (2017). Exploring complex and big data, *International Journal of Applied Mathematics and Computer Science* **27**(4): 669–679, DOI: 10.1515/amcs-2017-0046.

Trugenberger, C.A. (2002). Quantum pattern recognition, *Quantum Information Processing* **1**(6): 471–493, DOI: 10.1023/A:1024022632303.

Veloso, B., Malheiro, B. and Burguillo, J.C. (2015). A multi-agent brokerage platform for media content recommendation, *International Journal of Applied Mathematics and Computer Science* **25**(3): 513–527, DOI: 10.1515/amcs-2015-0038.

Walther, P., Resch, K.J., Rudolph, T., Schenck, E., Weinfurter, H., Vedral, V., Aspelmeyer, M. and Zeilinger, A. (2005). Experimental one-way quantum computing, *Nature* **434**(0): 169–176, DOI: 10.1038/nature03347.

Wiebe, N., Kapoor, A. and Svore, K. (2015). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning, *Quantum Information and Computation* **15**(3–4): 316–356.

Wiśniewska, J. and Sawerwain, M. (2018). Recognizing the pattern of binary Hermitian matrices by quantum knn and SVM methods, *Vietnam Journal of Computer Science* **5**(3): 197–204, DOI: 10.1007/s40595-018-0115-y.

Marek Sawerwain received the MSc and PhD degrees at the Faculty of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Góra, Poland, in 2004 and 2010, respectively. He is currently with the same faculty as an assistant professor. His present research interests include quantum communication methods, quantum computation methods, the theory of quantum programming languages, simulations of quantum computations models and creating effective algorithms for solutions based on modern multicore CPU and GPU technology.

Marek Wróblewski received the MSc degree at the Faculty of Computer, Electrical and Control Engineering, University of Zielona Góra, Poland, in 2015. He is currently associated with the same faculty as an assistant lecturer and a PhD student. His present research interests include big data technologies, quantum computation, databases and programming languages.

Received: 20 November 2018
Revised: 7 January 2019
Accepted: 18 January 2019