Quantum Information Processing (2023) 22:380
https://doi.org/10.1007/s11128-023-04139-2

Research on support vector machine optimization based on improved quantum genetic algorithm

Fei Wang[1] · Kunlun Xie[1] · Lin Han[2] · Menghui Han[3] · Zeshi Wang[1]

Received: 28 May 2023 / Accepted: 27 September 2023 / Published online: 17 October 2023

Abstract

Support vector machine (SVM) is one of the classical machine learning algorithms. It is widely used and researched for its generalizability and low sample data requirements. However, classical SVM often suffers from problems such as slow solving speed and insufficient accuracy. Quantum genetic algorithm (QGA), which is based on quantum computing principle, is characteristic of faster solving speed and better accuracy than that of classical genetic algorithm, but it is also deficient in easiness of resorting to local optimal solution and insufficient convergence rate in the face of complex problems. In this paper, we propose an improved quantum genetic algorithm (IQGA), which designs a crossover evolution strategy and dynamic rotation angle to avoid local optima. It also adds a quantum convergence gate to address the issue of local convergence. The improved quantum genetic algorithm is applied to SVM parameter optimization, thus its superiority through experimentation and analysis is demonstrated. The results indicate that the model based on improved quantum genetic algorithm support vector machine (IQGA-SVM) has higher prediction accuracy and faster convergence rate compared to back-propagation neural network, classical genetic algorithm support

✉ Lin Han
strollerlin@163.com

Fei Wang
wangfeiwfyx@163.com

Kunlun Xie
aaa11146@outlook.com

Menghui Han
2101566692@qq.com

Zeshi Wang
zeshi1234@163.com

1 School of Mechanics and Safety Engineering, Zhengzhou University, Zhengzhou 450001, Henan, China

2 National Supercomputing Center in Zhengzhou, Zhengzhou 450001, Henan, China

3 College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, Henan, China

vector machine (GA-SVM) and quantum genetic algorithm support vector machine (QGA-SVM) models.

Keywords Quantum genetic algorithm · Dynamic rotation angle · Quantum convergence gate · Support vector machine · Parameter optimization

1 Introduction

Support vector machine (SVM) is a rapidly evolving machine learning algorithm that was officially introduced by Cortes and Vapnik in 1995 [1]. Due to its high generalization ability, good robustness, and capability to achieve excellent results even with relatively small amounts of sample data, support vector machine has gained favor with many researchers. Parameter optimization is a key research area in SVM, as it plays a crucial role in SVM's performance.

In 2006, Huang et al. first proposed the use of genetic algorithms (GA) for parameter optimization in support vector machines [2]. Wang et al. used improved genetic algorithm to simultaneously optimize the penalty parameter, kernel parameter, and loss function of SVM, which was of great significance [3]. By redefining the optimal range of genetic algorithm parameters, Meng et al. proposed an adaptive genetic algorithm, which can help conventional genetic algorithm avoid getting trapped in local optimum solution and guarantee the efficiency of search [4]. In order to improve SVM accuracy and minimize training time, Sajan et al. proposed a genetic algorithm to obtain SVM parameter optimal value and applied it to online voltage stability monitoring [5]. In addition to GA, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are also used in SVM optimization. Gao et al. proposed an improved ACO for optimizing SVM parameters, which provided a new perspective for SVM optimization despite its relatively high complexity and long running time [6]. Hu et al. proposed a particle swarm optimization algorithm, which is applied to the diagnosis of analog circuits, thus improving the accuracy of the diagnosis [7]. Guo et al. proposed an SVM model based on particle swarm algorithm parameter optimization, which significantly improved the fitting accuracy and possessed good generalization ability [8].

In this paper, we come up with an improved quantum genetic algorithm (IQGA) for SVM parameter optimization so as to improve the classification accuracy and efficiency of SVMs. The QGA differs from conventional genetic algorithms in that it uses quantum bits (qubits) to represent individuals, which, in addition to their typical states, can be in a superposition of "0" and "1" states. Therefore, multiple states can be obtained by repeatedly measuring individuals with a relatively small population size, which not only possess superior diversity but also exhibit better convergence. However, traditional quantum genetic algorithms themselves are prone to local extremes and may fall into local optima. Furthermore, the fixed rotation angle adjustment strategy adopted in the algorithm for updating quantum rotation gates is inefficient and also poses a risk of getting stuck in local optima. Therefore, this paper proposes a crossover and evolution strategy while also incorporating an adaptive dynamic rotation angle to improve efficiency and address the problem of the algorithm being prone to local optima. To prevent the probability amplitude α and β of qubits from prematurely converging to 0 or 1 and causing the algorithm to become stuck in local optima, a quantum convergence gate H_ε is used to correct the magnitude of the qubit probability that gets too close to 0 or 1 [9]. Finally, the parameter optimization and classification experiments of support vector machine are carried out by using the IQGA.

To validate the performance of IQGA-SVM, it is used to classify the benchmark dataset and the results are compared with QGA-SVM, GA-SVM and back-propagation neural network. There are three steps to classification: firstly, selecting and preprocessing data samples, and splitting them into training sets and test sets; secondly, training the SVM using the training sets; and finally, testing the trained model to obtain classification results.

The remainder of the paper is as follows: Sects. 2 and 3 introduce the fundamental concepts of SVM and QGA, respectively. The IQGA in detail comes in Sect. 4. Section 5 illustrates the application of IQGA to SVM parameter optimization. In Sect. 6, we demonstrate the learning performance of IQGA-SVM in the classification of benchmark datasets. Finally, we summarize the research findings and future work of this study in Sect. 7.

2 Support vector machine

SVM is a classification and regression algorithm widely used in statistical learning. And it is a type of generalized linear classifier which is suitable for small sample data and classifies it binary. The idea is to maximize the margin, which can be converted into a convex quadratic programming problem [10, 11].

The SVM has the following advantages, such as efficient high-dimensional data processing, good robustness and good generalization ability on the small sample data set. Therefore, it has a wide range of applications in many fields, especially in classification and regression problems. SVM can be used for financial tasks such as credit scoring, risk assessment and stock market forecasting in the economic field. And it can also be used in medical image analysis, such as disease detection, cancer diagnosis and medical image classification in the medical field. This section will introduce the fundamental principles of SVM.

2.1 Linear support vector machine

If the samples are linearly separable, which means they can be separated into two categories by a separation hyperplane: $\omega \cdot x + b = 0$,then the maximum classification margin problem of SVM optimal hyperplane can be transformed into the optimal solution to the linear constraint problem:

$$\begin{aligned} &\max_{\omega, b} \frac{2}{\omega} \Leftrightarrow \min_{\omega, b} \frac{1}{2}\omega, \\ &\text{s.t. } y_i(\omega_i x + b) \geq 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

By introducing Lagrange multipliers, the convex quadratic programming problem can be turned into a constrained optimization problem for solving the functional by using the duality principle:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j,$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} y_i \alpha_i = 0 \\ \alpha_i \geq 0 \end{cases}, \tag{2}$$

of which $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the Lagrangian multiplier.

This is a quadratic programming problem under inequality constraints and has a unique solution. According to the KKT conditions, only a small subset of solution variables (usually a tiny fraction) will be non-zero. The samples corresponding to these solutions that are not zero are support vectors.

If α^* is the optimum solution to the convex quadratic programming problem, then the optimum solution to the original problem is:

$$\omega^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i. \tag{3}$$

$$b^* = y_i - \sum_{i=1}^{n} \alpha_i^* y_i x_i^{\mathrm{T}} x_j. \tag{4}$$

2.2 Nonlinear support vector machine

Nonlinear problems are commonly encountered when dealing with practical problems. With the introduction of kernel functions and the use of the idea of nonlinear transformation of inner product functions, it is possible to map samples in a linear space to a high-dimensional feature space via a nonlinear transformation, thereby constructing the optimal separation hyperplane in high-dimensional space and achieving linear separability [12, 13]. If there exists a mapping function $\phi : x \to \phi(x)$ that can represent the mapping from high-dimensional space to low-dimensional space, then there will be a separation hyperplane $\omega^{\mathrm{T}}\phi(x) + b$. Similarly, the problem of maximizing the classification margin can be formulated as finding the optimal solution to a linearly constrained problem, expressed as

$$\max_{\omega, b} \frac{2}{\omega} \Leftrightarrow \min_{\omega, b} \frac{1}{2} \omega,$$
$$\text{s.t. } y_i(\omega_i \phi(x) + b) \geq 1, \quad i = 1, 2, \ldots, n. \tag{5}$$

By introducing Lagrange multipliers, the convex quadratic programming problem can be turned into a constrained optimization problem for solving the functional by using the duality principle:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^{\mathrm{T}} \phi(x_j),$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} y_i \alpha_i = 0 \\ \alpha_i \geq 0 \end{cases}, \tag{6}$$

of which $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the Lagrangian multiplier.

To solve the above equation, which involves the formula: $\phi(x_i)^{\mathrm{T}}\phi(x_j)$, it is very difficult to directly calculate the internal product mapped to a high-dimensional space. If we consider a function:

$$\kappa\left(x_i, x_j\right) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^{\mathrm{T}} \phi\left(x_j\right). \tag{7}$$

Complex calculations can be avoided if the inner product between x_i and x_j in a high-dimensional feature space equals the value of the function $\kappa\left(x_i, x_j\right)$ obtained in the original space. Therefore, $\kappa\left(x_i, x_j\right)$ is kernel function.

Because RBF kernel functions have the advantages of broad dimension mapping, few parameter requirements and relatively simple computations, it is mainly used in SVM research [14, 15]. The RBF kernel function can be expressed as:

$$\kappa\left(x_i, x_j\right) = \exp\left(-g\|x_i - x_j\|\right)^2, \tag{8}$$

which g is kernel parameter.

2.3 Parameter optimization of SVM

Support vector machines have two critical parameters: the penalty parameter C and the kernel function g. The penalty parameter C describes how much the model tolerates error. The larger the penalty parameter C, the more the model is unwilling to accept errors, and the more severe the punishment for misclassified samples. This, in turn, may lead to overfitting. On the contrary, a smaller penalty parameter C indicates that the model is more tolerant of errors, resulting in lighter punishment for misclassified samples. However, this may lead to underfitting. If the penalty parameter C is too large or too small, it will lead to a poor generalization effect of the model, so as to fail to achieve the effect and purpose of learning. In summary, the penalty parameter C is to strike a balance between model accuracy and model complexity. Therefore, it is particularly important to choose an appropriate penalty parameter C.

Similarly, the kernel parameter g is a parameter of the RBF function after selecting it as the kernel function. Implicitly determines the data distribution after mapping to the new feature space. The larger the kernel parameter g, the fewer support vectors. On the contrary, smaller values of the kernel parameter g lead to more support vectors. The number of support vectors affects the speed of training and prediction.

The selection of the kernel function g has a significant impact on SVM performance, as does the penalty parameter C. In application, it is necessary to select the appropriate parameter combination according to the specific problem in order to obtain the best classification effect. Therefore, IQGA is chosen to automatically find the optimal parameters of SVM, and the feasibility of this method is proved in this paper.

3 Quantum genetic algorithm

Han et al. first proposed the quantum genetic algorithm [2]. The QGA is an evolutionary algorithm based on quantum computing, which combines the advantages of quantum computing and GA. An intuitive way to understand genetic algorithms: GA is a class of randomized search algorithms that learn from the mechanisms of natural selection and inheritance in biology. It simulates the evolution process of an artificial population. A group of individuals is retained in each iteration through selection, crossover and mutation. Repeating this process, after several generations of population evolution, its fitness reaches an approximately optimal state in the ideal case.

Unlike classical GA, QGA encodes chromosomes using qubits which can enrich the diversity of populations. QGA takes the current population's optimal individual as the target and evolves the current individuals toward the optimal individual through quantum gate operations instead of by selection or crossover. After each iteration update, the quantum state collapses into a definite state. QGA, with its unique encoding scheme and iterative operations, exhibits a faster convergence rate and higher accuracy. This section will introduce the fundamental principles of QGA.

3.1 Quantum chromosome

The equation $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ can be used to represent quantum states, where α and β represent the probability amplitudes of the quantum state. That means, there is a probability of α that the quantum state collapses into the '0' state, and there is a probability of β that the quantum state collapses into the '1' state, satisfying the normalization condition.

In QGA, $[\alpha, \beta]^{\mathrm{T}}$ can represent a qubit. A quantum system with m qubits can represent a set of chromosomes, which corresponds to an individual:

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array}\right] \tag{9}$$

3.2 Quantum rotation gate

QGA updates the population through quantum gate operations, causing current individuals to evolve toward the optimal individuals[16, 17]. The quantum rotational gate matrix can be expressed as follows:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{10}$$

Performing a quantum gate operation on the quantum state $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ yields the following expression:

$$\left|\varphi'\right\rangle = R(\theta) \times |\varphi\rangle = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix}, \tag{11}$$

where $|\varphi\rangle$ and $|\varphi\prime\rangle$ represent quantum states before and after the operation of the quantum gate, respectively. Figure 1 is a diagram of quantum rotation gate.

The specific adjustment strategy for determining rotation angle of the quantum gate and direction are shown in Table 1. Moreover, $f(x)$ and $f(b)$ in Table 1 represent the fitness values of the current individual and the best individual, respectively. The update strategy of the algorithm is to compare $f(x)$ and $f(b)$, then the individual will evolve toward the direction of higher fitness, finally the rotation direction will be determined by $s(\alpha_i\beta_i)$, while $\Delta\theta_i$ determines the angle size of the quantum gate rotation[9].

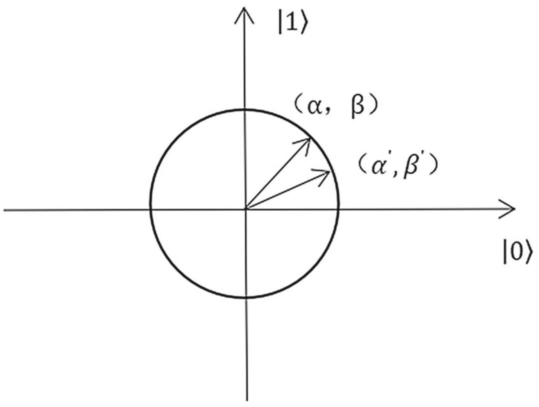


Fig. 1 Quantum rotation gate

Table 1 Quantum rotation gate update strategy table

x_i	b_i	$f(x) \geq f(b)$	$\Delta\theta_i$	$s(\alpha_i\beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	ε	+ 1	− 1	0	± 1
0	1	True	ε	− 1	+ 1	± 1	0
1	0	False	ε	− 1	+ 1	± 1	0
1	0	True	ε	+ 1	− 1	0	± 1
1	1	False	0	0	0	0	0
1	1	True	0	0	0	0	0

4 Improved quantum genetic algorithm

Traditional QGA can easily produce local extreme value and be trapped into local optimal solution [18, 19]. In addition, the efficiency of quantum gate updating operations using a fixed rotation angle adjustment strategy is low, which further increases the risk of getting trapped in local extreme values. To address these issues, this paper proposes the following optimization strategies:

1. Crossover and evolution strategy.
2. Dynamic rotation angle.
3. Adds a quantum convergence gate.

4.1 Crossover and evolution strategy

In the later stages of a quantum genetic algorithm, there is often a large number of individuals gathered around a solution, without searching the global space, leading to the inability to evolve new individuals[20]. This increases the risk of getting trapped in local optimal solutions when solving complex problems.

To address the aforementioned issue, a novel crossover and evolution strategy is proposed: dividing the population into N subpopulations with equal numbers of individuals and performing evolution operations on each subpopulation separately. In the later stage of evolution, randomly selected individuals from different subpopulations are exchanged through crossover operations, followed by continued iterative evolution.

The idea behind the crossover and evolution strategy is to introduce new individuals to the population by exchanging them when it becomes trapped in local optima and is unable to produce any new individuals. Moreover, the newly introduced individuals have undergone a certain number of evolution operations and do not affect the efficiency of subsequent updates with evolutionary processes.

When initializing individuals in the quantum genetic algorithm, the probability amplitudes α and β are usually set to $\frac{1}{\sqrt{2}}$. In this paper, an improved algorithm sets the initial probability amplitudes as random numbers within the range of (0, 1), α and β satisfy the normalization condition. This enhances the diversity of the initial population.

4.2 Dynamic rotation angle

The QGA evolves the quantum state of the current individual toward the quantum state of the optimal individual through quantum rotation gates. In classical quantum genetic algorithms, fixed rotation angles are used for updating. In earlier stages, too small rotation angles can affect the convergence rate. While in later stages, too large rotation angles may also miss the optimum solution.

To address the above-mentioned problem, a dynamic rotation angle strategy is proposed: the size of the rotation angle that determines the convergence rate is determined by the fitness values of the current individual and the optimum individual. In the early stages, when there is a large difference between the fitness values of the two individuals, the rotation angle should be increased to accelerate the individual's convergence to the optimal solution; in the later stage, there is a small difference in fitness between the two, the rotation angle should be decreased to prevent excessive rotation and miss the optimal solution.

Optimization strategy for dynamic rotation angle $\Delta\theta_i$:

$$\Delta\theta_i = k(\theta_{\max} - \theta_{\min}) + \theta_{\min}, \tag{12}$$

$$k = \arcsin\left(\frac{f_{\text{opt}} - f_{\text{cur}}}{f_{\text{opt}}}\right), \tag{13}$$

else, f_{opt} is the optimal individual fitness, while f_{cur} is the current individual fitness. Whereas $\theta_{\max}$ and $\theta_{\min}$ are the maximum and minimum angles of rotation.

In the early stage of evolution, there is a large difference between individuals, and $\frac{f_{\text{opt}}-f_{\text{cur}}}{f_{\text{opt}}}$ tends to be 1 and $k = \arcsin(\frac{f_{\text{opt}}-f_{\text{cur}}}{f_{\text{opt}}})$ approaches $\frac{\pi}{2}$. As a result, it can be inferred that the rotation angle approaching $\Delta\theta_i$ is a slightly larger angle than $\theta_{\max}$, which is beneficial to the current solution to quickly approach the optimal solution. With evolution, the current individual evolves toward the optimal individual, and $\Delta\theta_i$ decreases accordingly. In the later stage of evolution, there is a small difference between individuals, and $\frac{f_{\text{opt}}-f_{\text{cur}}}{f_{\text{opt}}}$ tends to be 0 and $k = \arcsin(\frac{f_{\text{opt}}-f_{\text{cur}}}{f_{\text{opt}}})$ also tends to be 0. As a result, it can be inferred that the rotation angle $\Delta\theta_i$ approaches $\theta_{\min}$. Moreover, as the inverse trigonometric function is defined in the domain of (0,1), its value of the independent variable decreases as the value of the function decreases, and its derivative is also directly related to the value of the function. Therefore, in the later stage, as the rotation angle decreases, the rate of decrease also slows down, effectively preventing excessive rotation and facilitating thorough exploration of the solution space, leading to the discovery of the global optimal solution.

4.3 Quantum convergence gate

Since the initial individual probability amplitude set in this paper is a random number between (0,1), it may be close to 0 or 1. To prevent premature convergence to local optima and loss of global optima, a quantum convergence gate is introduced to correct the amplitudes that are too close to 0 or 1. The quantum convergence gate H_ε:

$$[\alpha_i{\prime}\beta_i{\prime}] = H_\varepsilon\big(\alpha_i,\ \beta_i,\ \theta_i\big). \tag{14}$$

For the equation $\left[\alpha_i''\beta_i''\right]^{\text{T}} = U\big(\theta_i[\alpha_i{\prime}\beta_i{\prime}]^{\text{T}}\big)$.

1. If $\left|\alpha_i''\right|^2 \le \varepsilon$ and $\left|\beta_i''\right|^2 \ge 1-\varepsilon$, then $[\alpha_i{\prime}\beta_i{\prime}]^{\text{T}} = \left[\sqrt{\varepsilon}\sqrt{1-\varepsilon}\right]^{\text{T}}$.
2. If $\left|\alpha_i''\right|^2 \ge 1-\varepsilon$ and $\left|\beta_i''\right|^2 \le \varepsilon$, then $[\alpha_i{\prime}\beta_i{\prime}]^{\text{T}} = \left[\sqrt{1-\varepsilon}\sqrt{\varepsilon}\right]^{\text{T}}$.
3. Else $[\alpha_i{\prime}\beta_i{\prime}]^{\text{T}} = \left[\alpha_i''\beta_i''\right]^{\text{T}}$.

4.4 Function testing

In order to test the feasibility of the above enhanced strategy and verify its superiority compared to classical genetic algorithms, we conduct experiments using IQGA and QGA to solve complex binary functions in this paper.

Complex binary functions:

$$f(x,\ y) = x\sin(4\pi x) + y\sin(20\pi y),\ \begin{cases} -3.0 \le x \le 12.1 \\ 4.1 \le y \le 5.8 \end{cases} \tag{15}$$

Conventional optimization algorithms are highly susceptible to getting trapped in local optima or oscillating among different local optima. However, as we can observe from Fig. 2, this nonlinear function is distributed with numerous local extreme values in the given range, so it is suitable for verifying the superiority of quantum genetic algorithm.

The maximum value of this function approaches 17.3503 within its domain. Figure 3 shows the process of two algorithms solving complex binary functions and reveals that compared with the traditional quantum genetic algorithm, the IQGA exhibits faster solving speed and is less prone to get trapped in local optima. Table 2 shows the fitness values of the two algorithms and the minimum number of iterations to obtain the optimal solution when running 20 times separately until the end of iteration (or obtaining the optimum solution). The fitness values of IQGA tend to be 17.3503 when the iteration ends (or the optimum solution is obtained), indicating that the optimal solution has been obtained. The average fitness value of QGA at the end of iteration (or obtaining the optimum solution) was 17.2677, and the worst fitness value was

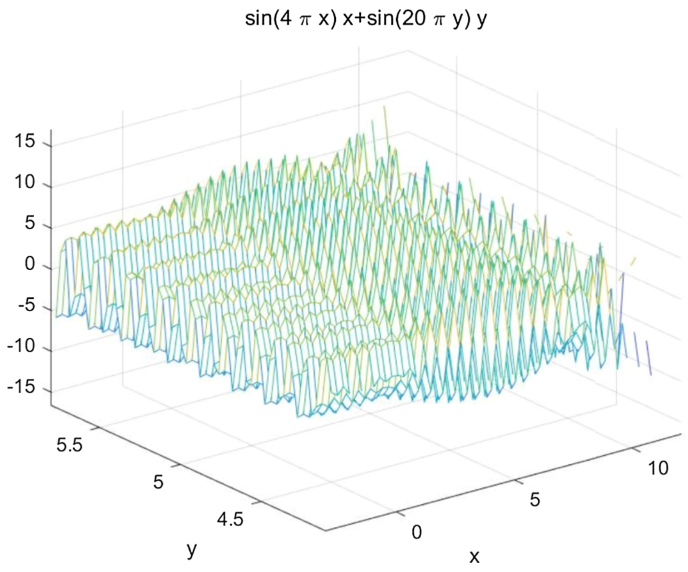


Fig. 2 Grid diagram of the function

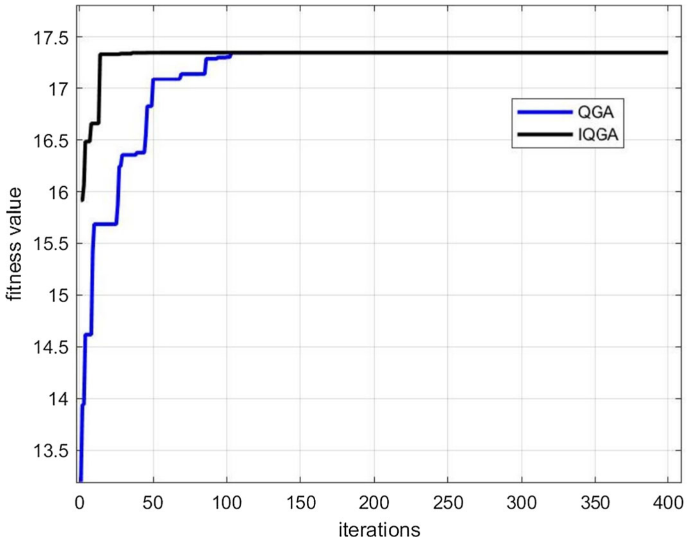


Fig. 3 The comparison process of the two algorithms

Table 2 Comparison of the results of the two algorithms

	Optimal value	Worst value	Mean value	Iterations
IQGA	17.3503	17.3503	17.3503	57
QGA	17.3503	16.9501	17.2677	108

16.9501. Therefore, it can be concluded that IQGA is significantly better than QGA in terms of solving accuracy. IQGA converges to the optimum solution with a minimum number of iterations of 57, whereas QGA converges with 108 iterations, which further indicates that IQGA has better solving speed. In a conclusion, the test shows that IQGA has better accuracy and convergence rate.

5 SVM based on IQGA

As mentioned in Sect. 2.2, the RBF kernel function has advantages in wide mapping dimension, less parameters, and relatively simple computations. Therefore, it is mainly used in the study of SVM. The performance of the SVM depends on the selection of the kernel parameter g and penalty parameter C of the RBF kernel function, and the quality of these parameters significantly affects the accuracy of the algorithm. The performance of SVM depends on the choice of kernel parameter g and penalty parameter C of the RBF kernel function. Parameter selection exits a significant impact on the algorithm. Furthermore, the two parameters are randomly selected within a positive range, and there are many possible combinations of these parameters [21]. If traditional trial-and-error methods are used for parameter tuning, it would greatly affect efficiency and may lead to overfitting or underfitting. Therefore, genetic algorithms are often used for parameter optimization in SVM [22].

To verify the performance of the IQGA, GA, QGA, and IQGA were used to optimize the parameters g and C in the SVM modeling process. The optimal combination of the parameters g and C selected by GA-SVM, QGA-SVM, and IQGA-SVM was used to establish models, which were then tested for performance using benchmark datasets.

In this paper, IQGA is used to help SVM find the optimal combination of g and C parameters, and its specific steps are shown in Fig. 4:

Step 1: Initialize the chromosome (individual) number $P(t) = \{P^t{}_1, P^t{}_2, P^t{}_3, \ldots, P^t{}_n\}$ of quantum genetic algorithm population, where n represents the population size, and $P^t{}_n$ represents a single individual in generation t population.
Step 2: Measure all individuals.
Step 3: Compute the fitness of all the individuals.
Step 4: Record the best individual and its fitness in the current generation of evolution.
Step 5: Judge whether the current population of optimal individuals meets the target requirements. If the optimal parameters of SVM are found or the maximum number of iterations is reached, IQGA will be terminated and the results will be output to SVM. Otherwise, use the optimum individual and its fitness value as the next generation population to continue execution.
Step 6: Use the improved dynamic rotation angle quantum gate for updating.
Step 7: Judge whether half of the maximum iteration number has been reached, and if so, exchange some individuals with other populations. Otherwise, go back to Step 5.

6 Simulation

6.1 Benchmark datasets

The benchmark datasets used in this paper are iris dataset, blood dataset and seeds dataset. All datasets are collected from the UCI Machine Learning Repository. The data structures of benchmark datasets are shown in Table 3:

6.2 Calculation results and analysis

6.2.1 Iris dataset

The optimal combination of g and C parameters selected by GA-SVM, QGA-SVM, and IQGA-SVM was used to establish a model, which was then tested for performance using the iris dataset. In this experiment, the population size is chosen as 40 and evenly divided into two sub-populations, and the maximum evolutionary iteration number is set to 500. At half of the maximum iteration number, 1/4 of individuals in the each sub-population were randomly selected for exchange. We show the fitness curves in Fig. 5, use three methods to make five predictions on the benchmark dataset, and select the most frequent data group for display, as shown in Fig. 6.

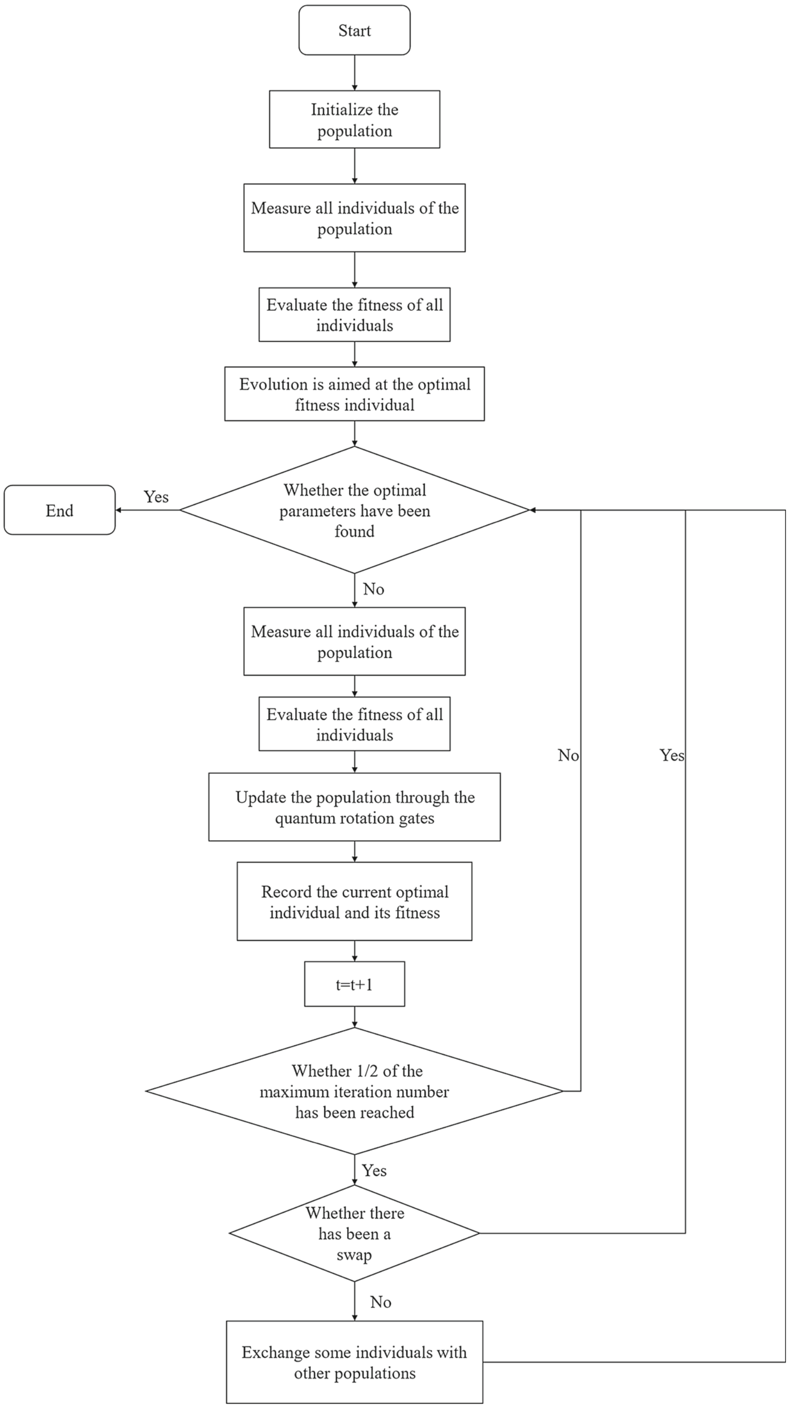

Start
Initialize the population
Measure all individuals of the population
Evaluate the fitness of all individuals
Evolution is aimed at the optimal fitness individual
Whether the optimal parameters have been found
Yes
End
No
Measure all individuals of the population
Evaluate the fitness of all individuals
Update the population through the quantum rotation gates
Record the current optimal individual and its fitness
t=t+1
Whether 1/2 of the maximum iteration number has been reached
No
Yes
Yes
Whether there has been a swap
No
Exchange some individuals with other populations

Fig. 4 Algorithm procedure

Table 3 Benchmark datasets

Datasets	No. of real attributes	No. of classes	No. of instances	Training instances	Test instances
Iris	4	3	150	105	45
Blood	4	2	721	498	223
Seeds	7	3	210	150	60

Figure 5a shows the maximum fitness curves of the three algorithms, while Fig. 5b, c shows the average fitness curves without fitting and after fitting. Figure 6 shows the prediction results of the three algorithms.

From Fig. 5a, it can be observed that the maximum fitness of 95.2381% is reached in the 195th generation of the iteration, and the optimal combination of prediction model parameters selected at this time is: Best $c = 12.8615$ and Best $g = 33.2836$. As shown in Fig. 6a, the prediction accuracy at this point is 97.7778%.

From Fig. 5a, it can be observed that the maximum fitness of 96.1905% is reached in the 37th generation of iteration, and the optimal combination of prediction model parameters selected at this time is: Best $c = 7.9179$ and Best $g = 0.29326$. As shown in Fig. 6b, the prediction accuracy at this point is 97.7778%.

From Fig. 5a, it can be observed that the maximum fitness of 98.0952% is reached in the 33rd generation of iteration, and the optimal combination of prediction model parameters selected at this time is: Best $c = 18.0841$ and Best $g = 0.097752$. As shown in Fig. 6c, the prediction accuracy at this point is 100%.

Comparing the three algorithms, it can be observed that the prediction accuracy of QGA-SVM is 2.222% higher than that of GA-SVM and IQGA-SVM. Moreover, the maximum fitness of IQGA-SVM is 1.9047% higher than that of QGA-SVM, and when IQGA-SVM reaches its maximum fitness, the evolution generation is 4 generations less than that of QGA-SVM. Therefore, we can conclude that IQGA-SVM has a better prediction accuracy and a faster convergence rate.

6.2.2 Blood dataset

The optimal combination of g and C parameters selected by GA-SVM, QGA-SVM, and IQGA-SVM was used to establish a model, which was then tested for performance using the blood dataset. In this experiment, the population size is chosen as 40 and evenly divided into two sub-populations, and the maximum evolutionary iteration number is set to 500. At half of the maximum iteration number, randomly select 1/4 of individuals in the each sub-population for exchange. We show the fitness curves in Fig. 7, use three methods to make 5 predictions on the benchmark dataset, and select the most frequent data group for display, as shown in Fig. 8.

Figure 7a shows the maximum fitness curves of the three algorithms, while Fig. 7b, c shows the average fitness curves without and after fitting. Figure 8 shows the prediction results of the three algorithms.

Fig. 5 Fitness curves of three algorithms in iris dataset

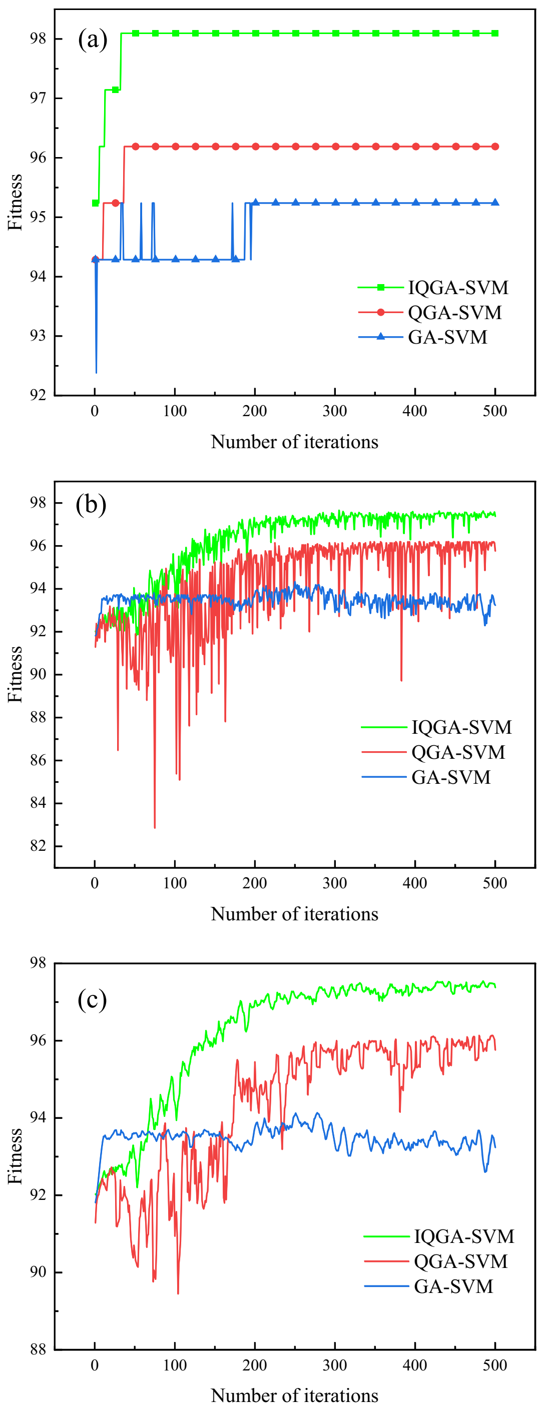
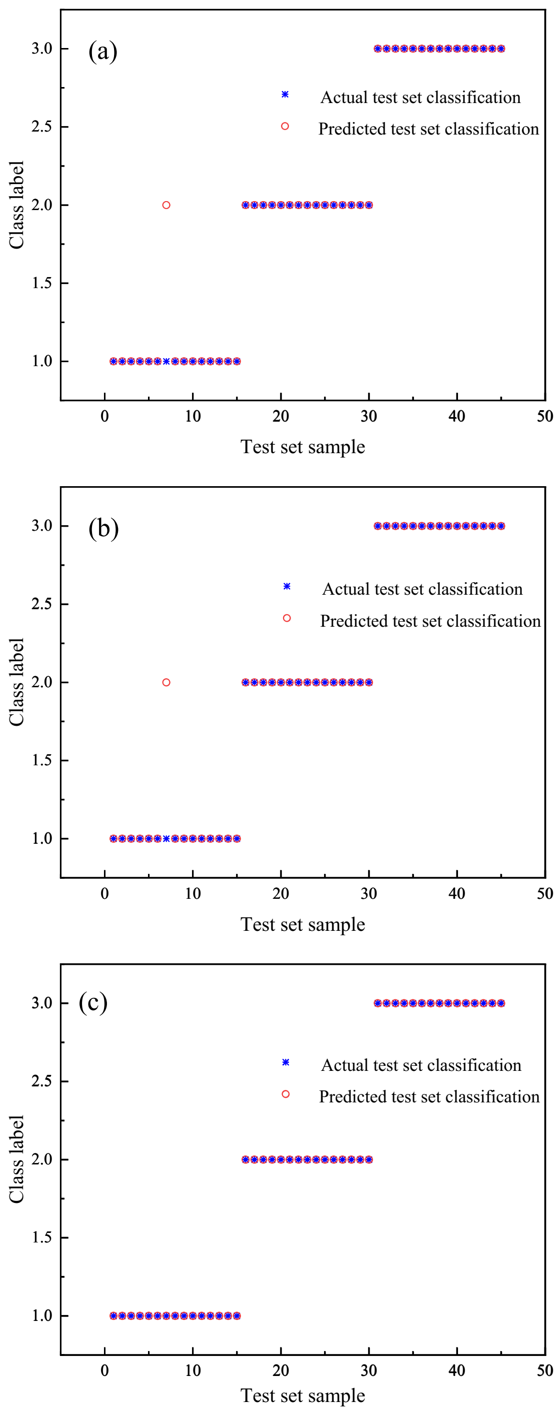


Fig. 6 The prediction results of three algorithms in iris dataset



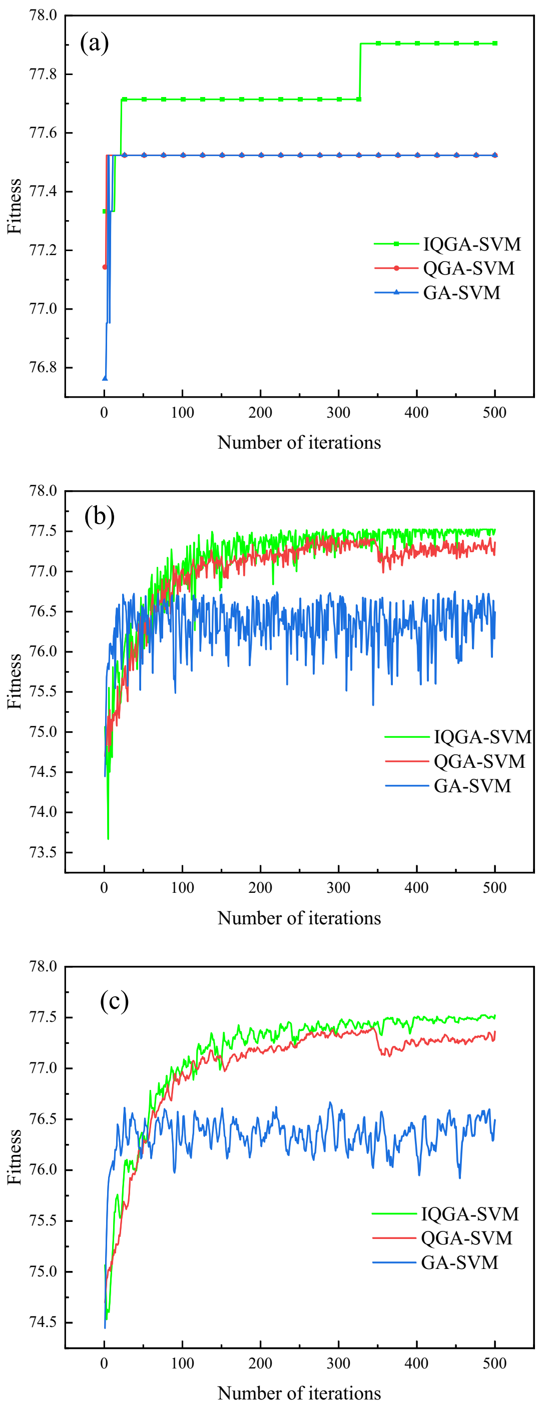


Fig. 7 Fitness curves of three algorithms in blood dataset

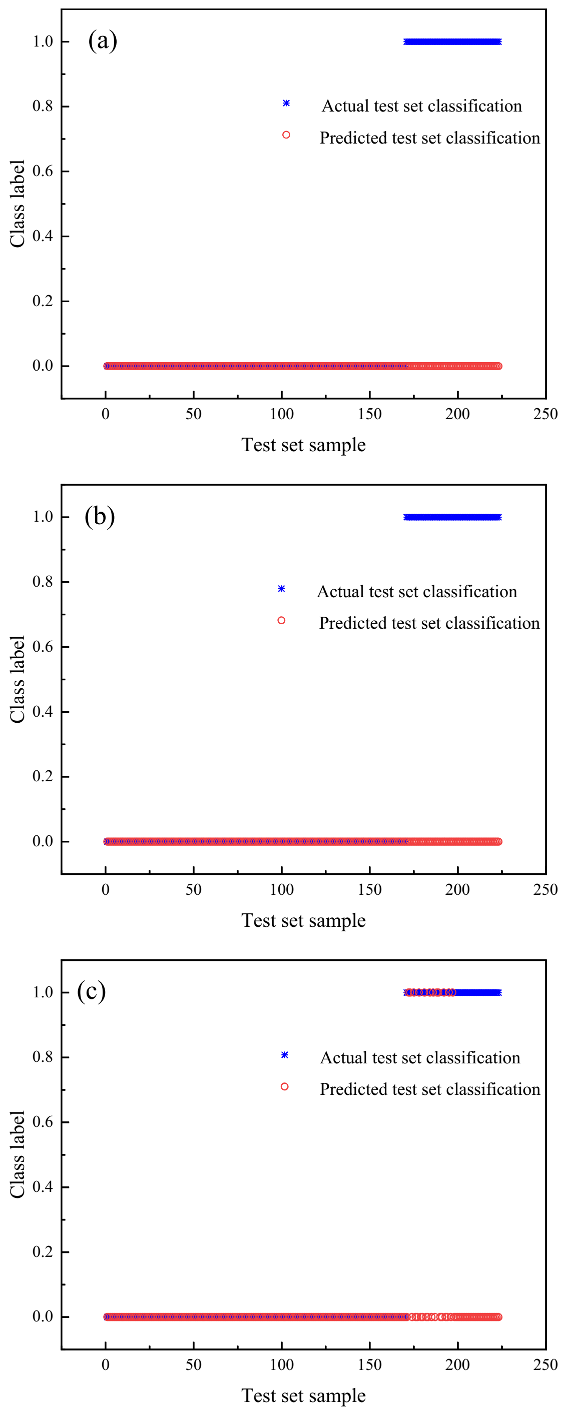


Fig. 8 The prediction results of three algorithms in blood dataset

From Fig. 7a, it can be observed that the maximum fitness of 77.5238% is reached in the 5th generation of iteration and the optimal combination of prediction model parameters at this point is: Best $c = 15.814$ and Best $g = 1.2342$. As shown in Fig. 8a, the prediction accuracy at this point is 76.2332%.

From Fig. 7a, it can be observed that the maximum fitness of 77.5238% is reached in the 2nd generation of iteration and the optimal combination of prediction model parameters at this point is: Best $c = 64.6139$ and Best $g = 0.48876$. As shown in Fig. 8b, the prediction accuracy at this point is 76.2332%.

From Fig. 7a, it can be observed that the maximum fitness of 77.9048% is reached in the 341th generation of iteration, and the optimal combination of prediction model parameters selected at this time is: Best $c = 6.7449$ and Best $g = 36.1681$. As shown in Fig. 8c, the prediction accuracy at this point is 81.6143%.

Comparing three algorithms, it can be seen that GA-SVM achieved high fitness in the 5th generation and QGA-SVM in the 2nd generation (early stages of the algorithm), but subsequent iterations did not result in higher fitness. The algorithm was trapped in a local optimum and unable to escape. On the other hand, IQGA-SVM jumped out of local optima in the 341th generation (middle stage of the algorithm), confirming advantages of the IQGA in escaping local optima. The maximum fitness of IQGA-SVM is 0.381% higher than that of QGA-SVM and GA-SVM, and the prediction accuracy of IQGA-SVM is 5.3811% higher than that of QGA-SVM and GA-SVM. Therefore, we can conclude that IQGA-SVM has a better prediction accuracy and the advantage of breaking through local optimal solutions.

6.2.3 Seeds dataset

The optimal combination of g and C parameters selected by GA-SVM, QGA-SVM, and IQGA-SVM was used to establish a model, which was then tested for performance using the seeds dataset. In this experiment, the population size is chosen as 40 and evenly divided into two sub-populations, and the maximum evolutionary iteration number is set to 500. At half of the maximum iteration number, 1/4 of individuals in the each sub-population were randomly selected for exchange. We show the fitness curves in Fig. 9, use three methods to make 5 predictions on the benchmark dataset, and select the most frequent data group for display, as shown in Fig. 10.

Figure 9a shows the maximum fitness curves of the three algorithms, while Fig. 9b, c shows the average fitness curves without fitting and after fitting. Figure 10 shows the prediction results of the three algorithms.

From Fig. 9a, it can be observed that the maximum fitness of 98% is reached in the 76th generation of iteration, and the optimal combination of prediction model parameters selected at this time is: Best $c = 14.527$ and Best $g = 4.5343$. As shown in Fig. 10a, the prediction accuracy at this point is 81.6667%.

From Fig. 9a, it can be observed that the maximum fitness of 98% is reached in the 28th generation of iteration, and the optimal combination of prediction model parameters selected at this time is: Best $c = 0.1955$ and Best $g = 11.437$. As shown in Fig. 10b, the prediction accuracy at this point is 81.6667%.

From Fig. 9a, it can be observed that the maximum fitness is achieved at 29th generation of iteration, with a value of 98.6667%. The optimal parameter combination

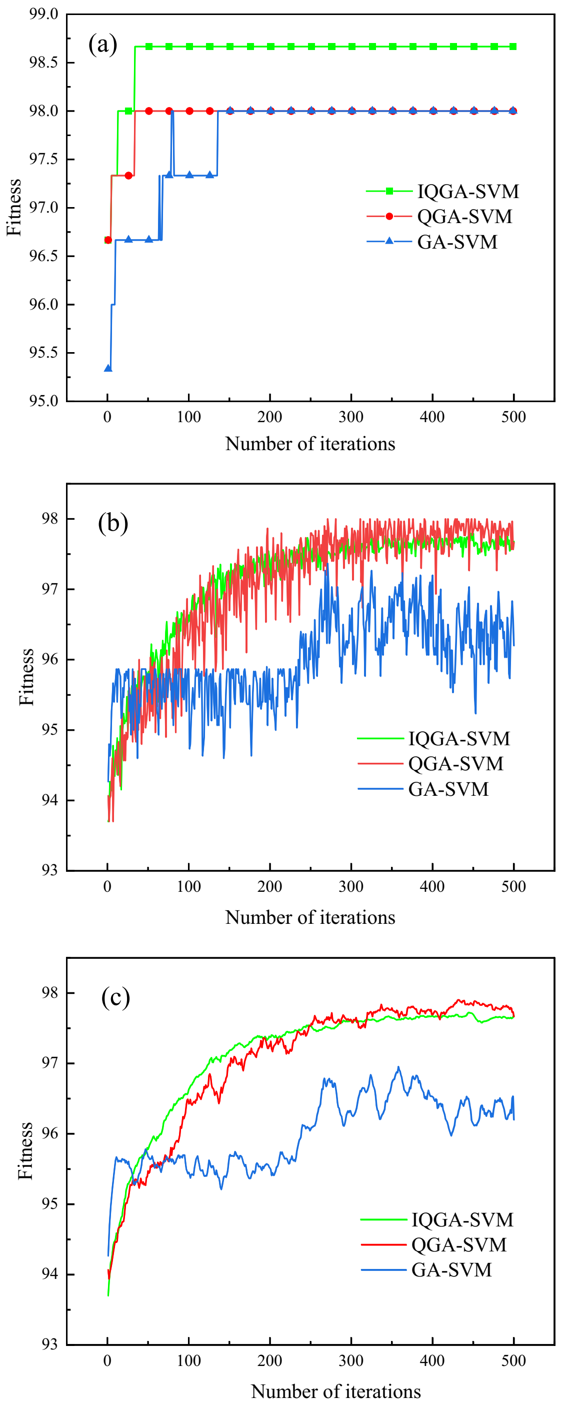
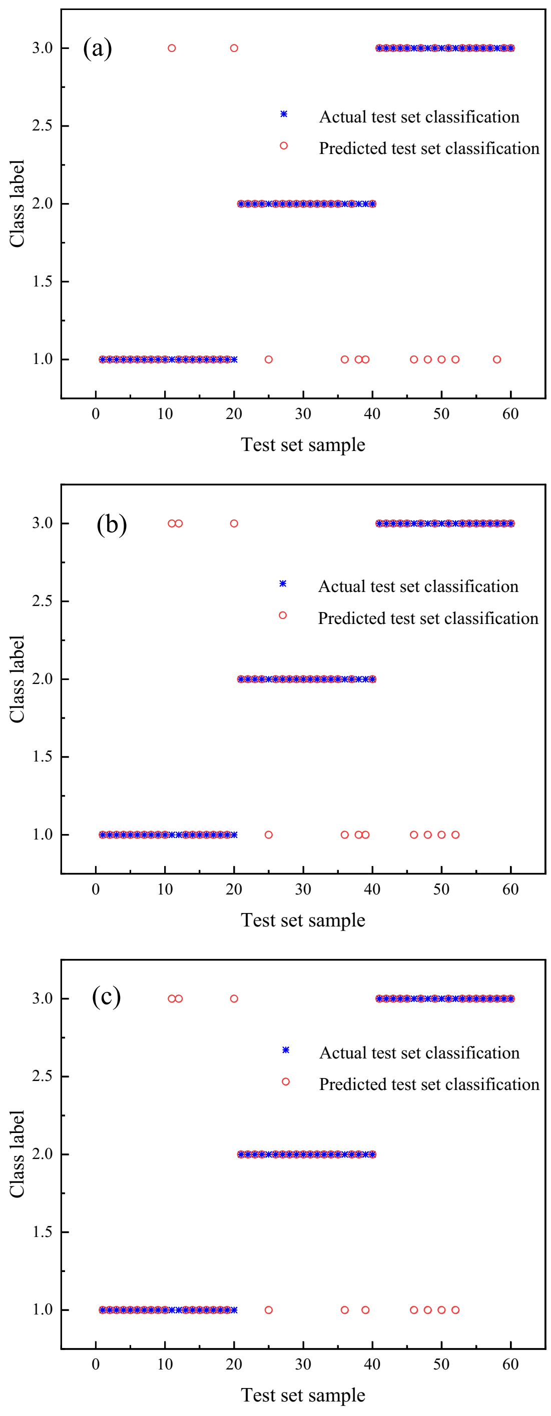


Fig. 9 Fitness curves of three algorithms in seeds dataset

Fig. 10 The prediction results of three algorithms in seeds dataset



for the best predicted model selected at this time is: Best $c = 5.5718$ and Best $g = 0.097752$. Additionally, as shown in Fig. 10c, the prediction accuracy obtained under this parameter combination is 83.3333%.

Comparing the three algorithms, we obtained that the prediction accuracy of IQGA-SVM is 1.6663% higher than that of GA-SVM. Moreover, the maximum fitness of IQGA-SVM is 0.6667% higher than that of QGA-SVM and GA-SVM. Furthermore, when IQGA-SVM reaches its maximum fitness, the evolution generation is 1 generation less than that of QGA-SVM. Thus, we can conclude that IQGA-SVM has better prediction accuracy and a faster convergence rate.

6.2.4 Comparison with back-propagation neural network

In the previous section, IQGA-SVM, GA-SVM, and QGA-SVM are used to classify the three benchmark datasets, and the experimental results are compared. The experimental results show that IQGA-SVM has better performance. However, the performance comparison between IQGA-SVM and neural network-based methods is necessary [23, 24].

We choose back-propagation neural network for comparison with IQGA-SVM, which is a multi-layer feed-forward neural network trained according to the error back-propagation algorithm and the most widely used neural network. Similarly, we use the same three benchmark datasets to test the performance of back-propagation neural networks.

In this experiment, the number of input and output layers varies according to the benchmark datasets, with the number of hidden layers equal to 6, the number of training epochs equal to 1000, the learning rate equal to 0.01, and the training objective minimum error equal to 0.0001. We select the most frequent data group for display, as shown in Fig. 11.

Figure 11a–c shows the results of back-propagation neural network prediction for iris dataset, blood dataset and seeds dataset, respectively. From Fig. 11a–c, the prediction accuracy can be obtained as 93.333%, 77.5785% and 80%, respectively.

According to the above results, the prediction accuracy of the bp neural network is inferior to that of IQGA-SVM. The accuracy of the back-propagation neural network in predicting blood dataset is better than that of GA-SVM and QGA-SVM, but it is worse than that of GA-SVM and QGA-SVM in predicting other datasets.

6.2.5 Results

As a result of these experiments, IQGA-SVM has shown some advantages in prediction results and speed, namely better prediction accuracy and faster convergence rate. The experimental results are shown in Table 4.

We also plot the error bars based on the results of five sets of experiments performed separately for these several different models to show the maximum and average values, as shown in Fig. 12. Figure 12a–c shows the five predictions made by these models on iris dataset, blood dataset and seeds dataset, respectively, and the experimental results are plotted as error bars.

Fig. 11 Prediction results of bp neural network

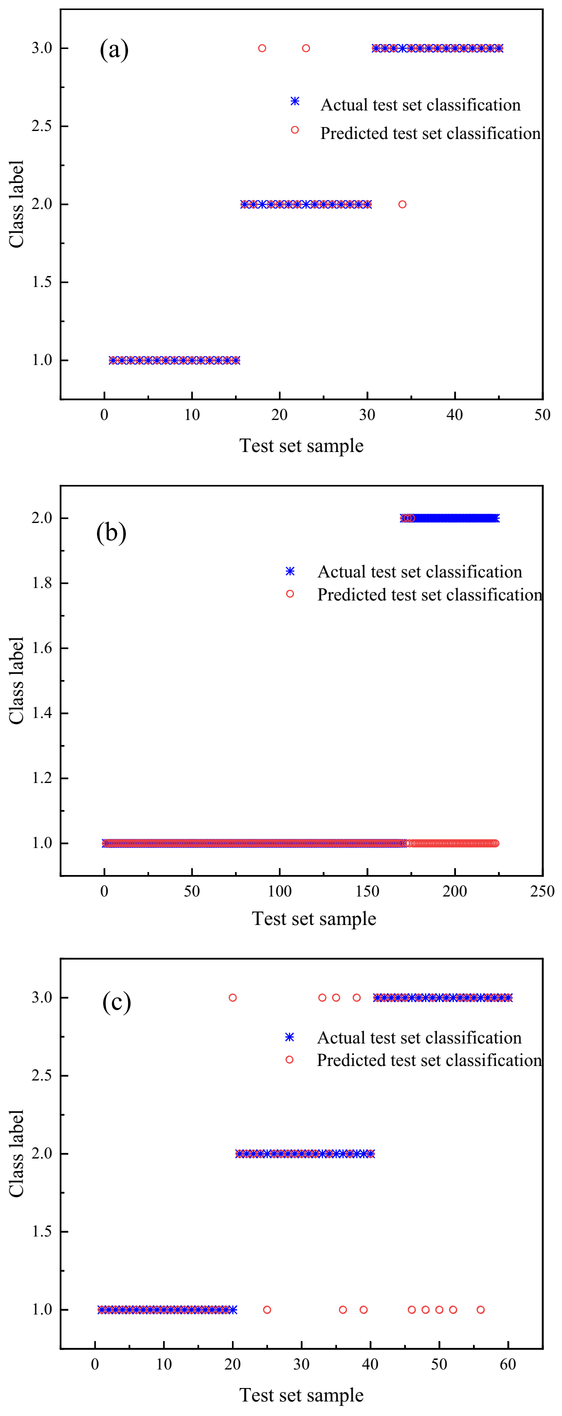


Table 4 Benchmark datasets test results

Datasets	Parameter	GA-SVM	QGA-SVM	IQGA-SVM	bp neural network
Iris	Maximum fitness	95.2381%	96.1905%	98.0952%	–
	Iterations	195	37	33	–
	Prediction accuracy	97.7778%	97.7778%	100%	93.3333%
Blood	Maximum fitness	77.5238%	77.5238%	77.9048%	–
	Iterations	5	2	341	–
	Prediction accuracy	76.2332%	76.2332%	81.6143%	77.5785%
Seeds	Maximum fitness	98%	98%	98.6667%	–
	Iterations	76	28	29	–
	Prediction accuracy	81.6667%	81.6667%	83.3333%	80%

Figure 12a shows that the standard deviation of IQGA-SVM is 0 with an average value of 100%, the standard deviation of QGA-SVM is 0.00993 with an average value of 99.556%, and the standard deviation of GA-SVM is 0.01216 with an average value of 98.668%. The standard deviation of bp neural network is 0.01991 with an average value of 94.666%.

Figure 12b shows that the standard deviation of IQGA-SVM is 0.00246 with an average value of 81.794%, the standard deviation of QGA-SVM is 0.02947 with an average value of 78.386%, and the standard deviation of GA-SVM is 0.02406 with an average value of 77.309%. The standard deviation of bp neural network is 0.0068 with an average value of 94.666%.

Figure 12c shows that the standard deviation of IQGA-SVM is 0.00745 with an average value of 83%, the standard deviation of QGA-SVM is 0.00913 with an average value of 82.333%, and the standard deviation of GA-SVM is 0.01178 with an average value of 81.667%. The standard deviation of bp neural network is 0.01816 with an average value of 79.667%.

From the above, IQGA-SVM has a smaller standard deviation and a larger average prediction accuracy.

7 Conclusion

In this paper, an IQGA is proposed and applied to parameter optimization of SVM, resulting in the establishment of the model called IQGA-SVM. The model is used for classification experiments, and finally ideal results are obtained.

The IQGA addresses the problem of local optima and being trapped in local optimal solutions that often occur in traditional quantum genetic algorithms, as well as the use of a fixed rotation angle adjustment strategy. To optimize the algorithm, a crossover and evolution strategy and a dynamic rotation angle strategy are proposed, and a quantum convergence gate is added to address the problem of premature convergence of quantum bit probability amplitudes to 0 or 1.

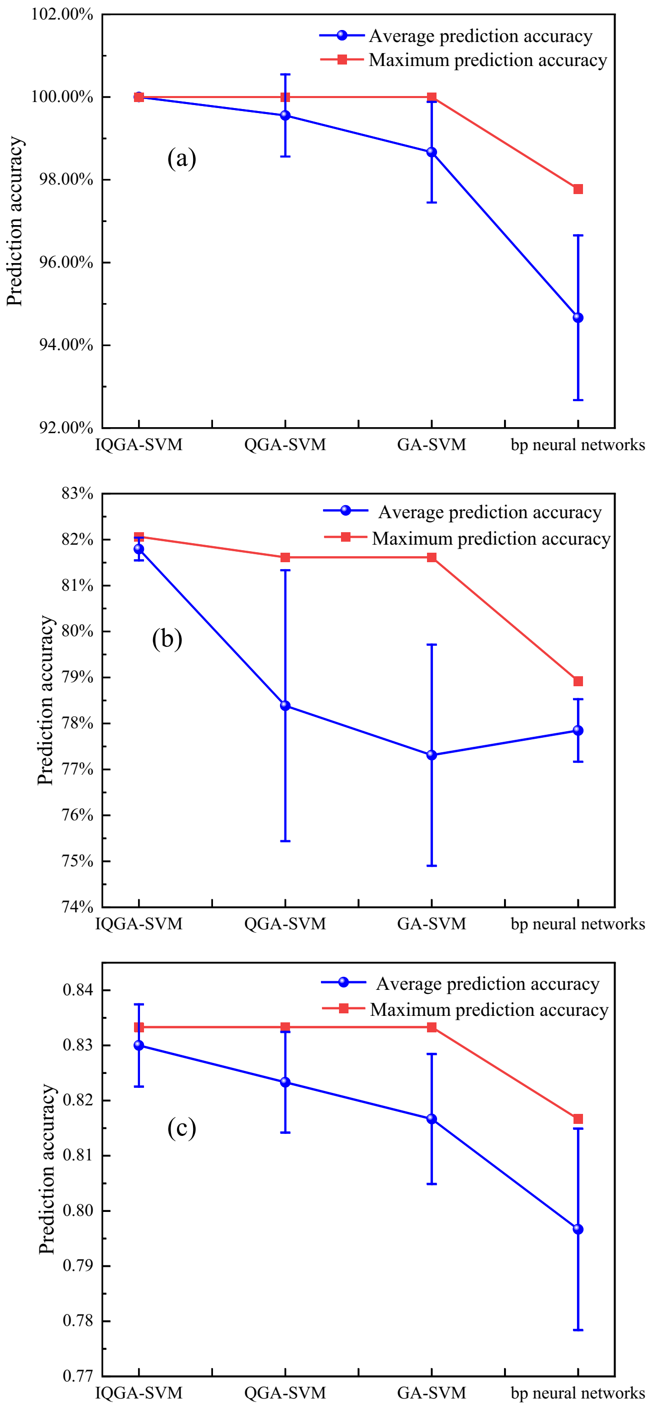


Fig. 12 Error bar

Then, the feasibility of the IQGA was verified by optimizing complex binary functions. Subsequently, three benchmark classification datasets were used to conduct experiments on IQGA-SVM and compare it with GA-SVM and QGA-SVM. Experiments demonstrate that IQGA-SVM performs better: When tested on the iris dataset, IQGA-SVM has a maximum fitness that is 0.9523% and 3.8095% higher than QGA-SVM and GA-SVM, respectively. Moreover, compared to QGA-SVM, IQGA-SVM requires 27 fewer iterations. Additionally, IQGA-SVM has a prediction accuracy that is 2.2222% higher than GA-SVM. When tested on the blood dataset, IQGA-SVM has a maximum fitness that is 0.381% higher than QGA-SVM and GA-SVM. Moreover, IQGA-SVM can break through local optimal solutions during the iteration process. Additionally, IQGA-SVM has a prediction accuracy that is 5.3811% higher than QGA-SVM and GA-SVM. When tested on the seeds dataset, IQGA-SVM has a maximum fitness that is 0.6667% and 2% higher than QGA-SVM and GA-SVM, respectively. Moreover, compared to QGA-SVM, IQGA-SVM requires 10 fewer iterations, and IQGA-SVM has a prediction accuracy that is 1.6666% higher than GA-SVM. The above results indicate that when applied to classification, IQGA-SVM application in classification shows some advantages in prediction result and speed, namely better prediction accuracy and faster convergence rate. We also introduce bp neural network for comparison, and conclude that bp neural network is worse than IQGA-SVM in the accuracy and stability of prediction.

In this paper, we conduct researches on the QGA and optimize the algorithm. The experiment proves that IQGA has better convergence rate and accuracy than that of QGA. However, the algorithm performs quantum computation on classical computers, which has certain limitations. Future work should be directed toward quantum computers and the experimental implementation of quantum machine learning algorithms on a real quantum computer.

Acknowledgements The work was supported by National Natural Science Foundation of China (Grant No. 51904272) and Major Science and Technology Projects in Henan Province, China (Grant No. 221100210600).

Availability of data and materials The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability The code generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration

Conflict of interest The authors declare that all data supporting the findings of this study are original within the article and have no competing interests.

References

1. Vapnik, V.: The Natural of Statistical Learning Theory. Springer, New York (1995)
2. Huang, C., Wang, J.: A GA-based feature selection and parameters optimization for support vector machines. Expert Syst. Appl. **31**(2), 231–240 (2006)
3. Wang, Q.: Application of Stock Price Short-Term Prediction Based on the Improved Support Vector Machine. Chongqing Jiaotong University, Chongqing (2015)

4. Meng, T., Zhou, X., Lei, Y.: A parameters optimization method for an SVM based on adaptive genetic algorithm. Comput. Meas. Control **24**(09), 215–217+223 (2016)
5. Sajan, K.S., Kumar, V., Tyagi, B.: Genetic algorithm based support vector machine for on-line voltage stability monitoring. Int. J. Electr. Power **73**, 200–208 (2015)
6. Gao, L., Zhang, X., Wang, F.: Application of improved ant colony algorithm in SVM parameter optimization selection. Comput. Eng. Appl. **51**(13), 139–144 (2015)
7. Hu, Y., Peng, M., Tian, C., Tan, H., Song, L., Shen, M.: Analog circuit fault diagnosis based on particle swarm optimization SVM. Appl. Res. Comput. **29**(11), 4053–4055 (2012)
8. Guo, F., Guo, C., Wang, Y., Liu, D.: The forecast model of mine water discharge based on particle swarm optimization and support vector machines. Comput. Eng. Sci. **34**(07), 177–181 (2012)
9. Jin, L.: Research and Application of Quantum Genetic Algorithm. Zhengzhou University, Zhengzhou (2021)
10. Lin, X., Yuan, D., Sun, Y., Wang, C., Chen, C.: The basic theory and research progress of support vector machine. J. Yangtze Univ. (Nat. Sci. Ed.) **15**(17), 48–53 (2018)
11. Li, Z.: Optimization Theory and Method. China University of Mining and Technology Press, Beijing (2012)
12. Osuna, E., Freund, R. M., Girosi, F.: Support Vector Machines: Training and Applications. A.i.memo (1997)
13. Liu, Y.: Study on Kernel Function of Support Vector Machine. Xidian University, Xi'An (2012)
14. Zhu, F., Zhao, Q., Wang, J.: The classification and application of SVM based on RBF kernel. J. Tangshan Univ. **30**(03), 13–17+39 (2017)
15. Shen, Y., Hu, G., Zhang, Z., Yan, Z.: An optimal parameter selection method in LSSVM RBF kernel function for multiple-input multiple-output regression and prediction. In: Proceedings of the 30th China Process Control Conference (CPCC). Kunming, Yunnan, China (2019)
16. Li, S., Zhang, P., Li, B., Wu, D., Hu, H.: Quantum genetic algorithm based on universal quantum gates and its applications. Comput. Eng. Appl. **53**(7), 54–59 (2017)
17. Zhou, S.: Manipulator based on improved quantum genetic algorithm trajectory optimization. Mod. Mach. Tool Autom. Manuf. Tech. **06**, 33–37 (2021)
18. Luo, C., Zhao, Q., Wei, Y., Li, S.: Distribution network fault classification based on an improved quantum genetic optimization support vector machine. J. Beijing Univ. Chem. Technol. (Nat. Sci. Ed.) **49**(06), 110–118 (2022)
19. Yang, S.: A self-adaptive quantum genetic algorithm for vehicle routing and scheduling. In: 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 1–7 (2019)
20. Michael, A.N.: A geometric approach to quantum circuit lower bounds. Quantum Inf. Comput. **6**(3), 213–262 (2005)
21. Xiong, Y., Chen, H., Miao, F., Wang, X.: A quantum genetic algorithm to solve combinatorial optimization problem. Acta Electron. Sin. **32**(011), 1855–1858 (2004)
22. Han, K., Kim, J.: Genetic quantum algorithm and its application to combinatorial optimization problem. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512). IEEE (2002)
23. Qi, C., Bi, Y., Li, Y.: Improved BP neural network algorithm model based on chaos genetic algorithm. In: Conference Proceedings of 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, pp. 698–701 (2017)
24. Ni, X., Yang, Z.: Inverter fault diagnosis based on BP neural network. In: Proceedings of 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE). Dalian, Liaoning, China, pp. 203–208 (2021)