Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 9, Number 1, pages 63–68, January 2020

Variational Quantum Support Vector Machine based on Deutsch-Jozsa Ranking

Koudai Shiba
Grid, Inc.
Tokyo, Japan
Engineering department,
The University of Electro-Communications
Tokyo, Japan

Katsuyoshi Sakamoto
Engineering deptpartment,
The University of Electro-Communications
Tokyo, Japan

Tomah Sogabe
i-PERC &
Engineering department,
The University of Electro-Communications
Tokyo, Japan
Grid, Inc.
Tokyo, Japan

***Abstract*—Recently, machine learning algorithms using quantum computers have been actively developed. Among them, the support vector machine (SVM) specialized for the classification problem is one of the algorithms that are attracting the most attention for improving the performance by the quantum computer. In this study, we propose an SVM that classifies test data according to energy level by using time evolution calculation of Ising model. Furthermore, we propose a method to determine the support vector by applying the concept of DBSCAN and Deutsch-Jozsa algorithm and examining the mixture of surrounding data classes. We show that our algorithm can be classified regardless of the difference in the dimensions of the teacher data, and also the difference between linear and nonlinear data.**

Keywords—Deutsch-Jozsa algorithm, machine learning, support vector machine, quantum algorithms

I. Introduction

In recent years, algorithms that use quantum computers have been developed one after another in anticipation of higher efficiency than computations in existing computers. Some quantum algorithms are known to be able to calculate faster than existing computers [1-3], and their applications are progressing in various fields such as optimization calculations, financial fields, quantum chemical calculations, and machine learning [4-7]. Among them, quantum machine learning has attracted attention as an application field of quantum algorithms due to the proposal of algorithms that can perform matrix calculations with quantum computers. It is the Harrow-Hassidim-Lloyd (HHL) algorithm [8], which can solve linear equations using the quantum phase estimation algorithm [9]. However, this algorithm has a deep quantum circuit and is difficult to calculate accurately with the current quantum computer hardware with a lot of noise [10]. Therefore, a hybrid algorithm that shortens the quantum circuit to be constructed and also uses a classical computer has been proposed, and a quantum variational algorithm (VQE) [11] and a quantum approximation optimization algorithm (QAOA) [12] have been developed.

Currently, various quantum machine learning algorithms have been proposed, such as unsupervised quantum Boltzmann machines [13], supervised quantum autoencoders [14], reinforcement learning using quantum Boltzmann machines [15] and quantum circuit learning [16], in which learning is performed with a quantum circuit with a short depth, has been proposed. Among them, the quantum support vector machine (QSVM) [17] is one of the most famous quantum machine learning algorithms. SVM [18] is a basic supervised learning that specializes in classification problems, and several SVM algorithms that use quantum computers have been considered. One of them is QSVM [17], which treats SVM as an approximate least squares problem by adding a hinge loss function term to the loss function and efficiently calculates the kernel matrix. The other is QSVM [19] that calculates the inner product of input vectors using a quantum circuit called the Hadamard test using phase kickback. Furthermore, QSVM [20] has been proposed in which input data is projected onto a unit circle, and then classification is performed by inputting the data to the amplitude of the qubit and observing it. Recently, QSVM [21] has been proposed in which the feature space is expanded to a high-dimensional Hilbert space and classification is performed using a quantum circuit learning algorithm.

In this study, we propose a method of applying the time evolution [4] of the Ising model calculated on the quantum gated quantum computer to SVM, which is one of the basic algorithms of machine learning. Time evolution simulation of the Ising model is one of the quantum algorithms applied to the optimization calculation, and is also called quantum annealing. We also propose a method for determining support vectors by ranking data points based on the number of data points of different classes around them. It applies the concepts of DBSCAN [22], one of machine learning clustering algorithms, and Deutsch-Jozsa algorithm [1], a quantum algorithm.

II. Quantum support vector machine based on Deutsch-Jozsa ranking

A. SVM using quantum adiabatic calculation

SVM [18] is a basic supervised learning algorithm specializing in classification problems. Fig. 1 is a schematic diagram of SVM in the linear binary classification problem. From the definition of SVM, classification is performed by maximizing the distance d between the hyperplane H_1 and H_{-1} in the schematic diagram of Fig. 1. H_1 and H_{-1} are $f(x) = wx + b = \pm 1$ lines, and the data points on the lines are called support vectors. In Fig. 1, four support vectors are displayed. For the point (x_0, y_0) in H_0, the distance of the line $Ax + By + c = 0$ can be written as follows.

$$d = 2\frac{|Ax_0 + By_0 + c|}{\sqrt{A^2 + B^2}} \quad (1)$$

Identify applicable funding agency here. If none, delete this text box.

Here, $A = w, B = 0, c = b$. The distance d can be expressed more simply by using the condition $wx_0 + b = \pm 1$ in H_1 and H_{-1}.

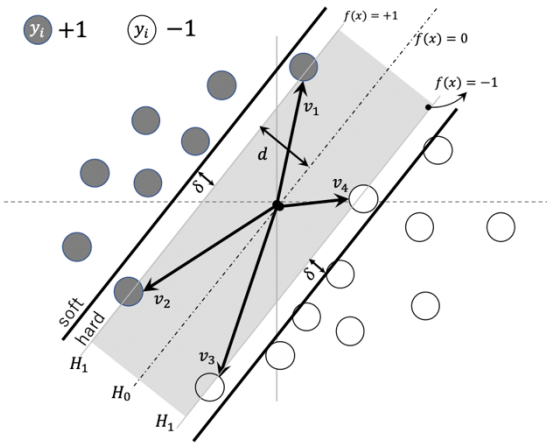

Fig. 1. Diagram of SVM in the linear binary classification problem.

$$d = 2\frac{|Ax_0 + By_0 + c|}{\sqrt{A^2 + B^2}} = 2\frac{|wx_0 + b|}{|w|} = \frac{2}{|w|} \quad (2)$$

In SVM problems, the problem of maximizing d is equivalent to the problem of minimizing $\frac{1}{2}|w|^2$. Therefore, calculating the Lagrangian of $\frac{1}{2}|w|^2$ can be expressed as follows.

$$\min L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}|\boldsymbol{w}|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{l} \alpha_i \quad (3)$$

$\alpha_i \geq 0$ is a Lagrange multiplier. Considering the constraints of $\frac{\partial L}{\partial b} = 0$ and $\frac{\partial L}{\partial \boldsymbol{w}} = 0$, the following equation is established.

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (4)$$

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i x_i \quad (5)$$

Assigning Eq. (4) and Eq. (5) to Eq. (3) gives the following equation, which can be transformed into a dual problem.

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \quad (6)$$

Eq. (6) satisfies $\alpha_i \geq 0$ and Eq. (4). Here, $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ represents the inner product of two vectors and can be considered as interaction energy. By developing the kernel matrix $K_{ij} = K(x, x_j)$, it is possible to deal with nonlinear problems.

On the other hand, in this study, we consider all data as physical particles, and express the relationship between the data using the Ising model Hamiltonian based on the correlation matrix and distance matrix. A term representing a correlation between data in the Hamiltonian and a bias term applied to each data can be expressed as follows.

$$H = \sum_{i<j} J_{ij} s_i s_j + \sum_i h_{z_i} s_i \quad (7)$$

The quantum computer can simulate the time evolution of the Hamiltonian and can determine the ground state energy. By reversing the sign of Eq. (7), we considered that it is

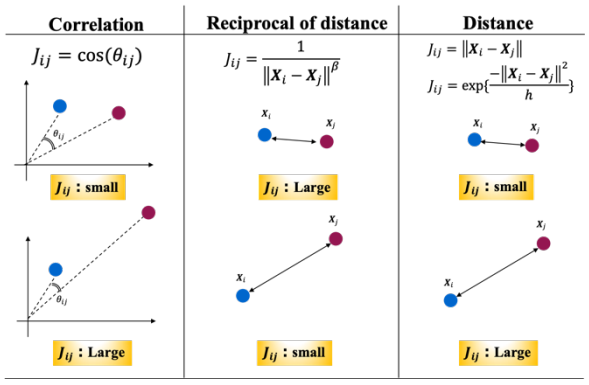

Fig. 2. Type of data to be input as coupling coefficient J_{ij}.

equivalent to Eq. (6), and we proposed an SVM that solves the classification problem by calculating the Hamiltonian basis energy of the Ising model on a quantum computer.

The coupling coefficient J_{ij} in Eq. (7) corresponds to the kernel matrix $K_{ij} = K(x, x_j)$ in Eq. (6), and correlation ($J_{ij} = \cos(\theta_{ij})$), distance ($J_{ij} = \|\boldsymbol{X}_i - \boldsymbol{X}_j\|$), Gaussian kernel ($J_{ij} = \exp(-\sigma\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2)$), reciprocal of distance ($J_{ij} = 1/\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^\beta$), etc. can be applied. These differences are shown in Fig. 2. The longitudinal magnetic field coefficient h_{z_i} of the Ising model corresponds to the class label data in the data $\boldsymbol{X}_i$.

The proposed quantum SVM has no learning mechanism and uses teacher data every time it performs calculations to predict the class of test data. Specifically, the following flow is performed:

1. Consider an Ising model based on multiple teacher data (support vectors) and one test data.
2. As shown in Fig. 3, Execute time evolution by quantum adiabatic calculation [23], and when the class label of the test data is 1 or -1, it is determined which is in the stable state of the Hamiltonian.
3. The value that results in a stable state is taken as the predicted value of the test data.

The Ising model's longitudinal magnetic field coefficient h_{z_i} takes three values, $h_{z_i} \in \{-1, 0, 1\}$. In the case of teacher data, the corresponding class label $h_{z_i} \in \{-1, 1\}$ and $h_{z_i} = 0$ is substituted for test data.

B. Quantum adiabatic calculation

Quantum adiabatic calculation [23] is one of the annealing calculation methods calculated using the Ising model. The Ising model is a model that models the behavior of spin in a magnetic material such as a ferromagnet or antiferromagnet, and has two types of states: upspin ($s_i = +1$) and downspin ($s_i = -1$).

The Hamiltonian of the entire system of the Ising model is expressed by the following equation using the coupling coefficient J_{ij} between the two spins s_i and s_j and the local

longitudinal magnetic field h_{z_i} applied to the inside of the spin s_i.

$$H = \sum_{i<j} J_{ij} s_i s_j + \sum_i h_{z_i} s_i \quad (8)$$

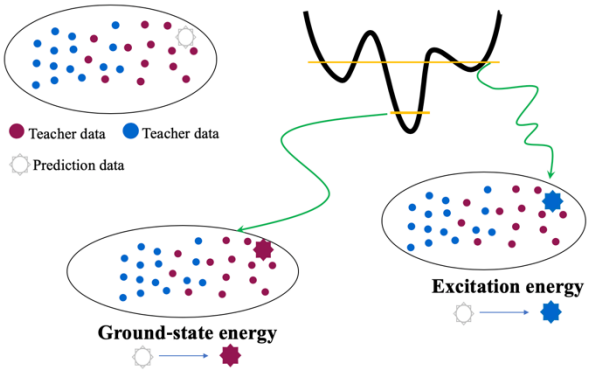

Fig. 4. Diagram of QSVM classifying by Hamiltonian basis energy calculation.

In the quantum adiabatic calculation, a transverse magnetic field h_x is added to set the initial state of the Hamiltonian. Also, since the spin s_i corresponds to the Pauli operator σ_i^z, it can be expressed by the phase inversion operation gate Z of the quantum gate represented by the same matrix. Further, a parameter $s(= t/t_f)$ in which time t is normalized by t_f is introduced, and $0 \leq s \leq 1$. Therefore, the Hamiltonian in quantum adiabatic computation is as follows.

$$H(s) = s\left[\sum_{i<j} J_{ij} Z_i Z_j + \sum_i h_{z_i} Z_i\right] + (1-s)\sum_i h_x X_i \quad (9)$$

In a quantum computer, the time evolution of the Schrödinger equation can be expressed by performing unitary transformations in order. Assuming that the state vector of the qubit is $|\psi\rangle$, the Schrodinger equation can be expressed as follows.

$$i\hbar \frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle \quad (10)$$

Solving the Schrödinger equation when the state vector depends on time and the Hamiltonian does not depend on time can be transformed as follows, and the unitary transformation $U(t)$ is derived.

$$|\psi(t)\rangle = e^{-i\frac{H}{\hbar}t}|\psi(0)\rangle \quad (11)$$

$$U(t) = e^{-i\frac{H}{\hbar}t} \quad (12)$$

Therefore, by substituting Eq. (2) for H in Eq. (5) and repeatedly performing unitary transformation $U(t)$, the minimum value of the Hamiltonian can be obtained and an optimum spin state can be obtained. $U(t)$ in Eq. (12) is called a time evolution operator, and the detailed quantum circuit is shown in Fig. 4. In H of Eq. (9), the term with the coupling coefficient corresponds to the combination of two CNOT gates and R_z gate of the quantum circuit, the longitudinal magnetic field term corresponds to the R_z gate, and the transverse magnetic field term corresponds to the R_x gate. Each coefficient, time evolution coefficient s, etc. are input as the input angle of the rotary gate. Thus, for example, when s evolves over 100 steps, the parts other than the two H gates in Fig. 4 are repeated 100 times.

Fig. 5. Part of the quantum circuit for quantum adiabatic computation in the case of two qubits.

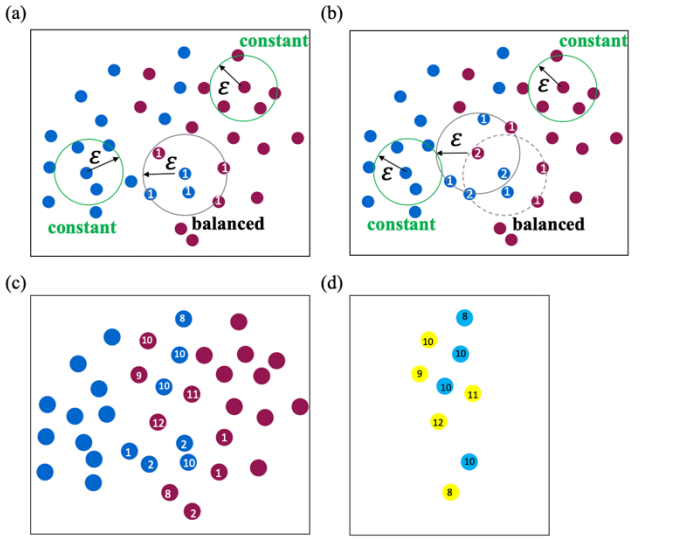

Fig. 3. (a)(b) Diagram of examining class labels of other teacher data points within a circle of radius ε centered on each teacher data point. (c) Schematic of the state where all teacher data points have been examined as the center of the circle. (d) Diagram of the top 1/3 ranking data determined as support vectors.

C. *Determine support vectors*

The support vector determination method is inspired by the idea of DBSCAN [22] and the Deutsch-Jozsa algorithm [1]. DBSCAN (Density-based spatial clustering of applications with noise) is one of clustering algorithms in machine learning. Each data point is classified into three types according to the number of other data points present in a circle with a radius ε centered on each data point, and a cluster is generated based on this classification. The Deutsch-Jozsa algorithm is a quantum algorithm that is determined by a single measurement whether the output $f(x) \in \{0, 1\}$ of the binary function $f(x)$ for a binary input $x \in \{0, 1\}^n$ of n qubits depends on the input x (balanced) or not (constant). The quantum circuit of the Deutsch-Jozsa algorithm is shown in Fig. 6 (a).

Applying these algorithm concepts, support vectors are determined from a lot of teacher data. Specifically, the following flow is performed. This flow is shown in Fig. 5:

1. Examine the class label of the other teacher data point existing in the circle of radius ε centered on each teacher data point.
2. Determine whether the teacher data in the circle including the central data point are all the same class (constant) or mixed with other classes (not constant).
3. When the classes are mixed, we recognize all data points existing in the circle as data points close to the boundaries of a plurality of clusters. For convenience, we call this data point "RP" (representative point).
4. Repeat steps 1 to 3 for all data points and rank the number certified as RP.

5. Decide the top $1/\alpha$ data in the ranking as the support vector for all teacher data.

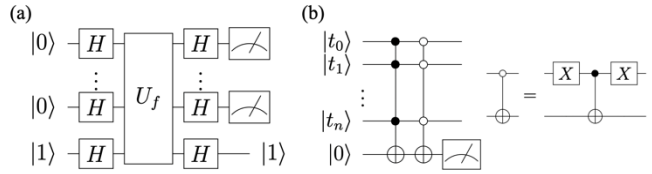

Fig. 7. (a) The quantum circuit for Deutsch-Jozsa algorithm. (b) The quantum circuit that determines whether all other teacher data within the circle of radius ε centered each teacher data point are of the same class.

The distance between each teacher data point can be considered as a coupling coefficient J_{ij} in the Ising model. This time, the coupling coefficient J_{ij} is the reciprocal of the distance between each teacher data point. Presence / absence of other teacher data points in a circle of radius ε is determined by the following relational expression.

$$J_{ij} = \frac{1}{\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^{\beta}} < \varepsilon \tag{13}$$

ε is a hyperparameter, and we can adjust the number of teacher data points contained in the circle.

Next, Fig. 6(b) shows a quantum circuit that determines the class of each teacher data contained in the circle. We prepare a quantum bit corresponding to the number of each teacher data point included in the circle and one ancilla bit, and input a label value $f(x) \in \{0, 1\}$ of each teacher data class. From the quantum circuit in Fig. 6(b), the ancilla bit is inverted only when the class of teacher data in the circle is all 0s or all 1s. Therefore, when the observed ancilla bit is 0, it is understood that the teacher data in the circle is a mixture of a plurality of classes, and the teacher data point in the circle is recognized as RP. The quantum circuit in Fig. 6(b) is a quantum circuit that uses the Multi-control NOT gate twice. Depending on the number of teacher data, the number of qubits used increases, making it difficult to implement the Multi-control NOT gate. It is necessary to search the optimum value of the number of data points included in the circle by adjusting the radius ε as a hyperparameter.

Finally, we aggregate the number certified as RP for all teacher data, and determine the top $1/\alpha$ data as support vectors. α is a hyperparameter, and the number of support vectors varies greatly depending on the distribution of teacher data. Therefore, α needs to be adjusted according to the distribution of teacher data.

III. Verification of the Proposed Method

A. Comparative verification of classification problems

We demonstrate the classification of test data for the two types of teacher data (linear data and nonlinear data) using the proposed method. In addition, we compare with the proposed method using scikit-learn SVM. The four types of verification results are shown in the Fig. 7. We used 20 teacher data (square points) for each verification. In the case of linear data, there are 11 teacher data for +1 label (red points), 9 teacher data for -1 label (blue points). In the case of non-linear data, there are 14 teacher data for +1 label (red points), and 6 teacher data of -1 label (blue points). Large square points in the teacher data are support vectors. The test data is indicated by round dots, and the number of test data is 17 for both linear data and nonlinear data. In the classification using DJ-QSVM, we set the radius of the scan circle to $\varepsilon = 0.5$ (linear data) and

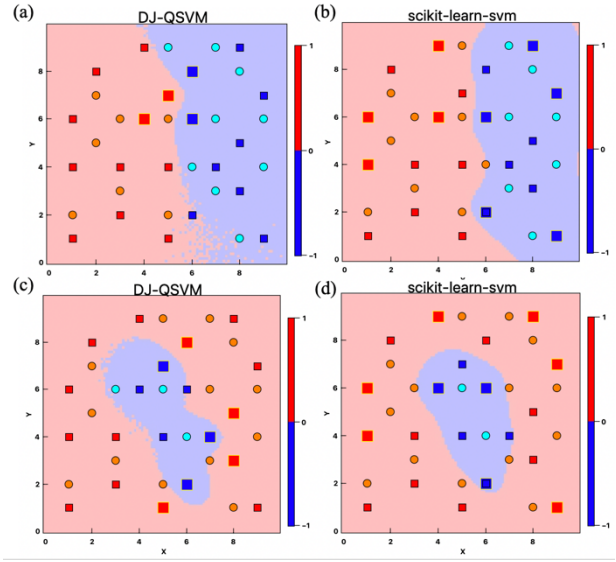

Fig. 6. Result of classification of test data. Square points are teacher data, large square points are support vectors, and round points are test data. (a) Linear data classification by DJ-QSVM. (b) Linear data classification by scikit-learn. (c) Nonlinear data classification by DJ-QSVM. (d) Nonlinear data classification by scikit-learn.

$\varepsilon = 0.6$ (nonlinear data), the power of the reciprocal of the distance between data to $\beta = 1$, and the criterion for determining the support vector from the ranking to $\alpha = 3$. In addition, we used scikit-learn.svm.SVC and classified all parameters with default values. From Fig. 7, we found that both linear classification and nonlinear classification by the proposed method can be classified like scikit-learn. However, if we change only the position without changing the number of teacher data, there are cases where it cannot be classified well. A possible cause is a small number of teacher data. As a result, if the number of teacher data is too small, the number of support vectors is insufficient and the classification accuracy may be reduced. Therefore, we think that we should classify using a lot of teacher data. However, with DJ-QSVM, quantum adiabatic calculations are performed using a number of qubits that combine all support vectors and one test data. For this reason, if the number of teacher data is too large, calculation cannot be performed correctly, or in the case of a simulator, calculation takes a long time. Therefore, in the future, it will be necessary to develop a method for calculating large-scale teacher data without increasing the number of qubits.

B. Verification of parameter differences

Next, we verified the effect on class label prediction by changing two parameters ε and β in Eq. (7) in the classification of linear data using DJ-QSVM. The verification results are shown in Fig. 8. First, Fig. 8(d) shows the time evolution in quantum adiabatic computation, with ε varied into three types, 0.3, 0.5, and 0.7. Finally, the one with the higher probability of 1 label or -1 label is the predicted label. Other hyperparameters were set to $\beta = 1$ and $\alpha = 3$. The support vectors when ε is changed are as shown in Fig. 8(a), (b), and (c). The white circles in the Fig. 8 are the test data coordinates $(x, y) = (6, 4)$ predicting the class. From Fig. 8(d), it was found that the prediction results differed depending on ε. In addition, when the +1 label teacher data is

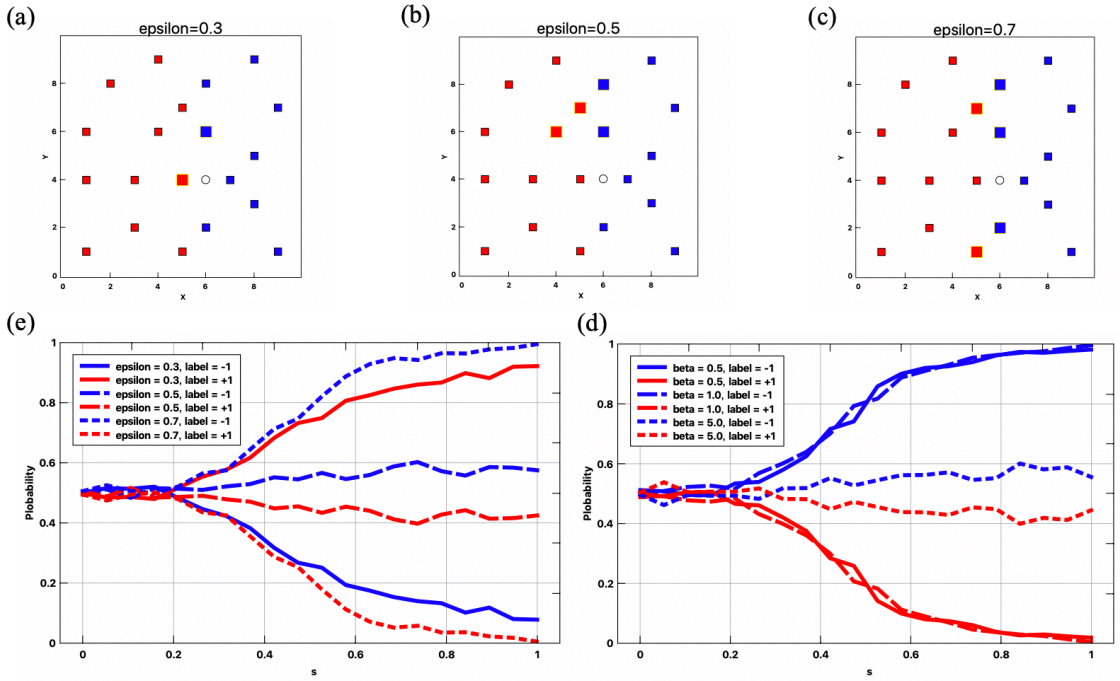

Fig. 8. (a) (b) (c) Differences in support vectors for the three types of ε. The white circles in the figure are the coordinates of the test data used for verification. (d) Process of time evolution in three types of ε. (e) Process of time evolution in three types of β.

$\varepsilon = 0.3$ with respect to the test data, the prediction result is +1, and when the -1 label teacher data is close $\varepsilon = 0.7$, the prediction result is -1. It was also found that the probabilities of +1 and -1 in the prediction results are almost equal when $\varepsilon = 0.5$, which has test data at the same distance for the two classes of teacher data.

Next, Fig. 8(e) shows the change in the prediction result when β is changed to three types of 0.5, 1.0, and 5.0. $\alpha = 3$, and ε and test data were the same as in Fig. 8(c). From Fig. 8(e), it was found that the larger β is, the smaller the difference from other data is, and the probability of +1 and -1 in the prediction result was almost equal. From the above results, it was found that the prediction results also change when ε and β are changed. These hyper parameters may change depending on the coordinates of test data, the number and types of teacher data, and the like. Therefore, in order to classify with high accuracy, it seems that you should avoid using the same hyperparameters in various classifications.

IV. Conclusion

In this study, we proposed an SVM that performs quantum adiabatic computation, which is a time evolution simulation algorithm of the Ising model, on a quantum computer and classifies test data using teacher data. In addition, we applied the concept of quantum algorithm Deutsch-Jozsa algorithm and proposed a method to determine support vectors by ranking data points by the number of data points of surrounding different classes. This proposed method is a method that can determine a support vector without performing complicated calculations, and can also handle high-dimensional teacher data. However, we did not get good results with complex teacher data, so we need to work on improving this algorithm in the future.

References

[1] David Deutsch and Richard Jozsa, "Rapid solution of problems by quantum computation," Proceedings of the Royal Society of London A, 439, 553, 1992.

[2] Lov K. Grover, "A fast quantum mechanical algorithm for database search," Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC), pp.212-219, 1996.

[3] Peter W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM journal on computing 26.5, 1484-1509, 1997.

[4] Tadashi Kadowaki and Hidetoshi Nishimori, "Quantum annealing in the transverse Ising model," Phys. Rev. E 58, 5355, 1998.

[5] Nikitas Stamatopoulos, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen and Stefan Woerner, "Option Pricing using Quantum Computers," arXiv: 1905.02666, 2019.

[6] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, A. Megrant, J. Y. Mutus, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik and J. M. Martinis, "Scalable Quantum Simulation of Molecular Energies," Phys. Rev. X 6, 031007, 2016.

[7] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. Schuyler Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, Colm A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, Robert S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, Blake R. Johnson, M. Reagor, M. P. da Silva and C. Rigetti, "Unsupervised Machine Learning on a Hybrid Quantum Computer," arXiv: 1712.05771, 2017.

[8] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd, "Quantum Algorithm for Linear Systems of Equations," Phys. Rev. Lett. 103, 150502, 2009.

[9] Gilles Brassar, Peter Hoyer, Michele Mosca and Alain Tapp, "Quantum Amplitude Amplification and Estimation," quant-ph/0005055, 2000.

[10] J. Preskill, "Quantum computing in the NISQ era and beyond," arXiv: 1801.00862, 2018.

[11] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow and Jay M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature 549, 242–246, 2017.

[12] Edward Farhi, Jeffrey Goldstone and Sam Gutmann, "A Quantum Approximate Optimization Algorithm," arXiv: 1411.4028, 2014.

[13] Vincent Dumoulin, Ian J. Goodfellow, Aaron Courville and Yoshua Bengio, “On the Challenges of Physical Implementations of RBMs,” arXiv: 1312.5258, 2013.

[14] Jonathan Romero, Jonathan P. Olson and Alan Aspuru-Guzik, “Quantum autoencoders for efficient compression of quantum data,” arXiv: 1612.02806, 2016.

[15] Daniel Crawford, Anna Levit, Navid Ghadermarzy, Jaspreet S. Oberoi and Pooya Ronagh, “Reinforcement Learning Using Quantum Boltzmann Machines,” arXiv: 1612.05695, 2016.

[16] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa and Keisuke Fujii, “Quantum circuit learning,” Phys. Rev. A 98, 032309, 2018.

[17] Patrick Rebentrost, Masoud Mohseni and Seth Lloyd, “Quantum support vector machine for big data classification,” arXiv: 1307.0471, 2013.

[18] V. Vapnik and A. Lerner, “Pattern recognition using generalized portrait method,” Automation and Remote Control, 24, 1963.

[19] M. Schuld, I. Sinayskiy and F. Petruccione, “An introduction to quantum machine learning,” arXiv: 1409.3097, 2014.

[20] Maria Schuld, Mark Fingerhuth and Francesco Petruccione, “Implementing a distance-based classier with a quantum interference circuit,” arXiv: 1703.10793, 2017.

[21] Vojtech Havlicek, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow and Jay M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” Nature 567, 209–212, 2019.

[22] Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” proceeding of 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226-231, 1996.

[23] Edward Farhi, Jeffrey Goldstone, Sam Gutmann and Michael Sipser, “Quantum Computation by Adiabatic Evolution,” quant-ph/0001106, 2000.