

Quantum annealing for semi-supervised learning

Yu-Lin Zheng(郑玉鳞), Wen Zhang(张文), Cheng Zhou(周诚), and Wei Geng(耿巍)
Citation: Chin. Phys. B, 2021, 30 (**4**): 040306. DOI: 10.1088/1674-1056/abe298
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Efficient self-testing system for quantum computations based on permutations

Shuquan Ma(马树泉), Changhua Zhu(朱畅华), Min Nie(聂敏), and Dongxiao Quan(权东晓)
Chin. Phys. B, 2021, 30 (**4**): 040305. DOI: 10.1088/1674-1056/abe29a

Pulse-gated mode of commercial superconducting nanowire single photon detectors

Fan Liu(刘帆), Mu-Sheng Jiang(江木生), Yi-Fei Lu(陆宜飞), Yang Wang(汪洋), and Wan-Su Bao(鲍皖苏)
Chin. Phys. B, 2021, 30 (**4**): 040302. DOI: 10.1088/1674-1056/abc2c2

On the time-independent Hamiltonian in real-time and imaginary-time quantum annealing

Jie Sun(孙杰)† and Songfeng Lu(路松峰)‡
Chin. Phys. B, 2020, 29 (**10**): 100303. DOI: 10.1088/1674-1056/aba2db

Experimental implementation of a continuous-time quantum random walk on a solid-state quantum information processor

Maimaitiyiming Tusun(麦麦提依明·吐孙), Yang Wu(伍旸), Wenquan Liu(刘文权), Xing Rong(荣星), Jiangfeng Du(杜江峰)
Chin. Phys. B, 2019, 28 (**11**): 110302. DOI: 10.1088/1674-1056/ab44ae

Nuclear magnetic resonance for quantum computing: Techniques and recent achievements

Tao Xin(辛涛), Bi-Xue Wang(王碧雪), Ke-Ren Li(李可仁), Xiang-Yu Kong(孔祥宇), Shi-Jie Wei(魏世杰), Tao Wang(王涛), Dong Ruan(阮东), Gui-Lu Long(龙桂鲁)
Chin. Phys. B, 2018, 27 (**2**): 020308. DOI: 10.1088/1674-1056/27/2/020308

SPECIAL TOPIC — Quantum computation and quantum simulation

Quantum annealing for semi-supervised learning

Yu-Lin Zheng(郑玉鳞), Wen Zhang(张文), Cheng Zhou(周诚), and Wei Geng(耿巍)†

Hisilicon Research, Huawei Technologies Co., Ltd., Shenzhen, China

(Received 14 October 2020; revised manuscript received 23 December 2020; accepted manuscript online 3 February 2021)

Recent advances in quantum technology have led to the development and the manufacturing of programmable quantum annealers that promise to solve certain combinatorial optimization problems faster than their classical counterparts. Semi-supervised learning is a machine learning technique that makes use of both labeled and unlabeled data for training, which enables a good classifier with only a small amount of labeled data. In this paper, we propose and theoretically analyze a graph-based semi-supervised learning method with the aid of the quantum annealing technique, which efficiently utilizes the quantum resources while maintaining good accuracy. We illustrate two classification examples, suggesting the feasibility of this method even with a small portion (30%) of labeled data involved.

Keywords: quantum annealing, semi-supervised learning, machine learning

PACS: 03.67.−a, 03.67.Lx **DOI:** 10.1088/1674-1056/abe298

1. Introduction

The recent developments of machine learning enable computers to infer patterns that were previously untenable from a large data set.[1,2] Quantum computing, on the other hand, has been proved to outperform classical computers in some specific algorithms.[3–7] To extend both advantages, increasing efforts have been made to explore the merging of these two disciplines.[8–10] For instance, the quantum version of linear models of machine learning, such as support vector machines (SVM),[11] principal component analysis (PCA),[12] can be potentially more efficient than their classical versions. Quantum generative models were also proposed with exponential speedups compared to the traditional models.[13] However, most of those algorithms require a large-scale fault-tolerant quantum computer that is beyond the ability of current hardware techniques.

Meanwhile, quantum annealer, as one of the noisy-intermediate scale quantum (NISQ) devices,[14] has been proved useful in many applications such as optimization,[15] simulation,[16] and machine learning.[17] In this work, we propose a method to tackle semi-supervised classification tasks on a quantum annealer. An encoding scheme and a similarity-calculation method that map the graph representation of the problem to the Hamiltonian of a quantum annealing (QA) system are suggested, which avoid the implementation of multi-qubit interaction. We show in two examples that good classification accuracy can be achieved using only a small amount of labeled data.

1.1. Semi-supervised learning

Semi-supervised learning (SSL) method usually refers to classification tasks where trainings are usually carried out with both labeled and unlabeled data when only a small amount of labeled data is available.[18–20] A well-known model of this kind is the graph-based method, by which all the data are represented by vertices of a graph, and similarities between data are represented by the edges of the graph.[21] Specifically, we are given a labeled data set $\mathcal{D}_l = \{(\boldsymbol{d}_1, lab_1), (\boldsymbol{d}_2, lab_2), \ldots, (\boldsymbol{d}_l, lab_l)\}$ and an unlabeled data set $\mathcal{D}_u = \{\boldsymbol{d}_{l+1}, \boldsymbol{d}_{l+2}, \ldots, \boldsymbol{d}_{l+u}\}$, where lab_i is the label of data $\boldsymbol{d}_i$, l and u are the numbers of labeled and unlabeled data, respectively. In most cases, the data set has fewer degrees of freedom than its original dimension, which allows us to map the data $\boldsymbol{d}_i$ from its original space to $\boldsymbol{x}_i$ in a certain manifold after some dimension-reduction processes, such as locally linear embedding[22] and principal component analysis.[23] Therefore, a graph $G = (V, S)$ can be built based on $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, in which the set of vertices $V = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+u}\}$ represents all the data vectors in the manifold. The adjacency matrix $S = [S_{ij}]$ of this graph represents the similarities between the i-th data and the j-th data for $1 \leq i, j \leq l+u$. Here we assume a symmetric similarity, i.e., $S_{ij} = S_{ji}$ for $i \neq j$ and $S_{ii} = 0$. During the training process, label information tends to propagate along the edges with greater similarities. When the training process ends, labels of the unlabelled vertices are decided according to their probability distributions. In practice, we need to deal with integer programming and combinatorial optimization when solving most SSL problems, which are usually non-convex or non-smooth.[20] In order to obtain a global optimal solution, the solving process usually involves high time and space complexities. In particular, classical graph-based semi-supervised learning needs to construct a graph containing all pieces of data, the time and space complexity of the algorithm is usually $\mathcal{O}(n^3)$,[24,25] where n is the

†Corresponding author. E-mail: wei.geng@huawei.com

© 2021 Chinese Physical Society and IOP Publishing Ltd

number of all data, such as local and global consistency (LGC) semi-supervised learning.[26] This poses a big problem when learning on large data sets. With the potential speed-up in solving the optimization problems, the QA method turns out to be a natural consideration to be applied in SSL.

1.2. Quantum annealing

In a QA process,[27] the system is firstly prepared in a ground state of an initial Hamiltonian. A target Hamiltonian is gradually applied to the system as it evolves following the time-dependent Schrödinger equation. If the application of the target Hamiltonian is slow enough, the system will adiabatically stay at the ground state of the instantaneous Hamiltonian and finally reach to the ground state of the target Hamiltonian, which encodes the solution of the problem. Demonstration of QA has been vastly reported using systems based on superconducting circuits.[28–31] When an Ising model is used in a QA system, the Hamiltonian of the annealing process is usually defined as below:

$$H = s(t)H_{\text{ini}} + (1 - s(t))\,H_{\text{tar}}, \tag{1}$$

$$H_{\text{tar}} = -\sum_{i=1}^{N} h_i\sigma_i - \sum_{i,j=1,i\neq j}^{N} J_{ij}\sigma_i\sigma_j, \tag{2}$$

in which H_{ini} and H_{tar} are the initial and target Hamiltonian of the system, respectively, h_i is the bias applied on the i-th qubit, σ_i is a Pauli Z operator on the i-th qubit, and J_{ij} is the coupling parameter between the i-th qubit and the j-th qubit. $s(t)$, as a function of time t, controls the annealing speed by monotonically decreasing from 1 to 0. For QA algorithms, it has been shown that the time complexity is only related to the energy difference between the energy levels, i.e., $\mathcal{O}(1/(\Delta e)^2)$, where Δe is the energy difference between ground state and the first excited state of the system.[32]

2. Method

In this section, we introduce the whole procedures of our algorithm, as illustrated in Fig. 1.

2.1. Label encoding

The graph G can be mapped onto a QA system by associating each vertex i with a group of α qubits $\{q_i^{(1)}, q_i^{(2)}, \ldots, q_i^{(\alpha)}\}$ that encodes the label, and by associating the edge S_{ij} between vertices i and j with the coupling strengths J_{ij} between qubit groups. The binary nature of qubits leads to an intuitive choice of a binary encoding for labels, which allows the group to encode up to $K = 2^\alpha$ labels. Such an encoding scheme usually calls for global constraints on the group of qubits in order to guarantee that a proper code is simultaneously attributed to the whole group at the end of the annealing process. However, this can only be achieved via multi-qubit interactions, a mechanism that current QA hardware can hardly address. One-hot encoding can overcome this requirement by reducing the interaction to a quadratic order at a cost of exponentially increased qubit resources with regard to the label number compared with the binary encoding.[33,34]

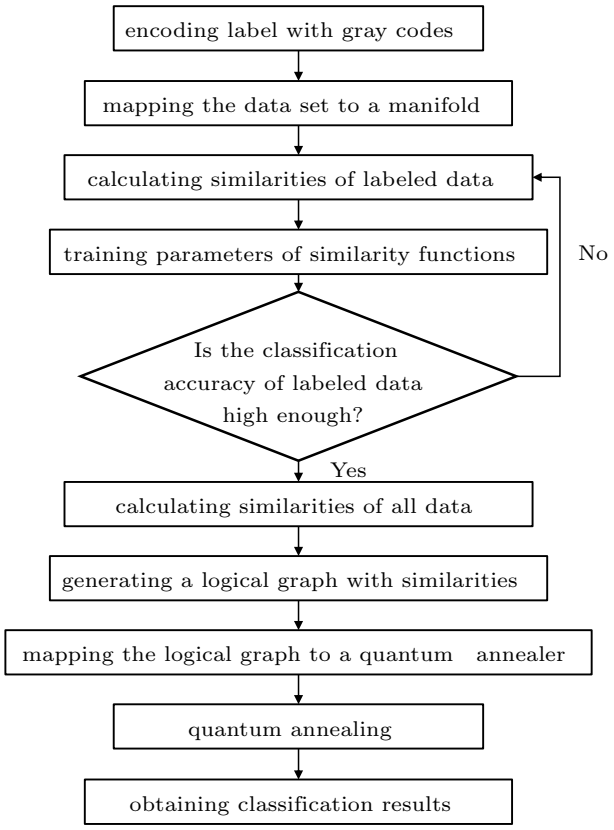


Fig. 1. A flowchart of our method.

Here we propose another encoding method that essentially uses binary codes while avoiding the hardware complexity introduced by multi-qubit interactions. We suggest that the a-th bit in the binary label code of all unlabeled data is determined independently by a QA process, which consists of the qubits $\{q_1^{(a)}, q_2^{(a)}, \ldots, q_{l+u}^{(a)}\}$. In this case, the global information shared within one group of qubit should be maintained by the consistency of similarity between arbitrary two data and the Hamming distance of their labels. Specifically, we can first calculate the barycenters of each label on the manifold by $\boldsymbol{X}^{(k)} = \frac{1}{n_k}\sum_i \boldsymbol{x}_i^{(k)}$ according to the labeled data set $\mathcal{D}_l$, where n_k denotes the number of data with the k-th label. A shortest path that visits all the barycenters is then searched in the manifold, leading to a sequence of all the barycenters: $\{\boldsymbol{X}^{(k)} | k \in \mathcal{K}\}$, where $\mathcal{K}$ is the set of all labels. According to this sequence, labels are assigned with an ordered gray code, which ensures that only one bit changes in the codes of two adjacent labels. Thus, the correlation of distances between arbitrary data and their label codes' Hamming distance is optimized. Searching for the shortest path of barycenters is an example to optimize the label encoding, we have to admit that

it is still a difficult problem for any algorithms to distinguish labels in cases where several label centers are mixed together.

Generally, we can calculate the centers of each label with the aid of distribution assumptions for different labeled data sets. For example, if the data set of a particular label is big enough and follows a particular normal distribution, we can calculate its center with a better accuracy than the barycenter.

Though the complexity of finding the shortest path in the manifold is equivalent to the well-known travelling salesman problem, in most cases, the number of label is far fewer than that of data in a given data set. If the number of label is too large to endure while solving by a classical computer, we can also apply a quantum annealer to the problem. It has also been shown that this kind of task could also be potentially accelerated by a QA device.[35]

There are certainly cases that $2^{\alpha-1} < K < 2^{\alpha}$. To avoid that the redundant codes are wrongly attributed to an unlabeled data, we can use up to two codes to encode one label while assuring the two codes are next to each other in the ordered gray code. For example, if a group of $\alpha = 3$ qubits is used to encode only $K = 5$ labels, the label codes can be attributed as follows: $\{000,001\}_{\text{label 1}}$, $\{011,010\}_{\text{label 2}}$, $\{110,111\}_{\text{label 3}}$, $\{101\}_{\text{label 4}}$, $\{100\}_{\text{label 5}}$.

2.2. Structure of the system

Based on the aforementioned encoding method, the whole training of the SSL classification task can be divided into α independent layers, of which the a-th corresponds to an annealing process of qubits $\{q_1^{(a)}, q_2^{(a)}, \ldots, q_{l+u}^{(a)}\}$. Because of the limited connectivity that a QA hardware can currently achieve, we only require that each qubit is logically coupled with at least ξ qubits in each layer. ξ may depend on the certain distribution of a data set and should be a small number compared with the total amount of data.

This system can naturally lead to a time-division multiplexing manner, such that each part of the training process can be operated separately in time using just one smaller system. This is especially advantageous when the number of qubits in a QA hardware is limited compared with the problem size. In fact, such a time-division multiplexing manner is equivalent to a dichotomy method, that is, by determining each bit of the binary label code, the total unlabeled data are sorted into two groups after each annealing process. An example of such a system is delineated in Fig. 2.

Moreover, we specify two configurations for labeled and unlabeled data separately:

Labeled data To assure that the qubits of labeled data reveal correct labels after being measured at the end of the annealing process, we should apply a bias h_i that is large enough to make the probability of their transition to wrong states close to 0 at the end of the QA process.

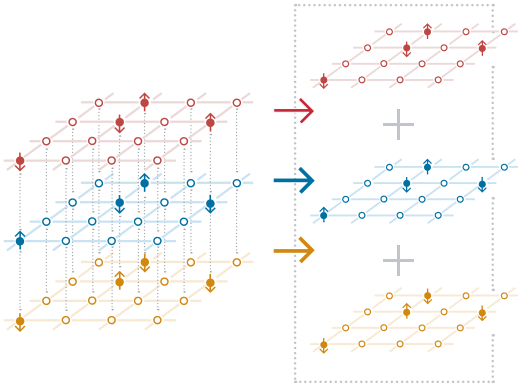


Fig. 2. Example of the QA structure that performs the SSL classification task. Each qubit, depicted in the solid or open circle, expresses one bit of the label code of a labeled or unlabeled data, respectively. A group of three qubits connected with a dashed line represents one data. Arrows on the labeled data indicate the directions of h_i on corresponding qubit q_i. In this example, each qubit in the same layer is topologically coupled with its 4 neighbors. A time-division multiplexing scheme can be used by dividing the system in to 3 smaller systems that are annealed individually.

Unlabeled data No bias is applied to the corresponding qubits.

Hence, Eq. (2) can be re-written as

$$H_{\text{tar}} = -\sum_{a=1}^{\alpha}\left(\sum_{i=1}^{l} h_i^{(a)}\sigma_i^{(a)} + \sum_{i,j=1}^{l+u} J_{ij}\sigma_i^{(a)}\sigma_j^{(a)} + \sum_{i=1}^{l+u}\sum_{j\in\text{emb}\{i\}} C_{ij}\sigma_i^{(a)}\sigma_j^{(a)}\right). \tag{3}$$

Here $\sigma_i^{(a)}$ is the Pauli Z operator on the i-th qubit at the a-th layer in the system. $\sum_{a=1}^{\alpha}\sum_{i}^{l} h_i^{(a)}\sigma_i^{(a)}$ is added as a constraint to ensure that the annealing results of labeled data are the same as their original labels. emb$\{i\}$ is the set of all qubits which are embedded to qubit i by term C_{ij}. When the value of C_{ij} is large enough, we can treat qubits i and j as one qubit as they will have the same state through out the annealing process. Thus, the number of qubits that are logically connected can be expanded compared with their actual physical connections. Here, C_{ij} can be a hyperparameter that depends on the hardware system. Figure 3 gives an example of this connecting method. Another example for mapping a graph to qubits connected by square lattices is shown in Fig. 4.

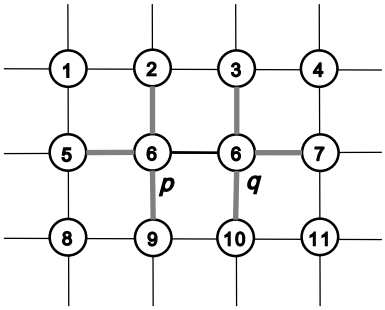


Fig. 3. An example of the connecting method that increases number of connections between qubits. Circles represent physical qubits and solid lines are physical couplings between two qubits. Each qubit is physically connected to its four surrounding qubits. The thicker lines represent a maximal coupling C_{pq} between qubits p and q, such that they could be treated as a single data qubit denoted as y_6. As a result, 6 qubits (i.e., $y_{2,3,5,7,9,10}$) are logically connected to y_6.

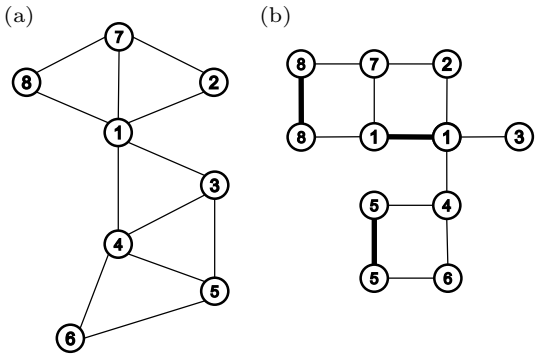


Fig. 4. Mapping a graph to qubits in square lattices. (a) The original graph to be mapped on a quantum annealer. (b) A way to connect physical qubits in square lattices to represent the graph shown in (a). The thick lines indicate that the qubits on the ends of the lines are maximally coupled.

In extreme cases, we can map an all-connected graph to a quantum annealer by King's graph as shown in Fig. 5.[36,37]

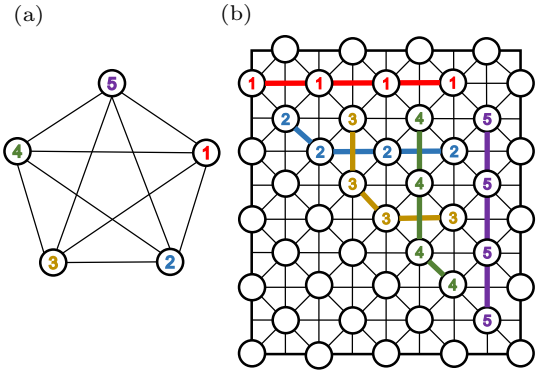


Fig. 5. (a) The original graph with full connection of 5 qubits. (b) An example for a connecting method on a quantum annealer with King's graph corresponding to the graph shown in (a).

After the annealing process, we can obtain the labels by measuring the corresponding qubits with the Pauli Z operator. The label we obtained is written as $y_i = \text{string}\, y_i^{(1)} \cdots y_i^{(\alpha)}$. In order to describe labels in quantum language, we use $y_i^{(a)} \in \{-1, +1\}$ rather than $\{0, 1\}$ which depends on the state after being measured denoted as $|y_i^{(a)}\rangle$, in which $|y_i^{(a)}\rangle\langle y_i^{(a)}| = \frac{1}{2}\left(1 + y_i^{(a)}\sigma_i^{(a)}\right)$.

2.3. Similarity and coupling parameters

In the QA model of Eq. (3), when $J_{ij} > 0$, the stronger the two qubits are coupled, the more likely they are to have the same orientations. Therefore, it is intuitive to map the similarity between two data to the coupling coefficient between two qubits in a QA system.

According to the vectors of two data in the manifold, the similarity between the two data can be calculated as below:

$$S_{ij} = \begin{cases} f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_p) & , \quad \text{if} \quad i \neq j, \\ 0 & , \quad \text{otherwise}, \end{cases} \tag{4}$$

where $\|\boldsymbol{\Theta}\|_p$ is the p-norm of vector $\boldsymbol{\Theta}$ and $f(\Theta)$ is a monotonically decreasing function of Θ. To better describe the similarities of a particular data set, $f(\Theta)$ may contain parameters that can be learned. For example, we can use Euclidean distance-based similarity

$$S_{ij} = \begin{cases} \frac{\beta_1}{\beta_2 + \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2} & , \quad \text{if} \quad i \neq j, \\ 0 & , \quad \text{otherwise}, \end{cases} \tag{5}$$

where β_1 and β_2 are parameters to be learned.

For those data sets that follow normal distributions in the manifold, we can also use the Gaussian-like function to describe the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as

$$S_{ij} = \begin{cases} \sum_{k=1}^{K} \frac{p(k)}{\sqrt{(2\pi)^d |\boldsymbol{B}_k|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{B}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)}, & \text{if } i \neq j, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

in which $\boldsymbol{B}_k$ is assumed to be a symmetric matrix whose elements are to be learned from the labeled data set $\mathcal{D}_l$, d is the dimension of the manifold, $p(k)$ is the proportion of the k-th label in a given labeled data set, i.e., $p(k) = n_k/l$ where n_k is the amount of data with the k-th labels.

We then map the similarity to the coupling parameter in a QA system. As mentioned above, a limited connectivity of a qubit with others in the system is usually favored to reduce the hardware complexity as well as to increase the accuracy. The latter is due to the consideration that an unlabeled data should be affected mainly by the surrounding ones in the manifold which are more likely to have smaller Hamming distance. The configuration of connection can be determined using a symmetric mask matrix $\boldsymbol{M}$ applied on the similarity matrix S. First, we search the ξ largest elements in row S_{i*} to find the ξ closest ones to data $\boldsymbol{x}_i$. A matrix $\boldsymbol{M}'$ is defined as

$$M'_{ij} = \begin{cases} 1 & , \text{ if } S_{ij} \text{ is among the } \xi \text{ largest of } S_{i*}, \\ 0 & , \quad \text{otherwise}. \end{cases} \tag{7}$$

The mask matrix is then calculated by $\boldsymbol{M} = (\boldsymbol{M}')^{\mathrm{T}}\mathrm{OR}(\boldsymbol{M}')$, in which $(\boldsymbol{A})\mathrm{OR}(\boldsymbol{B})$ means take bitwise OR operation between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. For each row of S_{ij}, find the ξ largest of S_{i*}, the time complexity is $O((l+u) \times \log \xi)$. The coupling parameters J_{ij} between two arbitrary qubits in Eq. (3) can be calculated by

$$J_{ij} = M_{ij} S_{ij}. \tag{8}$$

It should be noted that in this step, similarities between unlabeled data are also calculated, as we find out that the density information hidden in unlabeled data is also helpful during the QA process.

2.4. Parameters learning

In the final step, we attribute appropriate values to the parameters that are related to the system's Hamiltonian. Firstly, the parameters involved in the similarity calculation can be determined by a supervised learning process using the labeled data set. In the learning process, we have

$$H_{\text{learn}} = -\sum_{i=1}^{K} h_i \sigma_i - \sum_{a=1}^{\alpha} \sum_{i,j=1}^{l} J_{ij} \sigma_i^{(a)} \sigma_j^{(a)}, \tag{9}$$

where σ_i represents the i-th labeled data qubit in the labeled data set $\mathcal{D}_l$. The probability distribution of labels after the QA process follows the Boltzmann distribution

$$\begin{aligned} & p(y_1,\ldots,y_l|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_l;\boldsymbol{\theta}) \\ =&\frac{1}{Z}\exp\left(-\frac{E_{\text{learn}}(y_1,\ldots,y_l)}{T}\right) \\ =&\frac{1}{Z}\exp\left(-\frac{\sum_a^\alpha\sum_{i,j=1}^l\langle y_i^{(a)}y_j^{(a)}|H_{\text{learn}}|y_i^{(a)}y_j^{(a)}\rangle}{T}\right) \\ =&\frac{1}{Z}\exp\left(-\frac{-\sum_a^\alpha(\sum_i^K h_i\sigma_i^a+\sigma_i^a\sum_{i,j=1}^l J_{ij}y_i^{(a)}y_j^{(a)})}{T}\right), \end{aligned} \tag{10}$$

where E_{learn} is the system energy corresponding to H_{learn}, Z is the partition function, $Z=\text{Tr}[\exp\{-H_{\text{learn}}/T\}]$, and $\boldsymbol{\theta}$ is the vector containing all parameters to learn.

A negative log-likelihood function is therefore defined as below:

$$\begin{aligned} \mathcal{L}(\mathcal{D}_l;\boldsymbol{\theta}) &= -\log p(y_1,\ldots,y_l|\mathbf{x}_1,\ldots,\boldsymbol{x}_l;\boldsymbol{\theta}) \\ &= \frac{1}{Z}\left(-\frac{-\sum_a^\alpha(\sum_i^K h_i\sigma_i+\sum_{i,j=1}^l J_{ij}y_i^{(a)}y_j^{(a)})}{T}\right). \end{aligned} \tag{11}$$

Then the elements of $\boldsymbol{\theta}$ are determined by minimizing the conditional likelihood separately

$$\theta_j = \arg\min -\log p(lab_1,\ldots,lab_l|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_l;\boldsymbol{\theta}). \tag{12}$$

The iterative strategy is as follows:

$$\theta_j^{(t+1)} = \theta_j^{(t)} + \alpha\frac{\partial\mathcal{L}(\mathcal{D}_l;\boldsymbol{\theta})}{\partial\theta_j}, \tag{13}$$

in which α is the learning rate which controls the step of each round, and the gradient term can be easily calculated by sampling the annealing result. While the number of parameters is small, we can also traverse all the possible values.

Such a learning process is similar with the Boltzmann machine model,[33,38,39] except that the sampling process can be accelerated by iterated QA processes and project measurements of qubits.[17]

Again, some hyperparameters need to be tuned properly according to a specific data set and hardware system. For instance, the value of $h_i^{(a)}$ should be large enough such that the labels of labeled data are obtained correctly after the annealing process. ξ should be properly set to (i) ensure that every unlabeled data can reach to a labeled one through a connected path to avoid a random result and to (ii) reduce the implication of data that are far away in the manifold.

3. Example

Here we give two examples based on realistic database to verify the method discussed above. As a proof-of-principle demonstration, the annealing processes are simulated by a classical computer. It should be noted that a quantum annealer may exhibits control errors such that the actual connection coefficient is not exactly what we have calculated. So when we simulate the protocol on the classical computer, we add a random disturb about 3% on $\{h_i\}$, $\{J_{ij}\}$, and $\{C_{ij}\}$.

3.1. Example 1: iris

We first use a database of iris that has been widely used in pattern recognition literature.[40] There are three kinds of label in the data set, shown by points in three colors in Fig. 6(a). According to the labeled data (open circles), it is obvious that the shortest path that connects all the labels' barycenters is green–red–blue. Therefore, we encode the label by an ordered binary gray code as $\{00\}_{\text{Setosa}}$, $\{01\}_{\text{Versicolour}}$, and $\{10,11\}_{\text{Virginica}}$. We assume that the similarity between arbitrary two data follows a 2-dimensional mixed Gaussian-like function

$$\begin{aligned} S_{ij} = \sum_{k=1}^3 \frac{p(k)}{2\left[1-\left(\rho^{(k)}\right)^2\right]}\Bigg[&\frac{x_{i(1)}-x_{j(1)}}{\left(\beta_1^{(k)}\right)^2}+\frac{x_{i(2)}-x_{j(2)}}{\left(\beta_2^{(k)}\right)^2} \\ &-\frac{2\rho^{(k)}(x_{i(1)}-x_{j(1)})(x_{i(2)}-x_{j(2)})}{\beta_1^{(k)}\beta_2^{(k)}}\Bigg], \end{aligned} \tag{14}$$

where $x_{i(1)}$ ($x_{i(2)}$) is the first (second) element of data $\boldsymbol{x}_i$, and $\rho^{(l)}$ is the correlation coefficient between $\boldsymbol{x}_{i(1)}$ and $\boldsymbol{x}_{i(2)}$. In this example, the data have been processed with PCA,[41] so we can assume that $\rho^{(1,2,3)}=0$. J_{ij} are calculated using Eq. (8) with $\xi=6$. In the simulation, we set the bias $\{h_i\}$ and the embedding coefficient $\{C_{ij}\}$ to 10. During the simulation of this, we find out that the classification result is not sensitive to the parameters to be learned in similarity calculation. So long as the similarity is monotonically increasing with distance, the classification results are consistent. In the meantime, some hyperparameters, such as the number of connections of data and its surrounding data, have significantly impacts on the classification results.

In Eq. (14), $\beta_*^{(k)}$ are the parameters to be learned using the data of the k-th label. For each k, we substitute J_{ij} into Eqs. (11) and (10) to have

$$\begin{aligned} \mathcal{L}\left(\mathcal{D}_l;\beta_*^{(k)}\right) &= -\log p\left(lab_1,\ldots,lab_l|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_l;\beta_*^{(k)}\right) \\ &= \frac{1}{lZ(\theta)}\sum_a^\alpha\sum_{i,j=1}^l\sum_{k=1}^3\frac{p(k)}{2} \\ &\quad\times\left[\frac{x_{i(1)}-x_{j(1)}}{\left(\beta_1^{(k)}\right)^2T}+\frac{x_{i(2)}-x_{j(2)}}{\left(\beta_2^{(k)}\right)^2T}\right]lab_i^{(a)}lab_j^{(a)}, \end{aligned} \tag{15}$$

in which $lab_i^{(a)}$ is the a-th bit of lab_i with binary encoding.

The classification results are shown in Figs. 6(b)–6(d). When 30% of the data set is unlabeled, the accuracy of the algorithm is 100%. An accuracy of 94.26% can still be maintained when 80% unlabeled data is considered.

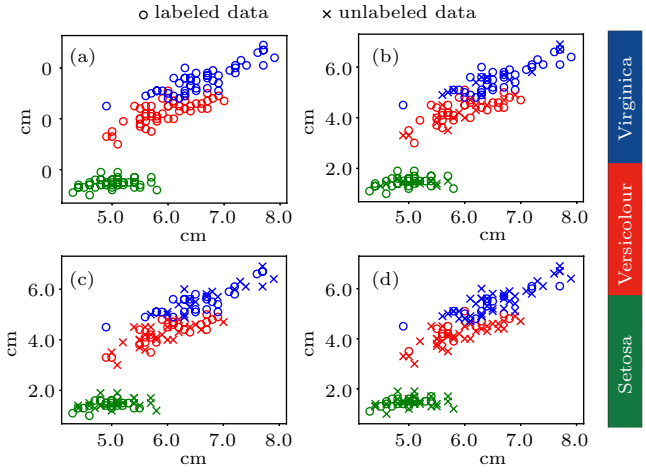


Fig. 6. The original iris data set (a) and the classification results using the algorithm proposed in this work when the portion of unlabeled is (b) 30% with 97.89% accuracy rate, (c) 50% with 94.44% accuracy rate, and (d) 80% with 96.26% accuracy rate. The circles in the picture represent labeled data and the crosses represent unlabeled data. The y axis of the graph is the petal length in cm and the x axis is the sepal length in cm.

3.2. Example 2: handwriting digital pictures

The second example is the handwritten digital recognition using the database from MNIST. We pick 250 pieces of 8×8 pixels images of digits 0, 4, 7, 8, and 9 from the original data set and reduce the original dimension to 2 by Isomap function as shown in Fig. 7(a). According to their barycenters on the manifold, we encode the 4 labels by $\{000, 100\}_0$ (blue), $\{001\}_4$(red), $\{011, 010\}_7$ (yellow), $\{111, 101\}_8$ (purple), and $\{110\}_9$ (green).

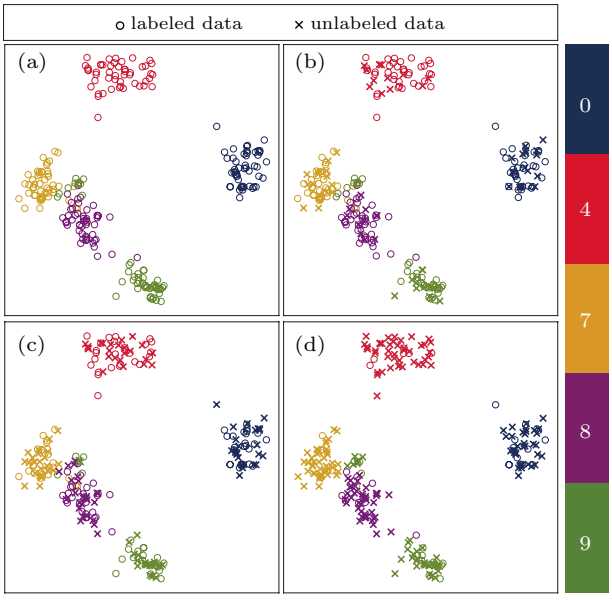


Fig. 7. The handwriting digits data set with reduced dimensions (a) and the digits 0, 4, 7, 8, 9 in blue, red, yellow, purple, and green, respectively. The classification results using the algorithm proposed in this work when the portion of unlabeled is (b) 30% with 98.55% accuracy rate, (c) 50% with 95.9% accuracy rate, and (d) 80% with 97.04% accuracy rate. The circuits in the picture represent labeled data and the crosses represent unlabeled data.

Here the Euclidean distance given by Eq. (5) is applied to calculate the similarity matrix S and coupling parameters J, in which $\xi = 4$ for 30% and 50% unlabeled and $\xi = 7$ for 80% unlabeled data. In the simulation, we set the bias $\{h_i\}$ to 10. The parameters concerning the similarity calculation are trained using similar approaches as the first example.

Figures 7(b)–7(d) show the classification results. The accuracy of QA-SSL changes from 96.15% to 92.13% as the portion of the unlabeled data in the whole data set increases from 30% to 80%, showing again the feasibility of this method.

4. Discussion

As mentioned above, in classical graph-based semi-supervised learning, the time complexity is at least $\mathcal{O}(n^3)$, where n is the size of the data set. To analyze the time complexity of our algorithm, we divide it into four parts. (i) Encoding label with gray codes: the time complexity of label encoding is only related to the label number (the exactly time complexity should equal to that of the classical sorting algorithm), which, in most cases, is much smaller than n, and therefore negligible compared with $\mathcal{O}(n^3)$. (ii) Parameter training in similarity calculation: only labeled data is used during this procedure and the number of parameters is only related to the dimension of the manifold, which are both much smaller compared with the size of the data set. Therefore, we believe that the time complexity of training the parameters is also negligible compared with that of the classical SSL algorithm. (iii) Finding the most ξ similar data around an arbitrary data: this procedure is equivalent to find the ξ smallest values in n data and has a time complexity of $\mathcal{O}(n \log \xi)$, which is small compared with $\mathcal{O}(n^3)$; (iv) QA: the computation time of QA is $\mathcal{O}(1/(\Delta e^2))$, where Δe is the energy difference between ground state and the first excited state of the system. Therefore, the time complexity of the QA process is a constant and independent of the amount of data. From the analysis above, we believe our algorithm has obvious advantages over classical graph-based semi-supervised learning when processing a large amount of data.

5. Conclusion

So far, quantum machine learning algorithms have been studied extensively on clustering (unsupervised learning)[34,42–44] or supervised learning classification algorithms.[11,45] In this paper we introduce a new semi-supervised learning method based on QA. In this method, the classification problem is mapped to the QA Hamiltonian through a graph representation, of which the vertices are efficiently implemented by qubits with an encoding scheme based on a binary gray code. Calculations of the similarity between data are improved with a learning process using various models. Compared with previous proposed classification method using QA, this scheme significantly saves the quantum resources while maintaining the ability to express the

original problem. The results of two proof-of-principle examples indicate that this method can still yield high accuracy for classification problem when the amount of labeled data is limited.

References

[1] Michie D, Spiegelhalter D J and Taylor C C 1994 *Machine Learning, Neural and Statistical Classification* (New Jersey: Ellis Horwood) pp. 1–6

[2] Christopher M B 2006 *Pattern recognition and machine learning* (New York: Springer) pp. 1–4

[3] Steane A 1998 *Rep. Prog. Phys.* **61** 117

[4] Hirvensalo M 2013 *Quantum Computing* (Springer Science & Business Media) pp. 49–71

[5] Shor P 1994 *Proceedings 35th annual symposium on foundations of computer science, IEEE*, pp. 124–134

[6] Zhang S, Duan Q H, Li T, Fu X Q, Huang H L, Wang X and Bao W S 2020 *Chin. Phys. B* **29** 010308

[7] Zheng S B 2005 *Chin. Phys.* **14** 2222

[8] Schuld M, Sinayskiy I and Petruccione F 2015 *Contemp. Phys.* **56** 172

[9] Jeremy A, Euan A, Matthew D, Stefan F, Janna H, Mack J, Sam M S, Sam P, Alasdair P and Stasja S 2015 arXiv:1512.02900 [quantu-ph]

[10] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195

[11] Rebentrost P, Mohseni M and Lloyd S 2014 *Phys. Rev. Lett.* **113** 130503

[12] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631

[13] Gao X, Zhang Z Y and Duan L M 2019 *Sci. Adv.* **4** eaat9004

[14] Preskill J 2018 *Quantum* **2** 79

[15] Neukart F, Compostella G, Seidel C, Von Dollen D, Yarkoni S and Parney B 2017 *Front. ICT* **4** 29

[16] Babbush R, Love P J and Aspuru-Guzik A 2014 *Sci. Rep.* **4** 6603

[17] Adachi S H and Henderson M P 2015 arXiv:1510.06356 [quant-ph]

[18] Chapelle O, Scholkopf B and Zien A 2009 *IEEE Trans. Nerual Netw. Learn. Syst.* **20** 542

[19] Zhu X and Goldberg A B 2009 *Synth. Lect. Artif. Intell. Mach. Learn.* **3** 1

[20] Zhou X and Belkin M 2014 *Academic Press Library in Signal Processing, Volume 1: Signal Processing Theory and Machine Learning, Chapter 22 - Semi-Supervised Learning* (Amsterdam: Elsevier) pp. 1239–1269

[21] Zha Z J, Mei T, Wang J D and Hua X S 2009 *J. Vis. Commun. Image Represent.* **20** 97

[22] Roweis S T and Saul L K 2000 *Science* **290** 2323

[23] Kambhatla N and Leen T K 1997 *Neural Comput.* **9** 1493

[24] Fergus R, Weiss Y and Torralba A 2009 *Advances in neural information processing systems 22*, December 7–10, 2009, Vancouver, British Columbia, Canada, pp. 522–530

[25] Liu W, He J and Chang S F 2010 *The 27th International Conference on Machine Learning*, June 21–24, 2010, Haifa, Israel, 2010, pp. 1–8

[26] Zhou D, Bousquet O, Lal T N and Schölkopf B 2004 *Conference and Workshop on Neural Information Processing Systems* December 13–18, 2004, Vancouver, British Columbia, Canada, pp. 595–602

[27] Kadowaki T and Nishimori H 1998 *Phys. Rev. E* **58** 5355

[28] Bunyk P I, Hoskinson E M, Johnson M W, Tolkacheva E, Altomare F, Berkley A J, Harris R, Hilton J P, Lanting T, Przybysz A J and Whittaker J 2014 *IEEE Trans. Appl. Supercond.* **24** 1

[29] Boixo S, Albash T, Spedalieri F M, Chancellor N and Lidar D A 2013 *Nat. Commun.* **4** 2067

[30] Boixo S, Rønnow T F, Isakov S V, Wang Z, Wecker D, Lidar D A, Martinis J M and Troyer M 2014 *Nat. Phys.* **10** 218

[31] Ray P, Chakrabarti B K and Chakrabarti A 1989 *Phys. Rev. B* **39** 11828

[32] Van Dam W, Mosca M and Vazirani U 2001 *In Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, October 8–11, 2001, Newport Beach, CA, USA, pp. 279–287

[33] Ulanov A E, Tiunov E S and Lvovsky A 2019 arXiv:1912.08480 [quant-ph]

[34] Kumar V, Bass G, Tomlin C and Dulny J 2018 *Quantum Inf. Process.* **17** 39

[35] Martoňák R, Santoro G E and Tosatti E 2004 *Phys. Rev. E* **70** 057701

[36] Tanaka S, Matsuda Y and Togawa N 2020 *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, January 13–16, 2020, Beijing, China, pp. 659–666

[37] Choi V 2008 *Quantum Inf. Process.* **7** 193

[38] Amin M H, Andryash E, Rolfe J, Kulchytskyy B and Melko R 2018 *Phys. Rev. X* **8** 021050

[39] Hinton G E and Sejnowski T J 1986 *Parallel distributed processing: Explorations in the microstructure of cognition* (Cambridge: MIT Press) **1** 282–317

[40] Fisher R A 1936 *Ann. Eugen.* **7** 179

[41] Kessy A, Lewin A and Strimmer, K 2018 *Am. Stat.* **72** 309

[42] Horn D and Gottlieb A 2001 *Phys. Rev. Lett.* **88** 018702

[43] Roitblat H L and Golbére B (U.S. Patent) 8 010 534 [2011-08-30]

[44] Kurihara K, Tanaka S and Miyashita S 2014 arXiv:1408.2035 [quant-ph]

[45] Li Z, Liu X, Xu N and Du J 2015 *Phys. Rev. Lett.* **114** 140504