Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.032190
Article

Quantum Fuzzy Support Vector Machine for Binary Classification

Xi Huang[1,2], Shibin Zhang[1,2,*], Chen Lin[1,2] and Jinyue Xia[3]

[1]School of Cybersecurity, Chengdu University of Information Technology, Chengdu, 610225, China
[2]Sichuan Key Laboratory of Advanced Cryptography and System Security, Chengdu, 610225, China
[3]International Business Machines Corporation (IBM), New York, 14201, USA
*Corresponding Author: Shibin Zhang. Email: cuitzsb@cuit.edu.cn
Received: 10 May 2022; Accepted: 01 August 2022

Abstract: In the objective world, how to deal with the complexity and uncertainty of big data efficiently and accurately has become the premise and key to machine learning. Fuzzy support vector machine (FSVM) not only deals with the classification problems for training samples with fuzzy information, but also assigns a fuzzy membership degree to each training sample, allowing different training samples to contribute differently in predicting an optimal hyperplane to separate two classes with maximum margin, reducing the effect of outliers and noise, Quantum computing has super parallel computing capabilities and holds the promise of faster algorithmic processing of data. However, FSVM and quantum computing are incapable of dealing with the complexity and uncertainty of big data in an efficient and accurate manner. This paper research and propose an efficient and accurate quantum fuzzy support vector machine (QFSVM) algorithm based on the fact that quantum computing can efficiently process large amounts of data and FSVM is easy to deal with the complexity and uncertainty problems. The central idea of the proposed algorithm is to use the quantum algorithm for solving linear systems of equations (HHL algorithm) and the least-squares method to solve the quadratic programming problem in the FSVM. The proposed algorithm can determine whether a sample belongs to the positive or negative class while also achieving a good generalization performance. Furthermore, this paper applies QFSVM to handwritten character recognition and demonstrates that QFSVM can be run on quantum computers, and achieve accurate classification of handwritten characters. When compared to FSVM, QFSVM's computational complexity decreases exponentially with the number of training samples.

Keywords: Quantum fuzzy support vector machine (QFSVM); fuzzy support vector machine (FSVM); quantum computing

1 Introduction

Support Vector Machine (SVM) is a machine learning algorithm that can overcome the local minimum and curse of dimensionality in traditional machine learning algorithms. It is based on the rule of Vapnik-Chervonenkis theory and structural risk minimization principle in statistical learning theory. It is a binary classification technique that uses the training dataset to predict an optimal hyperplane in an n-dimensional space. This hyperplane is used to classify new data sets. There are currently several types of SVM, such as twin SVM [1], Gaussian SVM [2], multi-kernel SVM [3] and so on. Furthermore, SVM has been widely applied in a variety of fields, for example, handwritten hindi character recognition [4], face recognition [5], network intrusion detection [6] and breast cancer diagnosis [7], and so on. Although SVM is a common algorithm for classification problems that treats all samples equally and ignores the effect of outliers and noise on the construction of optimal hyperplanes, it fails to perform well when classifying new sets of data with fuzzy information.

To address this issue, some researchers incorporate fuzzy set theory into SVM and propose fuzzy support vector machine (FSVM) algorithms. Inoue et al. [8] proposed the first FSVM algorithm in 2001, which introduces a fuzzy membership function and uses the FSVM classification technique to eliminate the inseparable area. To reduce the influence of noise or outliers, Taiwan scholars Lin et al. [9] combined fuzzy set theory and proposed an FSVM algorithm in 2002, which assigns a fuzzy membership grade to each training sample to make different training samples have different contributions in predicting an optimal hyperplane in an n-dimensional space. Following that, FSVM was rapidly developed. Wang et al. [10] proposed a bilateral-weighted fuzzy support vector machine (B-FSVM), which improves generalization by assigning different memberships to each training sample. Li et al. [11] proposed a regularized monotonic fuzzy support vector machine model that takes into account the various contributions of each training data as well as prior knowledge of monotonicity. Tao et al. [12] proposed an affinity and class probability-based fuzzy support vector machine technique (ACFSVM) that has a better generalization performance when dealing with imbalanced data set classification problems. The key to the FSVM is to construct the fuzzy membership function and choose a suitable fuzzy membership function, which can effectively reduce the effects of outliers and noise when solving the classification problem. Tang et al. [13] proposed a new fuzzy membership function combined with the concept of the k-nearest neighbor algorithm, in which the distance between each training sample and the center of classification, as well as sample affinity, are considered concurrently. Tang [14] proposed a new fuzzy membership function based on the structural information of two classes in the input space and feature space that can effectively distinguish the support vectors and the outliers. To deal with classification problems with outliers or noises, Yang et al. [15] proposed a kernel fuzzy c-means clustering-based fuzzy SVM algorithm (KFCM-FSVM). Currently, the FSVM algorithm has demonstrated superior generalization performance in a variety of fields, including bankruptcy prediction [16], short-term load forecasting [17], and Breast cancer diagnosis [18]. Despite the fact that the FSVM algorithm can deal with a set of data with fuzzy information, it cannot efficiently deal with the complexity and uncertainty of big data.

Due to the high parallelism of quantum computing, some researchers have set out to combine quantum computing and machine learning algorithms and propose quantum machine learning algorithms [19–22]. Rebentrost et al. [23] proposed a quantum support vector machine (QSVM) algorithm in 2014, which uses a swap test [24] to solve inner products and the HHL algorithm [25] to solve matrix inversion. This work demonstrated the feasibility of executing QSVM in a near-term quantum computer by implementing an experimental realization of QSVM for handwriting recognition on a four-qubit NMR test [26]. Ding et al. [27] proposed a quantum-inspired classical algorithm for the least-square support vector machine (LS-SVM) using an improved fast sampling technique, which was inspired by the QSVM algorithm. Lin et al. [28] proposed a novel quantum algorithm for simplifying quantum LS-SVM, as well as a hybrid quantum-classical version for sparse solutions of LS-SVM. Although QSVM is a generalization of the traditional SVM and it can hold the promise of faster algorithmic processing of data, it cannot deal with classification problems for training samples with fuzzy information, so research into QSVM with fuzzy training samples is very meaningful. Moreover, there are many uncertainties in the objective world, how to deal with the ambiguity of big data efficiently and accurately is the premise and key to the SVM algorithm.

Based on the analysis presented above, this paper proposes a novel quantum fuzzy support vector machine for binary classification, paving the way for dealing with the complexity and uncertainty of big data efficiently and accurately. This paper uses the fact that quantum computing can efficiently process big data and FSVM is easy to deal with the complexity and uncertainty problems to research and propose an efficient and accurate QFSVM algorithm. The core idea of the proposed algorithm is to use the least-squares method to convert the quadratic programming problem in FSVM into a linear equation and then solve the linear equation using the HHL algorithm, which can effectively improve the computational complexity of FSVM. The proposed QFSVM algorithm can efficiently deal with classification problems for training samples with fuzzy information, and it can determine whether a sample belongs to the positive or negative class. Moreover, it is applied to the handwritten characters and the experimental results show that the proposed QFSVM algorithm can achieve accurate classification of handwritten characters, and that executing QFSVM in a near-term quantum computer is feasible.

The rest of this paper is structured as follows. A fuzzy support vector machine is introduced in Section 2. The proposed quantum fuzzy support vector machine for binary classification is described in Section 3. Experimental realization of quantum fuzzy support vector machine is given in Section 4. Discussion and conclusion are contained in Section 5.

2 Fuzzy Support Vector Machine

2.1 The Concept of Fuzzy Sets

Classical set theory: crisp set A of X is defined by the characteristic function $f_A(x)$ of set A.

$$f_A(x){:}X \rightarrow 0,\ 1 \tag{1}$$

where $\chi_A(u) = \begin{cases} 0 & u \notin A \\ 1 & u \in A \end{cases}.$

Fuzzy Set Theory [29]: If the set X denotes a collection of objects defined by x, then a fuzzy set A in X can be formulated as a set of ordered pairs:

$$A = \{(x,\ \mu_A(x))|x \in X\} \tag{2}$$

where $\mu_A(x){:}X \rightarrow [0,\ 1]$, and $\mu_A(x)$ is the membership degree of x belonging to X. Membership function $\mu_A(x)$ assigns each element of x to a membership degree between 0 and 1 (include).

2.2 Fuzzy Support Vector Machine

The data points of FSVM are given by

$$S = \{(x_i,\ y_i,\ \mu_i)\}_{i=1}^{l} = (x_1,\ y_1,\ \mu_1),\ (x_2,\ y_2,\ \mu_2),\ \cdots,\ (x_l,\ y_l,\ \mu_l) \tag{3}$$

where $x_i \in R^d$ denotes the characteristic of a training point. $y_i \in \{+1,\ -1\}$ donates the class label of train point x_i, the membership degree $\mu_i \in [0,\ 1]$ donates the degree that the training point x_i belongs to y_i.

The FSVM algorithm, like the SVM algorithm, seeks an optimal separate hyperplane $w^T\phi(x) + b = 0$ that maximizes the margin between two classes. When the sample is nonlinearly separable, finding the optimal hyperplane problem in a high-dimensional feature space is considered the solution to the following optimization problem:

$$\begin{cases} \min\limits_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \mu_i \xi_i \\ s.t.\ y_i(w^T\phi(x_i)+b) - 1 + \xi_i \geq 0 \\ \xi_i \geq 0,\ i = 1,\ 2,\ 3,\ \ldots,\ l \end{cases} \tag{4}$$

where $\xi = (\xi_1,\ \ldots,\ \xi_l)^T$ is a slack variable, $C > 0$ is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error term.

To solve the above optimization problem, we construct the following Lagrangian function:

$$L(w,\ b,\ \alpha,\ \beta,\ \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \mu_i\xi_i + \sum_{i=1}^{l} \alpha_i\left(1 - \xi_i - y_i\left(w^T\phi(x_i) + b\right)\right) - \sum_{i=1}^{l} \beta_i\xi_i \tag{5}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are Lagrangian multipliers corresponding with each training point.

we differentiate Eq. (5) to $w,\ b,\ \xi_i$ and setting the results equal to zero, then we obtain:

$$\begin{cases} \frac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial w} = w - \sum_{i=1}^{l} \alpha_i y_i \phi(x_i) = 0 \\ \frac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0 \\ \frac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial \xi_i} = \mu_i C - \alpha_i - \beta_i = 0 \end{cases} \tag{6}$$

Substituting Eq. (6) into Eq. (4), maximizing α_i, and considering the knowledge of kernel function, the optimization problem is converted as the following quadratic programming problem:

$$\begin{cases} \max\limits_{\alpha_i} \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j K\left(x_i,\ x_j\right) \\ s.t.\ \sum_{i=1}^{l} \alpha_i y_i = 0,\quad 0 \leq \alpha_i \leq s_i C,\ i = 1,\ 2,\ \ldots,\ l \end{cases} \tag{7}$$

where $K\left(x_i,\ x_j\right) = \phi(x_i)^T\phi\left(x_j\right)$ is a kernel function, which makes inner products in the input space mapped into a high dimensional feature space through the nonlinear mapping, to solve the curse of dimensionality. If α meets the condition of $\alpha_i = 0$, the corresponding vector is not a support vector; if $0 < \alpha_i \leq s_i C$, the corresponding vector is a support vector, and only the support vectors are decisive for finding an optimal separate hyperplane.

To solve the quadratic programming problem, we can get $\alpha^* = (\alpha_1,\ \alpha_2,\ \cdots,\ \alpha_l)^T$. And the decision function becomes

$$f(x) = sgn\left(w^T\phi(x) + b\right) = sgn\left(\sum_{i=1}^{l} \alpha_i^* y_i K(x_i,\ x) + b^*\right) \tag{8}$$

where $b^* = y_j - \sum_{i=1}^{l} \alpha_i^* y_i(x_i,\ x), i \in \{i | 0 < \alpha_i \leq s_i C\}$. For a new data x_i, and substituting x_i into the Eq. (8), we can know its class label.

3 Quantum Fuzzy Support Vector Machine for Binary Classification

3.1 Fuzzy Membership Function Based on K-Nearest Neighbor Algorithm

The fuzzy membership function based on K-nearest neighbor algorithm is described in Algorithm 1.

Algorithm 1 The fuzzy membership function based on K-nearest neighbor algorithm

Input: the sample x_i

Output: fuzzy membership degree $\mu(x_i)$

1. Calculate the average distance $d_{average}$ between the sample x_i and its K-nearest neighbor sets:

$$d_{average} = \frac{1}{k}(d_1 + d_2 + \cdots + d_k) \tag{9}$$

2. Calculate d_{max} and d_{min}:

$$\begin{cases} d_{max} = max\left(d_{average}|x_j \in R^n\right) \\ d_{min} = min\left(d_{average}|x_j \in R^n\right) \end{cases} \tag{10}$$

3. Calculate fuzzy membership degree $\mu(x_i)$:

$$\mu\left(x_j\right) = 1 - (1-\theta) \cdot \left(\frac{d_{average} - d_{min}}{d_{max} - d_{min}}\right)^f \tag{11}$$

where θ is a positive number less than 1 and close to 0, and f is the control function.

From Eq. (11), it can be seen that when $d_{average}$ is closer to d_{min}, the value of $\mu(x_i)$ is closer to 1, the sample point x_i is most likely to be noise or outliers. When $d_{average}$ is closer to d_{max}, the probability that the sample point x_i is noise or outliers is θ.

3.2 Quantum Fuzzy Support Vector Machine

The solution to the quadratic programming problem of Eq. (4) has high computational complexity. By introducing the least squares method, Eq. (4) is transformed into the following optimization problem to solve the quadratic programming problem:

$$\begin{cases} \min\limits_{w,b,\xi} \frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{l}\mu_i\xi_i^2 \\ s.t.\ \ y_i(w^T\phi(x_i) + b) = 1 - \xi_i \\ \quad \xi_i \geq 0,\ i = 1,\ 2,\ 3,\ \cdots,\ l \end{cases} \tag{12}$$

To solve the above quadratic programming problem, we construct the following Lagrangian function:

$$L(w,\ b,\ \alpha,\ \beta,\ \xi) = \frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{l}\mu_i\xi_i^2 + \sum_{i=1}^{l}\alpha_i\left(1 - \xi_i - y_i\left(w^T\phi(x_i) + b\right)\right) - \sum_{i=1}^{l}\beta_i\xi_i \tag{13}$$

We differentiate the Lagrangian function $L(w,\ b,\ \alpha,\ \xi)$ in Eq. (13) to $w,\ b,\ \xi_i,\ \alpha_i$, and setting the results equal to zero, we can get:

$$\begin{cases} \dfrac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial w} = w - \sum_{i=1}^{l}\alpha_i y_i \phi(x_i) = 0 \\ \dfrac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial b} = -\sum_{i=1}^{l}\alpha_i y_i = 0 \\ \dfrac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial \xi_i} = \mu_i C - \alpha_i - \beta_i = 0 \\ \dfrac{\partial L(w,\ b,\ \alpha,\ \beta,\ \xi)}{\partial \alpha_i} = 1 - \xi_i - y_i\left(w^T\phi(x_i) + b\right) = 0 \end{cases} \tag{14}$$

After eliminating variables ξ_i, w, the optimization problem in Eq. (12) is converted as the following matrix equation:

$$\begin{bmatrix} 0 & \vec{1}_v^T \\ \vec{1}_v & K + (C\mu_i)^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix} \tag{15}$$

where $K_{ij} = \vec{x}_i^T \vec{x}_j$ is again the kernel matrix, $\vec{1}_v = (1, \cdots, 1)^T$, $\vec{y} = (y_1, \cdots, y_l)^T$, and $\vec{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_l)$. Set $F = \begin{bmatrix} 0 & \vec{1}_v^T \\ \vec{1}_v & K + (C\mu_i)^{-1} I \end{bmatrix}$, the matrix F is $(l+1) \times (l+1)$ dimensional.

FSVM parameters b, $\vec{\alpha}$ are equal to the solution of the following equation:

$$\begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & \vec{1}_v^T \\ \vec{1}_v & K + (C\mu_i)^{-1} I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix} = F^{-1} \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix} \tag{16}$$

Therefore, the SVM parameters are determined schematically by $\left(b, \vec{\alpha}^T\right) = F^{-1}\left(0, \vec{y}^T\right)^T$. The HHL algorithm can be used to obtain FSVM parameters b, $\vec{\alpha}$.

A step-by-step procedure for obtaining the SVM parameters b, $\vec{\alpha}$ is given in Algorithm 2. It should be noted that subscript *a, b,* and *c* represent the quantum state in the ancilla register, clock register, and input register respectively.

Algorithm 2 HHL algorithm for solving FSVM parameters b, $\vec{\alpha}$

Input: $|0, y\rangle$, unitary matrix $\hat{F}$

Output: $|b, \vec{\alpha}\rangle$

1. Initialization: prepare the initial quantum state:

$$|\psi_0\rangle = |0\rangle_a |0\rangle_c |y\rangle_b = |0\rangle_a |0\rangle_c \sum_{j=1}^{l+1} \langle u_j | y\rangle |u_j\rangle_b \tag{17}$$

2. Estimate the eigenvalues λ_j of matrix $\hat{F}$: apply the phase estimation algorithm to $|\psi_0\rangle$:

$$|\psi_0\rangle \rightarrow |\psi_1\rangle = |0\rangle_a \sum_{j=1}^{l+1} \langle u_j | y\rangle |\tilde{\lambda}_j\rangle_c |u_j\rangle_b \tag{18}$$

3. Controlled rotation: apply a controlled rotation R to auxiliary qubit according to the estimated eigenvalues:

$$|\psi_1\rangle \rightarrow |\psi_2\rangle = \sum_{j=1}^{l+1} \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{C}{\tilde{\lambda}_j} |1\rangle \right)_a \langle u_j | y\rangle |\tilde{\lambda}_j\rangle_c |u_j\rangle_b \tag{19}$$

where $C = b^2 + \sum_{k=1}^{l} \alpha_k^2$.

4. Reverse phase estimation: apply the reverse phase estimation algorithm to $|\psi_2\rangle$ and make $|\tilde{\lambda}_j\rangle_c \rightarrow |0\rangle_c$:

$$|\psi_2\rangle \rightarrow |\psi_3\rangle = \sum_{j=1}^{l+1} \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{C}{\tilde{\lambda}_j} |1\rangle \right)_a \langle u_j | y\rangle |0\rangle_c |u_j\rangle_b \tag{20}$$

5. Amplitude amplification: Amplitude amplification to boost the amplitude for $|1\rangle$.

6. Measure: measure the auxiliary qubit. If we get the result $|1\rangle$, the result of the input register is proportional to $\sum_{j=1}^{l+1} \frac{\langle u_j | y\rangle}{\tilde{\lambda}_j} |u_j\rangle$. Otherwise, restart Algorithm 2.

(Continued)

Algorithm 2: (continued)

After the whole process in Algorithm 2, the desired SVM parameters $b, \vec{\alpha}$ are encoded in the amplitudes of the following final state:

$$|b, \vec{\alpha}\rangle = \frac{1}{\sqrt{C}}\left(b|0\rangle + \sum\nolimits_{k=1}^{l} \alpha_k|k\rangle\right) \tag{21}$$

The final state in Eq. (21) can be used for classifying the new data x, and determine whether it belongs to the positive or negative class ($y = +1$ or $y = -1$). The swap test in quantum computing can be used to replace the decision function that is used to categorize new data. Therefore, the decision function of FSVM is given by

$$f(x) = \text{sgn}(\langle u|x\rangle) \tag{22}$$

where $|u\rangle$ and $|x\rangle$ denote the quantum states of training samples and the new data respectively.

The classification process is shown in Algorithm 3.

Algorithm 3 The classification process

Input: $|b, \vec{\alpha}\rangle = \frac{1}{\sqrt{C}}\left(b|0\rangle + \sum\nolimits_{k=1}^{l} \alpha_k|k\rangle\right), |x\rangle$

Output: $y_i \in \{+1, -1\}$

1. By calling the training-data oracle [23], construct $|u\rangle$ and the query state $|x\rangle$:

$$|u\rangle = \frac{1}{\sqrt{N_u}}\left(b|0\rangle|0\rangle + \sum\nolimits_{k=1}^{l} \alpha_k|\vec{x}_k||k\rangle|\vec{x}_k\rangle\right) \tag{23}$$

$$|x\rangle = \frac{1}{\sqrt{N_x}}\left(|0\rangle|0\rangle + \sum\nolimits_{k=1}^{l} |\vec{x}||k\rangle|\vec{x}\rangle\right) \tag{24}$$

where $N_u = b^2 + \sum\nolimits_{k=1}^{l} \alpha_k^2|\vec{x}_k|^2$, $N_x = l|\vec{x}|^2+1$.

2. Performing a quantum swap-test algorithm on $|u\rangle$ and $|x\rangle$.

3. Using an ancilla to construct the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|u\rangle + |1\rangle|x\rangle) \tag{25}$$

4. Using $|\varphi\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ to measure the ancilla, the success probability

$$P = |\langle\psi|-\rangle|^2 = \frac{1}{2}(1 - \langle u|x\rangle) \tag{26}$$

where $\langle u|x\rangle = \frac{1}{\sqrt{N_u N_x}}\left(b + \sum\nolimits_{k=1}^{l} \alpha_k|\vec{x}_k||\vec{x}|\langle\vec{x}_k|\vec{x}\rangle\right)$ corresponds to the decision function. If $P < 1/2$, the new date $|x\rangle$ belongs $y = +1$, Otherwise, the new data $|x\rangle$ belongs $y = -1$.

The flowchart of the proposed QFSVM algorithm can be seen in Fig. 1.

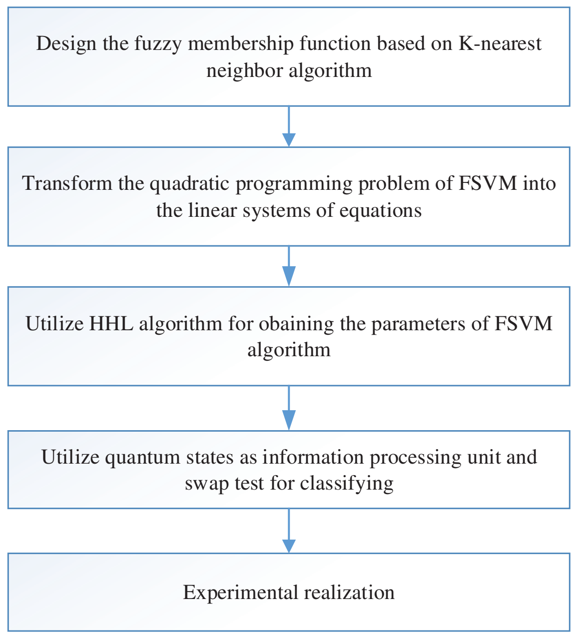


Figure 1: The flowchart of the proposed QFSVM algorithm

For nonlinear FSVM, a kernel function is introduced, a nonlinear mapping $x \mapsto \phi(x)$ into a higher-dimensional vector space is performed, and the kernel function becomes a nonlinear function:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{27}$$

Assuming that the kernel function is a polynomial function $K(x_i, x_j) = (x_i \cdot x_j)^d$, map each sample $|x_i\rangle$ and $|x_j\rangle$ into the d-times tensor product:

$$|\phi(x_i)\rangle = |x_i\rangle^{\otimes d} = |x_i\rangle \otimes \cdots |x_i\rangle \tag{28}$$

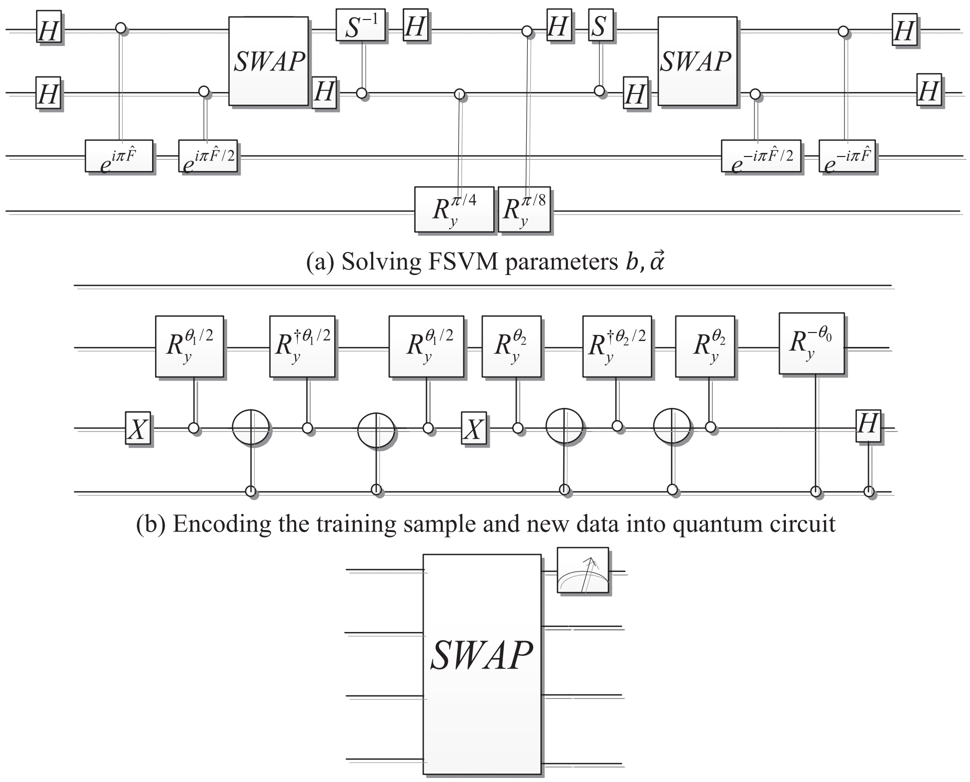
$$|\phi(x_j)\rangle = |x_j\rangle^{\otimes d} = |x_j\rangle \otimes \cdots |x_j\rangle \tag{29}$$

Solving the nonlinear function is equivalent to the inner product of $|\phi(x_i)\rangle$ and $|\phi(x_j)\rangle$:

$$\langle \phi(x_i)|\phi(x_j)\rangle = \langle x_i|x_j\rangle^d \tag{30}$$

The inner product can be calculated by swap test algorithm. Arbitrary polynomial kernels can be constructed using this trick. The polynomial kernel in the original space is converted into a linear hyperplane optimization in the d-times tensor product space.

The whole quantum circuit of QFSVM for binary classification can be seen in Fig. 2 below.



(a) Solving FSVM parameters $b, \vec{\alpha}$

(b) Encoding the training sample and new data into quantum circuit

(c) The classification process

Figure 2: Quantum circuit of QFSVM for binary classification. H denotes the Hadamard gate, SWAP denotes the swap gate, S denotes the phase gate, S^{-1} denotes the inverse phase gate, X denotes the quantum NOT gate. $\theta_i (i = 0, 1, 2)$ is the angle value converted from the feature information of the given datasets

4 Experimental Realization

This section demonstrates the experimental realization of a quantum fuzzy support vector machine for binary classification. The proposed QFSVM algorithm is trained with the handwritten characters "d" and "q", and then eight handwritten characters "d" and "q" chosen from the Modified National Institute of Standards and Technology database (MNIST database) are divided into two-character groups by performing the algorithm. It is worth noting that each handwritten character should be preprocessed, including resizing the pixels and calculating the features. In our experiment, the feature values of the handwritten character are chosen as the horizontal (HR) and vertical ratios (VR), which can be calculated from the pixels in the left (upper) half over the right (lower) half. For the handwritten character picture, calculating its horizontal ratio (HR) and vertical ratio (VR), its angle value $\theta = arccot(HR/VR)$, and the Hermite matrix $\hat{F}$ as the input of quantum circuit in Fig. 2, the class label of new data can be obtained. The training datasets of handwritten characters "d" and "q" can be seen in Tab. 1 below. The testing datasets of handwritten characters "d" and "q" are shown in Tab. 2. The Hermite matrix $\hat{F} = \frac{F}{trF} = \begin{pmatrix} 0.4798 & 0.3613 \\ 0.3613 & 0.5202 \end{pmatrix}$ is calculated by the handwritten characters "d" and "q", and it can be used to obtain the desired SVM parameters b, $\vec{\alpha}$ in Algorithm 2. The recognition results of the handwritten character "d" and "q" are shown in Tab. 3. The test data is classified as d if the amplitude is less than zero. Otherwise, the test data is labeled q. The recognition results in Tab. 3 show that the proposed QFSVM algorithm can achieve accurate classification of handwritten characters when applied to handwritten characters, and that executing QFSVM in a near-term quantum computer is feasible.

Table 1: The training datasets of handwritten characters "d" and "q"

Train datasets	(HR, VR)	θ	Membership degree	Classification label
	(0.6957, 0.5600)	0.6777	0.9	$y = +1$
	(0.7196, 0.9368)	0.9158	0.9	$y = -1$

Table 2: The test datasets of handwritten characters "d" and "q"

Testing datasets	(HR, VR)	θ	Testing datasets	(HR, VR)	θ
	(0.5172, 0.4667)	0.7341		(0.3571, 2.1667)	1.4075
	(0.5658, 0.5063)	0.7300		(0.7813, 0.9000)	0.8559
	(0.8444, 0.4821)	0.5188		(0.8158, 2)	1.1835
	(1.3529, 0.5385)	0.3788		(0.6580, 1.3704)	1.1232

Table 3: The recognition results of handwritten characters "d" and "q"

Handwritten characters	Amplitude	Recognition results	Handwritten characters	Amplitude	Recognition results
	0.0005	d		−0.0195	q
	0.0007	d		−0.0033	q
	0.0074	d		−0.0134	q
	0.0116	d		−0.1160	q

5 Discussion and Conclusion

As an excellent classifier, the train points in (quantum) support vector machines must be specific sets, and the QSVM and SVM cannot deal with the classification problems for training samples with fuzzy information. In comparison to QSVM and SVM, the proposed QFSVM is a generalization of FSVM, and it can not only deal with the training samples with fuzzy information efficiently and accurately, but also assign a fuzzy membership degree to each training sample to reduce the effect of outliers and noise in constructing an optimal hyperplane. In comparison to FSVM, which has the computational complexity of

$O(poly(N,\ M))$, where N is the number of dimensions of the feature space and M is the number of training samples. QFSVM gives exponential speed-up over the FSVM, and the computational complexity is $O(\log(N)\log(M))$.

In conclusion, a quantum fuzzy support vector machine for binary classification is proposed in this paper. It is derived from the fuzzy support vector machine algorithm and can process training samples with fuzzy membership efficiently and accurately, as well as deal with the complexity and uncertainty of big data efficiently and accurately. The proposed algorithm is applied to the handwritten characters, and experimental results show that the proposed QFSVM has good classification accuracy and that executing QFSVM in a near-term quantum computer is feasible. More importantly, it opens up a new path for processing large amounts of data with fuzzy information. In the future, we will focus on the quantum fuzzy support vector machine for multiclass classification and the quantum fuzzy support vector machine for privacy protection.

Funding Statement: This work is supported by the National Natural Science Foundation of China (No. 62076042), the Key Research and Development Project of Sichuan Province (No. 2021YFSY0012, No. 2020YFG0307, No. 2021YFG0332), the Science and Technology Innovation Project of Sichuan (No. 2020017), the Key Research and Development Project of Chengdu (No. 2019-YF05-02028-GX), the Innovation Team of Quantum Security Communication of Sichuan Province (No. 17TD0009), the Academic and Technical Leaders Training Funding Support Projects of Sichuan Province (No. 2016120080102643).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Qing, Y. Kang and A. Wang, "A novel semi-supervised multi-label twin support vector machine," *Intelligent Automation and Soft Computing*, vol. 27, no. 1, pp. 205–220, 2021.

[2] K. S. Bhuvaneshwari, J. Uma, K. Venkatachalam, M. Masud, M. Abouhawwash *et al.,* "Gaussian support vector machine algorithm based air pollution prediction," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 683–695, 2022.

[3] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, T. A. Tabbakh and Z. A. Almusaylim, "Heart failure patient survival analysis with multi kernel support vector machine," *Intelligent Automation & Soft Computing*, vol. 32, no. 1, pp. 115–129, 2022.

[4] S. Dhankhar, M. K. Gupta, F. H. Memon, S. Bhatia, P. Dadheech *et al.,* "Support vector machine based handwritten hindi character recognition and summarization," *Computer Systems Science and Engineering*, vol. 43, no. 1, pp. 397–412, 2022.

[5] S. Guo, S. Chen and Y. Li, "Face recognition based on convolutional neural network and support vector machine," in *IEEE Int. Conf. on Information and Automation (ICIA)*, Ningbo, China, IEEE, pp. 1787–1792, 2016.

[6] Y. Chang, W. Li and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *IEEE Int. Conf. on Computational Science and Engineering (CSE) and IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, IEEE, vol. 1, pp. 635–638, 2017.

[7] G. Jayandhi, J. Leena Jasmine and S. Mary Joans, "Mammogram learning system for breast cancer diagnosis using deep learning svm," *Computer Systems Science and Engineering*, vol. 40, no. 2, pp. 491–503, 2022.

[8] T. Inoue and S. Abe, "Fuzzy support vector machines for pattern classification," in *IJCNN'01. Int. Joint Conf. on Neural Networks. Proc. (Cat. No. 01CH37222)*, Washington, D.C., USA, IEEE, vol. 2, pp. 1449–1454, 2001.

[9] C. F. Lin and S. D. Wang, "Fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, 2002.

[10] Y. Wang, S. Wang and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820–831, 2005.

[11] S. T. Li and C. C. Chen, “A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1713–1727, 2014.

[12] X. Tao, Q. Li, C. Ren, W. Guo, Q. He *et al.,* “Affinity and class probability-based fuzzy support vector machine for imbalanced data sets,” *Neural Networks*, vol. 122, pp. 289–307, 2020.

[13] H. Tang and L. Qu, “Fuzzy support vector machine with a new fuzzy membership function for pattern classification,” in *Int. Conf. on Machine Learning and Cybernetics*, Kunming, China, IEEE, vol. 2, pp. 768–773, 2008.

[14] W. M. Tang, “Fuzzy SVM with a new fuzzy membership function to solve the two-class problems,” *Neural Processing Letters*, vol. 34, no. 3, pp. 209–219, 2011.

[15] X. Yang, G. Zhang, J. Lu and J. Ma, “A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 105–115, 2010.

[16] A. Chaudhuri and K. De, “Fuzzy support vector machine for bankruptcy prediction,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2472–2486, 2011.

[17] J. Yang, Y. Tang and H. Duan, “Application of fuzzy support vector machine in short-term power load forecasting,” *Journal of Cases on Information Technology (JCIT)*, vol. 24, no. 5, pp. 1–10, 2022.

[18] B. Zheng, S. W. Yoon and S. S. Lam, “Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.

[19] S. Lloyd, M. Mohseni and P. Rebentrost, “Quantum algorithms for supervised and unsupervised machine learning,” arXiv preprint arXiv, 1307.0411, 2013.

[20] S. Lloyd, M. Mohseni and P. Rebentrost, “Quantum principal component analysis,” *Nature Physics*, vol. 10, no. 9, pp. 631–633, 2014.

[21] M. Schuld, I. Sinayskiy and F. Petruccione, “Prediction by linear regression on a quantum computer,” *Physical Review A*, vol. 94, no. 2, pp. 022342, 2016.

[22] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe *et al.,* “Quantum machine learning,” *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[23] P. Rebentrost, M. Mohseni and S. Lloyd, “Quantum support vector machine for big data classification,” *Physical Review Letters*, vol. 113, no. 13, pp. 130503, 2014.

[24] H. Buhrman, R. Cleve, J. Watrous and R. D. Wolf, “Quantum fingerprinting,” *Physical Review Letters*, vol. 87, no. 16, pp.167902, 2001.

[25] A. W. Harrow, A. Hassidim and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Physical Review Letters*, vol. 103, no. 15, pp.150502, 2009.

[26] Z. Li, X. Liu, N. Xu and J. Du, “Experimental realization of a quantum support vector machine,” *Physical Review Letters*, vol. 114, no. 4, pp. 140504, 2015.

[27] C. Ding, T. Y. Bao and H. L. Huang, “Quantum-inspired support vector machine,” arXiv preprint arXiv, 1906.08902, 2019.

[28] J. Lin, D. B. Zhang, S. Zhang, T. Li, X. Wang *et al.,* “Quantum-enhanced least-square support vector machine: Simplified quantum algorithm and sparse solutions,” *Physics Letters A*, vol. 384, no. 25, pp. 126590, 2020.

[29] H. J. Zimmermann, “Fuzzy set theory,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 317–332, 2010.