entropy

IMPACT FACTOR 2.738

Indexed in: PubMed

Article

An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance

Congcong Feng, Bo Zhao, Xin Zhou, Xiaodong Ding and Zheng Shan

Special Issue
Advances in Quantum Computing

Edited by
Dr. Brian R. La Cour and Dr. Giuliano Benenti

MDPI

https://doi.org/10.3390/e25010127

entropy

MDPI

Article

An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance

Congcong Feng [1,2,†], Bo Zhao [2,3,†], Xin Zhou [2], Xiaodong Ding [2] and Zheng Shan [2,3,*]

[1] School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China
[2] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[3] Songshan Laboratory, Zhengzhou 450001, China
* Correspondence: zzzhengming@163.com
† These authors contributed equally to this work.

Abstract: The K-nearest neighbor (KNN) algorithm is one of the most extensively used classification algorithms, while its high time complexity limits its performance in the era of big data. The quantum K-nearest neighbor (QKNN) algorithm can handle the above problem with satisfactory efficiency; however, its accuracy is sacrificed when directly applying the traditional similarity measure based on Euclidean distance. Inspired by the Polar coordinate system and the quantum property, this work proposes a new similarity measure to replace the Euclidean distance, which is defined as Polar distance. Polar distance considers both angular and module length information, introducing a weight parameter adjusted to the specific application data. To validate the efficiency of Polar distance, we conducted various experiments using several typical datasets. For the conventional KNN algorithm, the accuracy performance is comparable when using Polar distance for similarity measurement, while for the QKNN algorithm, it significantly outperforms the Euclidean distance in terms of classification accuracy. Furthermore, the Polar distance shows scalability and robustness superior to the Euclidean distance, providing an opportunity for the large-scale application of QKNN in practice.

Keywords: quantum computation; quantum machine learning; K-nearest neighbor algorithm; quantum K-nearest neighbor algorithm

check for updates

Citation: Feng, C.; Zhao, B.; Zhou, X.; Ding, X.; Shan, Z. An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance. *Entropy* **2023**, *25*, 127. https://doi.org/10.3390/e25010127

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 4 December 2022
Revised: 4 January 2023
Accepted: 4 January 2023
Published: 8 January 2023

1. Introduction

Machine learning has made remarkable achievements in various artificial intelligence applications, such as object detection [1–4], image classification [5–8], and natural language processing [9–11]. However, in the era of big data, we are facing the problem of rapid growth in the amount and type of data. We urgently need to find more high-efficiency computing methods. The quantum system with natural parallelism looks like a good choice. With the in-depth study of quantum technology, many quantum algorithms showing quantum superiority have been proposed [12–15]. Researchers found that quantum and machine learning algorithms can be combined to improve the performance of the algorithm. The concept of quantum machine learning was born [16,17]. Many quantum machine learning algorithms [18–22] are significantly better than their classical counterparts. In this context, a KNN algorithm with a simple idea but high time complexity has attracted the interest of researchers. It requires little to no prior knowledge when classifying [23]. Similarity calculation and K-nearest neighbors search are two important parts of KNN. In recent years, many quantum methods for these two processes have been proposed. In 2001, Harry Buhrman et al. proposed the swap test quantum circuit for calculating the cosine distance of two vectors [24]. In 2013, Lloyd et al. proposed a quantum Euclidean distance estimator based on the swap test circuit [25]. Based on this, Wiebe et al. proposed a quantum nearest neighbor algorithm [26] in 2014 and used Dürr and Høyer's algorithm for finding the minimum value in a database [27] to find the nearest neighbor. For nonnumerical data, a quantum K-nearest neighbor algorithm based on Hamming distance is proposed [28,29].

The similarity measure, which affects the accuracy of the algorithm classification, lies at the heart of the K-nearest neighbor algorithm [30]. A similarity measure is used to measure how similar two things are [31]. To date, many similarity measures have been proposed, such as Euclidean distance, cosine distance, Hamming distance, and so on. However, there is no one similarity distance measure that can best solve all problems [31]. Choosing an appropriate similarity measure will significantly improve the K-nearest neighbor algorithm's classification accuracy. The Euclidean distance is the most frequently applied similarity measure. However, the result of the quantum Euclidean distance estimator has poor stability, and there is a significant difference with the actual result [32]. Therefore, we need to find a new similarity measure to replace the use of Euclidean distance in QKNN.

In machine learning, a sample is usually regarded as a vector with both magnitude and direction. Inspired by this, in addition to using Cartesian coordinates, we can also use Polar coordinates to represent a sample. So, we propose a new similarity measure that we call Polar distance, which considers both angular and module length information. The cosine theorem shows that the Euclidean distance is a combination of angular and module length information. The Polar distance introduces an adjustable parameter to adjust the ratio of angular and module length information according to the specific application. Then, we propose a quantum circuit to calculate the Polar distance. The frame diagram of the quantum part of the QKNN algorithm is shown in Figure 1. We optimize Figure 1b in this work. The following is a list of our major contributions:

(1) We propose a new similarity measure, called Polar distance, which integrates both angular and module length information and combines the two proportionally according to practical applications. Its classification accuracy in KNN is comparable to that of Euclidean distance;

(2) We design a quantum circuit to calculate the Polar distance. Compared with the quantum Euclidean distance estimator, it can directly obtain the desired results and has less difference with the classical results;

(3) We carry out KNN and QKNN(quantum simulation) experiments on different datasets. The KNN's experimental results show that Polar distance is comparable to Euclidean distance in classification accuracy. The QKNN's experimental results show that Polar distance is better than Euclidean distance in classification accuracy.

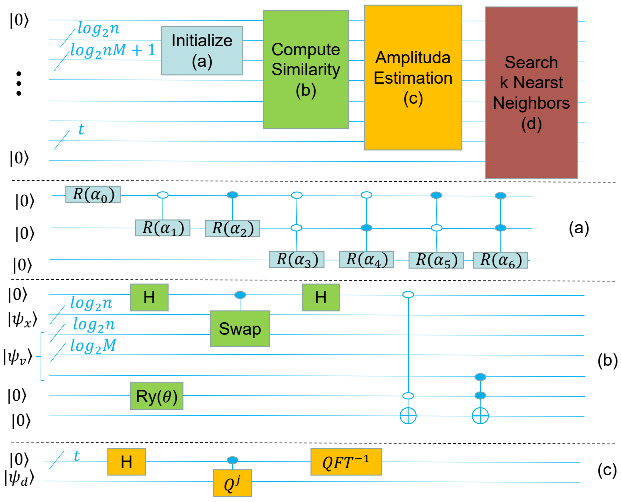

Figure 1. Frame diagram of quantum part of QKNN algorithm: (**a**) initialize any quantum state circuit, where α is the calculated parameter based on the quantum state vector representation; (**b**) quantum circuit for calculating Polar distance, where $|\psi_x\rangle$ represents the test sample and $|\psi_v\rangle(|\psi_v\rangle = |v_i\rangle|i\rangle|d_{r_i}\rangle)$ represents the entangled superposition state of the module length similarity and the training set; (**c**) amplitude estimation circuit diagram.

2. Materials and Methods

2.1. KNN

K-nearest neighbor algorithm is a supervised machine learning algorithm [23]. Its general idea is: if most of the K-nearest samples to a sample belong to a given category in the feature space, then the sample belongs to that category as well. The whole process of KNN is shown in Algorithm 1. Its main steps are as follows: first, calculate the similarity between the test sample and all training samples; then find the k training samples that are most like the test sample; finally, according to the category of k training samples, the category of the test sample is determined according to the principle that the minority obeys the majority. For example, as shown in Figure 2, the question mark represents the unknown test sample, and the red circle and blue cross denote two categories of training samples. At $k = 1$, the category of the question mark is consistent with the red circle category. When $k = 5$, the category of the question mark is consistent with the blue cross category. Obviously, the classification results will be affected by k value. Furthermore, similarity measure is another factor influencing the results of classification.

Algorithm 1 KNN

```
Input: A test sample and some training samples
Output: The test sample's category
for number of training samples do do
    calculate the similarity between the test sample and a training sample
end for
find the $k$ training samples that are most like the test sample
determine test sample's category
```

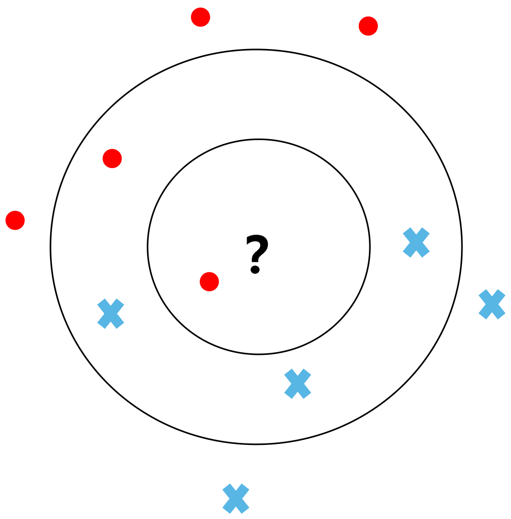

Figure 2. Schematic of the KNN algorithm.

2.2. QKNN

The quantum K-nearest neighbor algorithm is consistent with the overall idea of the classical K-nearest neighbor algorithm. Quantum K-nearest neighbor algorithm quantizes the part of K-nearest neighbor algorithm with high time complexity. It uses the natural parallelism of quantum computing to reduce the time complexity of the algorithm. As shown in Figure 1, the quantum part of the quantum K-nearest neighbor proposed in this paper consists of four parts: Initialize, Compute Similarity, Amplitude Estimation, and Search K-Nearest Neighbors. Finally, the test sample category is determined by the classical method. We describe these five parts in detail later.

2.2.1. Initialize

In order to process classical data using the quantum system, we need to encode classical data into quantum state. At present, there are many methods to encode classical data into quantum states [33–35]. The coding methods can be divided into two categories: using the amplitude of the quantum state to encode information and directly using the quantum state to encode information. Amplitude coding is one of the common coding methods in quantum machine learning algorithms [36]. In this paper, we also use amplitude coding. Its main idea is to use the amplitude of quantum states to represent classical data. In order to represent the classical vector by amplitude, we must first normalize the vector so that the vector module length is 1. After that, we should ensure that the dimension of the vector is 2^n, and n is the number of qubits required to encode the vector. When the vector does not meet this condition, it is completed by supplementing 0. Take vector $\vec{a}$ as an example.

$$\vec{a} = (a_0, a_1, \ldots, a_{2^n-1}) \tag{1}$$

Its quantum representation is as follows:

$$|\psi_a\rangle = \sum_{i=0}^{2^n-1} \frac{a_i}{\sqrt{|a_0|^2 + |a_1|^2 + \ldots + |a_{2^n-1}|^2}} |i\rangle \tag{2}$$

Then, we need to initialize the register of n qubits as $|\psi_a\rangle$. The initial state is $|0\ldots0\rangle$, we start from the high position, and the quantum circuit is shown in Figure 1a. The Ry represents single-qubit rotation about the Y-axis, the solid point represents 1 control, and the hollow point represents 0 control.

2.2.2. Compute Similarity

Inspired by Polar coordinates, we propose a parametric similarity measure that combines cosine similarity and module length similarity, which we call Polar distance.

$$d = d_c \cdot (1-\omega) + d_r \cdot \omega \tag{3}$$

Among them, d_c and d_r represent cosine similarity and module length similarity, respectively, ω represents an adjustable parameter, and the value range of the three is [0, 1]. As d increases, the similarity between the two samples is stronger, otherwise this similarity becomes weaker. We can adjust the value of ω to improve the classification accuracy according to the actual application. The formula for calculating the cosine similarity is as follows (θ represents the angle between two vectors):

$$d_c = 0.5 \cdot (1 + cos^2\theta) \tag{4}$$

The formula for calculating the module length similarity is as follows:

$$d_r = 1 - |r_x - r_v| \tag{5}$$

where r_x refers to the module length of the test sample and r_v represents the module length of the training sample. The closer the d_r value is to 1, the greater the similarity between the two samples, and the closer the d_r value is to 0, the smaller the similarity between the two samples.

As shown in Figure 1b, the similarity calculation is divided into two steps: first, the swap test circuit is applied to calculate the cosine similarity [24], and then the weighted summation circuit proposed by us is used to realize the weighted summation of the cosine similarity and the module length similarity.

Next, we take the calculation of the similarity between two samples x and v as an example to introduce the calculation process in detail. The initial state of the quantum system is:

$$|s_0\rangle = |0\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{6}$$

First of all, after a Hadamard gate is utilized, the state of the quantum system becomes:

$$|s_1\rangle = |+\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{7}$$

Then, after the usage of CSWAP gate, the state of the quantum system transforms into:

$$|s_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|x\rangle|v\rangle + |1\rangle|v\rangle|x\rangle)(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{8}$$

At the third stage, another Hadamard gate is applied. The state of the quantum system is given as:

$$\begin{aligned} |s_3\rangle = \frac{1}{2}(|0\rangle|x\rangle|v\rangle + |1\rangle|x\rangle|v\rangle + |0\rangle|v\rangle|x\rangle - |1\rangle|v\rangle|x\rangle) \\ (\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \end{aligned} \tag{9}$$

At the fourth stage, the extended general Toffoli gate is applied, and the state of the quantum system reads:

$$\begin{aligned} |s_4\rangle = &\frac{1}{2}|0\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|1\rangle + \sqrt{\omega}|1\rangle|0\rangle) \\ &+ \frac{1}{2}|1\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|0\rangle + \sqrt{\omega}|1\rangle|0\rangle) \\ &+ \frac{1}{2}|0\rangle|v\rangle|x\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|1\rangle + \sqrt{\omega}|1\rangle|0\rangle) \\ &+ \frac{1}{2}|1\rangle|v\rangle|x\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|0\rangle + \sqrt{\omega}|1\rangle|0\rangle) \end{aligned} \tag{10}$$

At the fifth stage, the Toffoli gate is applied, and the state of the quantum system becomes:

$$\begin{aligned} |s_5\rangle = &\frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle \\ &+ \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle \\ &+ \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|0\rangle + \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle \\ &+ \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|0\rangle + \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle \\ &+ \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle \\ &+ \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle \\ &- \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|0\rangle - \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle \\ &- \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|0\rangle - \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle \end{aligned} \tag{11}$$

Finally, the last qubit is measured. The probability of getting state $|1\rangle$, which measures the last qubit with the basis state $|1\rangle$, is given by:

$$p(|1\rangle) = (\frac{1}{2} + \frac{1}{2}|\langle x|v\rangle|^2)(1-\omega) + d_r\omega \tag{12}$$

2.2.3. Amplitude Estimation

There are two methods to obtain the results of similarity calculation of quantum circuits. The first method is to obtain statistical results through multiple measurements. The disadvantage of this method is that it cannot determine the accuracy of the results. The other method is to use an amplitude estimation algorithm [37]. Amplitude estimation can control the accuracy of the results by adjusting the number of qubits. In addition, the use of amplitude estimation is more convenient for subsequent quantum steps. So, we use amplitude estimation. In this article, we only introduce the usage of amplitude estimation. For more details, please refer to [37]. The function of amplitude estimation is to calculate the value of a in Equation (13). The circuit of amplitude estimation is shown in Figure 1c. The first step is to initialize the two registers with status $|0\rangle A|0\rangle$. The second step is to apply QFT to the first register. The third step is to apply a controlled $Q^j(Q = -AS_0A^{-1}S_X)$. The fourth step is to apply QFT^{-1} to the first register. The fifth step is to measure the first register and denote the outcome $|y\rangle$. Finally, calculate the amplitude $a = sin(\pi\frac{y}{2^t})$. We can control the precision of the result by adjusting the number of qubits t. The higher the value of t, the higher the precision of the result. On the contrary, the lower the precision. The functions of unitary operators A, S_0 and S_X are as follows:

$$A|0\rangle = a|\psi\rangle + \sqrt{1-a^2}|\psi_\perp\rangle \tag{13}$$

$$S_0 = I - 2|0\rangle\langle 0| \tag{14}$$

$$S_X = I - 2|\psi\rangle\langle\psi| \tag{15}$$

In order to apply the quantum algorithm for finding the K-nearest neighbors, we need to make the amplitude estimation step reversible. Wiebe et al. call this form of amplitude estimation coherent amplitude estimation [26]. This results in a state that is, up to local isometries, approximately

$$\frac{1}{\sqrt{M}}\sum_{j=0}^{M}|j\rangle||x-v_j|\rangle \tag{16}$$

2.2.4. Search K-Nearest Neighbors

Searching K-nearest neighbors is a part of a KNN with high time complexity. The appearance of the Grover algorithm opens up a new idea for the unordered search problem [12]. Dürr proposed a quantum algorithm [38] for finding k-minimum values in 2004. Miyamoto proposed a quantum algorithm to find K-minimum values with another idea in 2019 [39]. Both their algorithms are capable of finding k-minimum values in M data with a time complexity of $O(\sqrt{kM})$. Miyamoto's algorithm is simpler and easier to implement. Here, we will present their method. He introduced a parameter t, which is used to find out k values that are less than t. The quantum algorithm for finding k-Minima is summarized as follows:

(1) Apply algorithm [27] for finding minimum and record the last k indexes of finding minimum algorithm process;

(2) Use binary search to find the minimum algorithm record of the threshold index t that meets the condition that the number less than t is close to k. The quantum counting algorithm is used to determine whether the condition is satisfied;

(3) Apply Grover algorithm to search k values that are less than t.

According to [29], the time complexity of the first step is $O(\sqrt{M})$. The second step combines the quantum counting algorithm with the classical binary search. Its time complexity is $O(\sqrt{M}logk)$. Finally, the time complexity of searching K indexes is $O(\sqrt{k}M)$. To sum up, the overall time complexity of the algorithm is given as:

$$(\sqrt{M}) + O(\sqrt{M}logk) + O(\sqrt{kM}) = O(\sqrt{kM}) \tag{17}$$

2.2.5. Determine Category

Finally, we determine the category of the test sample according to the k most similar training samples. Suppose the number of each category in the k most similar training samples is k_i. The category of test samples is consistent with the index of $max(k_i)$. However, in practical application, $max(k_i) = k_a = k_b(a \neq b)$ may take place, which makes it impossible to determine the type of the test sample. In this paper, once the onset of above issue, we make $k = k + 1$ until we can determine the category of the test sample.

3. Results

In this section, we first demonstrate theoretically that the Polar distance can be used as a measure of sample similarity. We then compare the performance of Polar distance and Euclidean distance in KNN on the Iris, Wine, Liver, and Overflow Vulnerability datasets. Finally, we compared the performance of Polar distance and Euclidean distance in QKNN on the same dataset. The accuracy of all experiments is the average of 30 10-fold cross-validations.

3.1. A New Similarity Distance Measure

Similarity measure is a metric for comparing the similarity of two samples. When comparing two samples, distance is usually used to determine their similarity. In this paper, we proposed a new similarity distance measure called Polar distance that considers both the information of angle and module length by combining them into a weighted value. In general, distance should meet the following three properties: non-negativity, symmetry, and trigonometric inequality. Derived by cosine similarity, the angle can be used as an index to measure the similarity. Here, we prove that module length can be used as an indicator for similarity measurement from the three properties of distance above. In this work, we define the module length distance of two samples A and B as:

$$|r_A - r_B| \tag{18}$$

The first is non-negativity and symmetry. Obviously,

$$|r_A - r_B| \geq 0 \tag{19}$$

$$|r_A - r_B| = |r_B - r_A| \tag{20}$$

Finally, it is proved that it satisfies the trigonometric inequality. Considering any three samples, such as A, B, and C, it is necessary to prove

$$|r_A - r_B| + |r_A - r_C| \geq |r_B - r_A + r_A - r_C| = |r_B - r_C| \tag{21}$$

Obviously, it satisfies the trigonometric inequality. Module length distance can be used as an indicator to measure similarity. In order to combine the angle and module length as indicators to measure similarity, we define the new similarity measurement method as the following form:

$$d = 0.5 \cdot (1 + cos^2\theta) \cdot (1 - \omega) + (1 - |r_A - r_B|) \cdot \omega \tag{22}$$

where module length r_A and r_B are scaled so that their value range is [0, 1]. The ω values in this paper were determined using cross-validation. Specifically, the value of k under Euclidean distance is first determined using cross-validation. The value of k is then held constant, and we determine the parameter ω for the Polar distance using the cross-validation method.

3.2. Polar Distance and Euclidean Distance in KNN

To verify that the Polar distance can replace the Euclidean distance in KNN, we first tested the classification accuracy of two similarity distance measures in KNN under different datasets. Iris and Wine are datasets with three classes. Overflow Vulnerability and Liver are datasets with two classes. There is not much difference in the classification accuracy of the two similarity distance measures, as shown in Figures 3–6. Table 1's KNN column shows that the difference between the two similarity distance measures is still small in the best shape of the classification accuracy. Therefore, we consider that the two similarity distance measures are approximately equivalent in KNN.

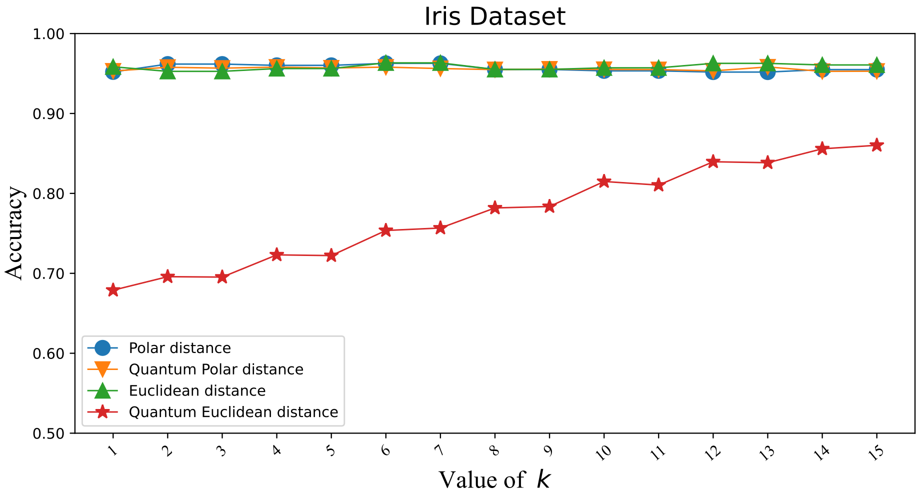

Figure 3. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Iris dataset.

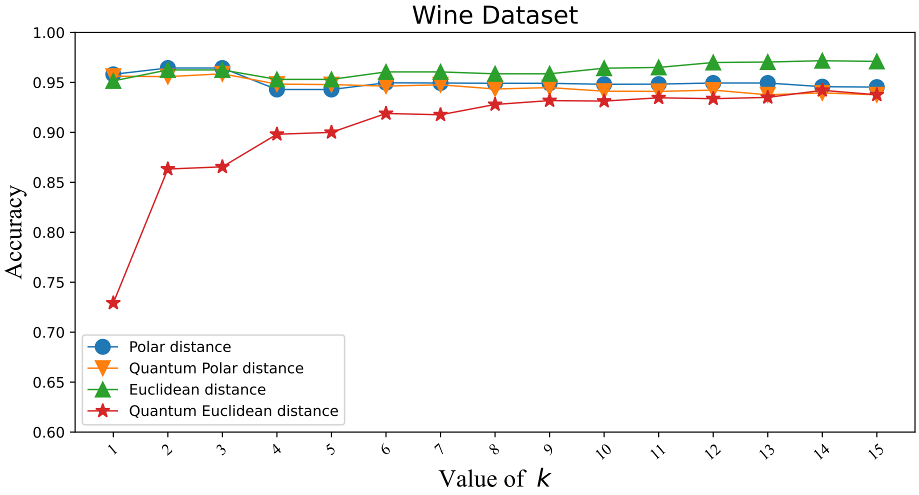

Figure 4. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Wine dataset.

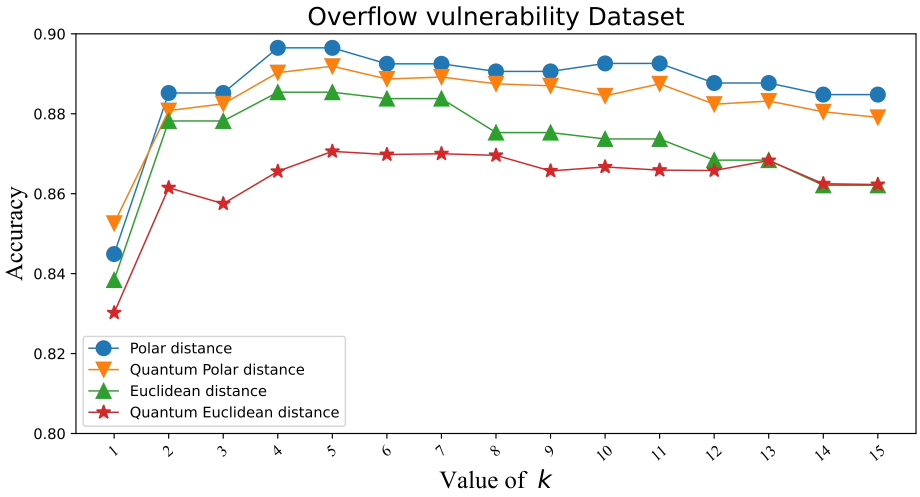

Figure 5. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Overflow Vulnerability dataset.

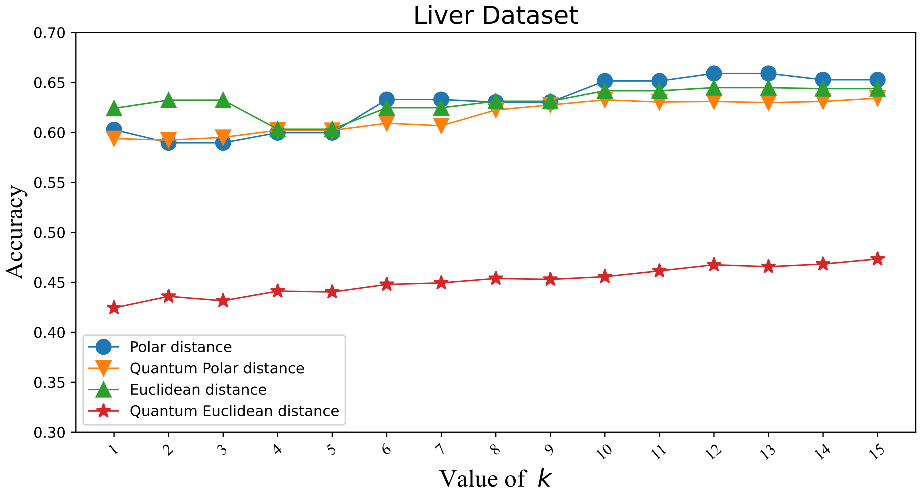

Figure 6. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Liver dataset.

Table 1. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on four datasets.

Datasets	**KNN**		**QKNN**	
	Polar Distance	**Euclidean Distance**	**Polar Distance**	**Euclidean Distance**
Iris	96.27%	96.33%	95.82%	86.02%
Wine	96.44%	97.17%	95.86%	94.21%
Overflow	89.65%	88.54%	89.19%	87.06%
Liver	65.90%	64.48%	63.42%	47.33%

3.3. Polar Distance and Euclidean Distance in QKNN

To validate that the Polar distance can replace the Euclidean distance in QKNN, we make similar experiments as above. As shown in Figures 3–6, we can easily see that the gap between the Polar distance and quantum Polar distance is significantly smaller than the gap between the Euclidean distance and quantum Euclidean distance. From the results in the QKNN column of Table 1, the Polar distance we proposed is better than the Euclidean distance in classification accuracy. For the dataset of Iris, the accuracy of Polar distance is 95.82%, achieving a 9.8% accuracy gain against Euclidean distance. For the dataset of Wine, the accuracy of Polar distance is 95.86%, achieving a 1.65% accuracy gain against Euclidean distance. For the dataset of Liver, the accuracy of Polar distance is 89.19%, achieving a 2.13% accuracy gain against Euclidean distance. For the dataset of Overflow Vulnerability, the accuracy of Polar distance is 63.42%, achieving a 16.09% accuracy gain against Euclidean distance. It is well known that there are deviations between quantum results and theoretical values. Although our QKNN experiments are performed by a quantum simulator, this deviation still exists due to Monte Carlo sampling. So why is the deviation from the quantum Euclidean distance greater? This starts with the calculation of the quantum Euclidean distance [32]. The formula for calculating the quantum Euclidean distance is as follows:

$$d = \sqrt{2 * (r_1^2 + r_2^2) * (2 * p(|0\rangle) - 1)} \tag{23}$$

Assume that the error of the quantum measurement result is δ. Obviously the error of the quantum Polar distance is δ. The errors in the quantum Euclidean data are as follows:

$$\frac{|\triangle d|}{d} = \frac{2\delta p}{\sqrt{2p-1} * (\sqrt{2p-1} + \sqrt{2 * (1 \pm \delta)p - 1})} \tag{24}$$

The error of the quantum Euclidean distance is $\frac{|\triangle d|}{d} = \frac{2p}{\sqrt{2p-1}*(\sqrt{2p-1}+\sqrt{2*(1\pm\delta)p-1})}$ (the value of the equation is 1 to $+\infty$ since $p \in [0.5, 1]$) times that of the quantum Polar distance. The classical part of the quantum Euclidean distance estimator amplifies the error in the quantum part, so the results differ significantly from the true results. This leads to less satisfactory results for the quantum K-nearest neighbor (QKNN) algorithm based on Euclidean distance.

4. Discussion

In this paper, we proposed a new similarity distance measure to replace the Euclidean distance for use in QKNN. We call it Polar distance. From the experimental results, the Polar distance can achieve the following results in terms of classification accuracy:

(1) The Polar and Euclidean distances are comparable in KNN;
(2) The Polar distances are comparable in KNN and QKNN;
(3) The Polar distances perform significantly better than the Euclidean distances in QKNN.

However, the disadvantage of the Polar distance is also obvious, namely the introduction of a new parameter ω. This not only increases the computational complexity but also makes Polar distance only applicable in supervised machine learning algorithms. We can try to find the right value of ω quickly by using gradient descent. Experiments have shown that the value of ω is not the same under different datasets. We can also delve into the relationship between the value of ω and the distribution of samples in the dataset to address the above problems. This is worthy of further study.

Author Contributions: Conceptualization, Z.S. and C.F.; methodology, C.F.; software, X.Z.; validation, C.F., B.Z. and X.D.; formal analysis, B.Z.; investigation, X.D.; resources, Z.S.; data curation, X.Z.; writing—original draft preparation, C.F.; writing—review and editing, B.Z.; visualization, X.Z.; supervision, Z.S.; project administration, Z.S.; funding acquisition, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Major Science and Technology Projects in Henan Province, China: 221100210600.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support thefindings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: We appreciate the support of the Nature Science Foundation of China (62006210, 62001284). In addition, we acknowledge the use of Origin Quantum services for this work. The views expressed are those of the authors and do not reflect the official policy or position of Origin Quantum or any other quantum team.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KNN	K-nearest neighbor
QKNN	Quantum K-nearest neighbor

References

1. Lin, T.-Y.; Dlloar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
3. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
4. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *6*, 1097–1105. [CrossRef]
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large scale image recognition. In Proceedings of the International Conference on Learning Representations, San Diego, NV, USA, 7–9 May 2015.
8. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
9. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
10. Sak, H.; Senior, A.W.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic moduleling. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Lukasz, K.; Polosukhin, I. Attention is all you need. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
12. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing, STOC '96, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219. [CrossRef]
13. Shor, P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
14. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103* 150502. [CrossRef]
15. Jordan, S. The Quantum Algorithm Zoo. Available online: http://math.nist.gov/quantum/zoo/ (accessed on 1 May 2022).
16. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [CrossRef]
17. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef]
18. Chang, W.-L.; Chen, J.-C.; Chung, W.-Y.; Hsiao, C.-Y.; Wong, R.; Vasilakos, A.V. Quantum speedup and mathematical solutions of implementing bio-molecular solutions for the independent set problem on IBM quantum computers. *IEEE Trans. Nanobiosci.* **2021** *20*, 354–376. [CrossRef]
19. Wong, R.; Chang, W.-L. Fast Quantum Algorithm for Protein Structure Prediction in Hydrophobic-Hydrophilic modulel. *J. Parallel Distrib. Comput.* **2022**, *164*, 178–190. [CrossRef]
20. Chang, W.-L.; Chen, J.-C.; Chung, W.-Y.; Hsiao, C.-Y.; Wong, R.; Vasilakos, A.V. Quantum Speedup for Inferring the Value of Each Bit of a Solution State in Unsorted Databases Using a Bio-Molecular Algorithm on IBM Quantum's Computers. *IEEE Trans. Nanobiosci.* **2022**, *21*, 286–293. [CrossRef]
21. Wong, R.; Chang, W.-L. Quantum Speedup for Protein Structure Prediction. *IEEE Trans. Nanobiosci.* **2021**, *20*, 323–330. [CrossRef]

22. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big feature and big data classification. *Phys. Rev. Lett.* **2013**, *113*, 130503. [CrossRef]
23. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
24. Buhrman, H.; Cleve, R.; Watrous, J.; de Wolf, R. Quantum fingerprinting. *Phys. Rev. Lett.* **2001**, *87*, 167902. [CrossRef]
25. Seth Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* **2013**, arXiv:1307.0411.
26. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Inf. Comput.* **2015**, *15*, 316–356. [CrossRef]
27. Dürr, C.; Høyer, P. A Quantum Algorithm for Finding the Minimum. *arXiv* **1996**, arXiv:quant-ph/9607014.
28. Ruan, Y.; Xue, X.; Liu, H.; Tan, J.; Li, X. Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance. *Int. J. Theor. Phys.* **2017**, *56*, 3496–3507. [CrossRef]
29. Li, J.; Lin, S.; Yu, K.; Guo, G. Quantum K-nearest neighbor classification algorithm based on Hamming distance. *Quantum Inf. Process.* **2022**, *21*, 18. [CrossRef]
30. Abu Alfeilat, H.A.; Hassanat, A.; Lasassmeh, O.; Tarawneh, A.S.; Alhasanat, M.B.; Eyal Salman, H.S.; Prasath, V. Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data* **2019**, *7*, 221–248. [CrossRef]
31. Hassanat, A.B. Dimensionality Invariant Similarity Measure. *arXiv* **2014**, arXiv:1409.0923.
32. Getachew, A. Quantum K-medians Algorithm Using Parallel Euclidean Distance Estimator. *arXiv* **2020**, arXiv:2012.11139.
33. Kaye, P.; Mosca, M. Quantum Networks for Generating Arbitrary Quantum States. In Proceedings of the Optical Fiber Communication Conference and International Conference on Quantum Information, Anaheim, CA, USA, 17 March 2001.
34. Giovannetti, V.; Lloyd, S.; Maccone, L. Architectures for a quantum random access memory. *Phys. Rev. A* **2008**, *78*, 52310. [CrossRef]
35. Park, D.K.; Petruccione, F.; Rhee, J.-K.K. Circuit-Based Quantum Random Access Memory for Classical Data. *Sci. Rep.* **2019**, *9*, 3949. [CrossRef]
36. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 40504. [CrossRef]
37. Brassard, G.; Høyer, P.; Mosca, M.; Montreal, A.; Aarhus, B.U.; Waterloo, C.U. Quantum Amplitude Amplification and Estimation. *arXiv* **2000**, arXiv:quant-ph/0005055.
38. Dürr, C.; Heiligman, M.; Høyer, P.; Mhalla, M. Quantum Query Complexity of Some Graph Problems. *SIAM J. Comput.* **2004**, *35*, 1310–1328. [CrossRef]
39. Miyamoto, K.; Iwamura, M.; Kise, K. A Quantum Algorithm for Finding *k*-Minima. *arXiv* **2019**, arXiv:1907.03315.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.