SCIENCE CHINA
Information Sciences

• LETTER •

August 2020, Vol. 63 189501:1–189501:3
https://doi.org/10.1007/s11432-019-2783-7

# Quantum speedup of twin support vector machines

Zekun YE[1,3], Lvzhou LI[1,4*], Haozhen SITU[2,3] & Yuyi WANG[1]

[1]*School of Data and Computer Science, Sun Yat-Sen University, Guangzhou* 510006*, China;*
[2]*College of Mathematics and Informatics, South China Agricultural University, Guangzhou* 510642*, China;*
[3]*Center for Quantum Computing, Peng Cheng Laboratory, Shenzhen* 518055*, China;*
[4] *Ministry of Education Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-Sen University), Guangzhou* 510006*, China*

Received 31 August 2019/Revised 16 January 2020/Accepted 6 February 2020/Published online 23 April 2020

**Citation** Ye Z K, Li L Z, Situ H Z, et al. Quantum speedup of twin support vector machines. Sci China Inf Sci, 2020, 63(8): 189501, https://doi.org/10.1007/s11432-019-2783-7

Dear editor,
Machine learning enables agents or computers to learn from data and improves the performance of specific tasks without requiring explicit programming. As the amount of available data continues to grow, existing machine learning algorithms cannot satisfy the efficiency requirements of various practical applications, and several methods have been proposed to solve this problem, e.g., distributed computing. However, distributed computing, even in the best case, cannot achieve exponential acceleration. Additionally, there are typically extra communication costs. Quantum computers are not yet available for practical applications; however, efficient quantum algorithms have been proposed to address machine learning problems.

In this study, we provide further evidence that quantum computing has the potential to increase the speed of machine learning algorithms in the future. The basic unit of quantum information, i.e., qubits, could be in superposition states; quantum computers are expected to be very advantageous for processing high-dimensional data. Thus, the performance of certain machine learning models and algorithms could be significantly improved using quantum computers in terms of time and space complexity.

We devise new quantum algorithms that exponentially speeds up the training and prediction procedures of twin support vector machines (TSVM). To train TSVMs using quantum methods, we demonstrate how to prepare the desired input states according to classical data, and these states are used in the quantum algorithm for the system of linear equations. In the prediction process, we employ a quantum circuit to estimate the distances from a new sample to the hyperplanes and then make a decision. The proposed quantum algorithms can learn two non-parallel hyperplanes and classify a new sample by comparing the distances from the sample to the two hyperplanes in $O(\log mn)$ time, where $m$ is the sample size and $n$ is the dimension of each data point. In contrast, the corresponding classical algorithm requires polynomial time for both the training and prediction procedures.

*TSVM model.* In the binary classification problems, there are $m$ training examples sampled from some fixed but unknown distribution, including $m_1$ positive examples and $m_2$ negative examples, where each training example is a real vector. Herein, we use a matrix $A \in \mathbb{R}^{m_1\times n}$ to represent examples of the positive class, and each matrix row represents a positive example. Similarly, a matrix $B \in \mathbb{R}^{m_2\times n}$ represents examples of the negative class. When a new example, $\boldsymbol{x}$, arrives from the same distribution, we can classify it using information extracted from the training examples. TSVMs are widely used for binary classification models that attempt to find two non-parallel hyperplanes:

$$\boldsymbol{w}_1 \cdot \boldsymbol{x} + b_1 = 0, \boldsymbol{w}_2 \cdot \boldsymbol{x} + b_2 = 0, \tag{1}$$

* Corresponding author (email: lilvzh@mail.sysu.edu.cn)

 

where $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$, such that the positive samples are as close as possible to the first hyperplane and distant from the second. Negative samples are as close as possible to the second hyperplane and distant from the first [1]. Formally, TSVMs solve two quadratic programming problems with an objective function corresponding to one class and constraints corresponding to the other class. Compared with traditional SVMs, TSVMs are faster and have better generalizability. Additionally, to simplify quadratic programming problems, inequality constraints are replaced by equality constraints in least square TSVMs [2]:

$$\min_{\boldsymbol{w}_1,b_1} \quad \frac{1}{2}||A\boldsymbol{w}_1 + b_1\boldsymbol{e}_1||^2 + \frac{1}{2}c_1||\boldsymbol{\xi}_2||^2, \tag{2}$$

$$\text{s.t.} \quad -(B\boldsymbol{w}_1 + b_1\boldsymbol{e}_2) + \boldsymbol{\xi}_2 = \boldsymbol{e}_2, \tag{3}$$

$$\min_{\boldsymbol{w}_2,b_2} \quad \frac{1}{2}||B\boldsymbol{w}_2 + b_2\boldsymbol{e}_2||^2 + \frac{1}{2}c_2||\boldsymbol{\xi}_1||^2, \tag{4}$$

$$\text{s.t.} \quad (A\boldsymbol{w}_2 + b_2\boldsymbol{e}_1) + \boldsymbol{\xi}_1 = \boldsymbol{e}_1, \tag{5}$$

where $c_1, c_2$ are penalty parameters, $\boldsymbol{e}_1, \boldsymbol{e}_2$ are all-one column vectors, and $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2$ are non-negative slack variables. Here, let $L = \min_{\boldsymbol{w}_1,b_1} \frac{1}{2}||A\boldsymbol{w}_1 + b_1\boldsymbol{e}_1||^2 + \frac{1}{2}c_1||\boldsymbol{\xi}_2||^2$. By substituting the equality constraint (3) into the objective function (2), we obtain $L = \min_{\boldsymbol{w}_1,b_1} \quad \frac{1}{2}||A\boldsymbol{w}_1 + b_1\boldsymbol{e}_1||^2 + \frac{1}{2}c_1||B\boldsymbol{w}_1 + b_1\boldsymbol{e}_2 + \boldsymbol{e}_2||^2$. In addition, by setting the partial derivatives of the function to zero, we obtain the following:

$$A^{\mathrm{T}}(A\boldsymbol{w}_1 + b_1\boldsymbol{e}_1) + c_1B^{\mathrm{T}}(B\boldsymbol{w}_1 + b_1\boldsymbol{e}_2 + \boldsymbol{e}_2) = 0,$$
$$\boldsymbol{e}_1{}^{\mathrm{T}}(A\boldsymbol{w}_1 + b_1\boldsymbol{e}_1) + c_1\boldsymbol{e}_2{}^{\mathrm{T}}(B\boldsymbol{w}_1 + b_1\boldsymbol{e}_2 + \boldsymbol{e}_2) = 0,$$

which can be rewritten as follows:

$$\begin{pmatrix} \boldsymbol{w}_1 \\ b_1 \end{pmatrix} = -(\frac{1}{c_1}E^{\mathrm{T}}E + F^{\mathrm{T}}F)^{-1}F^{\mathrm{T}}\boldsymbol{e}_2, \tag{6}$$

where $E = [A \ \boldsymbol{e}_1], F = [B \ \boldsymbol{e}_2]$. Here, similarly, we obtain the following:

$$\begin{pmatrix} \boldsymbol{w}_2 \\ b_2 \end{pmatrix} = (E^{\mathrm{T}}E + \frac{1}{c_2}F^{\mathrm{T}}F)^{-1}E^{\mathrm{T}}\boldsymbol{e}_1. \tag{7}$$

Matrix multiplication requires at least $\Omega(n^2)$ time to calculate $F^{\mathrm{T}}F$ and $E^{\mathrm{T}}E$, find the inverse of $\frac{1}{c_1}E^{\mathrm{T}}E + F^{\mathrm{T}}F$ and $E^{\mathrm{T}}E + \frac{1}{c_2}F^{\mathrm{T}}F$, and calculate $F^{\mathrm{T}}\boldsymbol{e}_2$ and $E^{\mathrm{T}}\boldsymbol{e}_1$. Thus, the total running time required to solve the two hyperplanes in Eqs. (2)–(5) is $\Omega(n^2)$. To classify a new instance, we must calculate the distance from the new instance $\boldsymbol{x}$ to the two hyperplanes $|\boldsymbol{w}_1\boldsymbol{x} + b_1|/||\boldsymbol{w}_1||$ and $|\boldsymbol{w}_2\boldsymbol{x} + b_2|/||\boldsymbol{w}_2||$, and then compare them. Note that this step requires $\Omega(n)$ time.

*Quantum Algorithms for TSVMs.* Herein, we present the proposed quantum algorithms to increase the speed of the training and prediction procedures of TSVMs (Appendix A provides some background knowledge of quantum computation). Without loss of generality, we assume $||\binom{\boldsymbol{w}_i}{b_i}|| = 1$ for $i = 1, 2$, where $||\cdot||$ denotes the 2-norm of vectors. We introduce the following notations:

$$|F^{\mathrm{T}}\boldsymbol{e}_2\rangle = \frac{F^{\mathrm{T}}\boldsymbol{e}_2}{||F^{\mathrm{T}}\boldsymbol{e}_2||}, |E^{\mathrm{T}}\boldsymbol{e}_1\rangle = \frac{E^{\mathrm{T}}\boldsymbol{e}_1}{||E^{\mathrm{T}}\boldsymbol{e}_1||},$$
$$|\boldsymbol{w}_1, b_1\rangle = \begin{pmatrix} \boldsymbol{w}_1 \\ b_1 \end{pmatrix}, |\boldsymbol{w}_2, b_2\rangle = \begin{pmatrix} \boldsymbol{w}_2 \\ b_2 \end{pmatrix}.$$

For $\boldsymbol{x} = (x_0, ..., x_{n-1})$, we define $\tilde{\boldsymbol{x}} = (x_0, ..., x_{n-1}, 1)$ and $|\tilde{\boldsymbol{x}}\rangle = \frac{1}{\sqrt{N_{\tilde{\boldsymbol{x}}}}}(\sum_{i=0}^{n-1} x_i|i\rangle + 1|n\rangle)$, where $N_{\tilde{x}} = \sum_{i=0}^{n-1} x_i^2 + 1$. Then, we rewrite Eqs. (6) and (7) in the quantum setting as follows:

$$|\boldsymbol{w}_1, b_1\rangle = \frac{(\frac{1}{c_1}E^{\mathrm{T}}E + F^{\mathrm{T}}F)^{-1}|F^{\mathrm{T}}\boldsymbol{e}_2\rangle}{||(\frac{1}{c_1}E^{\mathrm{T}}E + F^{\mathrm{T}}F)^{-1}|F^{\mathrm{T}}\boldsymbol{e}_2\rangle||}, \tag{8}$$

$$|\boldsymbol{w}_2, b_2\rangle = \frac{(E^{\mathrm{T}}E + \frac{1}{c_2}F^{\mathrm{T}}F)^{-1}|E^{\mathrm{T}}\boldsymbol{e}_1\rangle}{||(E^{\mathrm{T}}E + \frac{1}{c_2}F^{\mathrm{T}}F)^{-1}|E^{\mathrm{T}}\boldsymbol{e}_1\rangle||}. \tag{9}$$

In the training procedure, we prepare quantum states $|F^{\mathrm{T}}\boldsymbol{e}_2\rangle$ and $|E^{\mathrm{T}}\boldsymbol{e}_1\rangle$, and then solve Eqs. (8) and (9) by calling the quantum linear system algorithm [3] and saving the solutions in quantum states $|\boldsymbol{w}_1, b_1\rangle$ and $|\boldsymbol{w}_2, b_2\rangle$ that represent two hyperplanes. In the classification procedures, we represent a new sample $\boldsymbol{x}$ as a quantum state $|\boldsymbol{x}, 1\rangle$ and obtain the distances between the sample and hyperplanes by estimating the inner product of $|\boldsymbol{x}, 1\rangle$ with $|\boldsymbol{w}_1, b_1\rangle$ and $|\boldsymbol{w}_2, b_2\rangle$, respectively.

The algorithmic procedures are shown in Algorithms 1 and 2, which are explained in detail as follows.

**Algorithm 1** QTSVM training process.

**Input:** $m_1$ positive samples and $m_2$ negative samples represented by matrices $A$ and $B$, where $A \in \mathbb{R}^{m_1 \times n}$ and $B \in \mathbb{R}^{m_2 \times n}$.

**Procedure:**

1. Prepare input quantum states $|F^{\mathrm{T}}\boldsymbol{e}_2\rangle$ and $|E^{\mathrm{T}}\boldsymbol{e}_1\rangle$, where $E = [A \ \boldsymbol{e}_1]$ and $F = [B \ \boldsymbol{e}_2]$.
2. Use the quantum algorithm for systems of linear equations as a subroutine to solve the linear equations shown in Eqs. (8) and (9), and obtain quantum states $|\boldsymbol{w}_1, b_1\rangle$ and $|\boldsymbol{w}_2, b_2\rangle$ that represent two hyperplanes.

**Output:** Quantum states $|\boldsymbol{w}_1, b_1\rangle$ and $|\boldsymbol{w}_2, b_2\rangle$ that represent two hyperplanes.

**Algorithm 2** QTSVM predicting process.

**Input:** Quantum states $|\boldsymbol{w}_1, b_1\rangle$ and $|\boldsymbol{w}_2, b_2\rangle$ that represent two hyperplanes. A new sample $\boldsymbol{x} \in \mathbb{R}^n$.

**Procedure:**

1. Prepare new sample $\boldsymbol{x}$ as a quantum state $|\boldsymbol{x}, 1\rangle$.
2. Use the SWAP test to find the distances from $\boldsymbol{x}$ to two hyperplanes respectively and compare them.
3. If $\boldsymbol{x}$ is closer to the first hyperplane, then label it positive; otherwise negative.

**Output:** The label of $\boldsymbol{x}$.

Additionally, Theorem 1 gives the time complexity of the proposed algorithms (Appendix D provides the proof of Theorem 1).

**Theorem 1.** The time complexities of Algorithms 1 and 2 are $O(\log mn)$ and $O(\log n)$, respectively, where $m = m_1 + m_2$.

Therefore, for a new sample, we first use Algorithm 1 to learn the two hyperplanes given by Eq. (1) and then use Algorithm 2 to classify it. The total time complexity is $O(\log mn)$.

*Training process.* First, we provide some notations for Algorithm 1. Let $K_1 = E^{\mathrm{T}}E, K_2 = F^{\mathrm{T}}F, H_1 = \frac{1}{c_1}K_1 + K_2, H_2 = K_1 + \frac{1}{c_2}K_2$, and $\hat{K}_1 = \frac{K_1}{tr(K_1)}, \hat{K}_2 = \frac{K_2}{tr(K_2)}, \hat{H}_1 = \frac{H_1}{tr(H_1)}, \hat{H}_2 = \frac{H_2}{tr(H_2)}$. Note that Appendix B provides the method to prepare quantum states $|F^T\boldsymbol{e}_2\rangle$ and $|E^{\mathrm{T}}\boldsymbol{e}_1\rangle$ based on quantum oracles [4]. In the quantum setting, it assumes that oracles for the training data return corresponding quantum states. One way to efficiently construct these states is via quantum RAM that uses $O(mn)$ hardware resources but only $O(\log mn)$ operations to access them [5].

To apply the quantum algorithm for systems of linear equations to solve Eqs. (6) and (7), $H_1$ and $H_2$ must be exponentiated efficiently. In other words, it is an important subroutine to calculate $e^{-i\hat{H}_1\Delta t}$ and $e^{-i\hat{H}_2\Delta t}$ efficiently, where $\hat{H}_1$ and $\hat{H}_2$ are the normalization of $H_1$ and $H_2$, respectively, and $\Delta t$ is a small time slice. Unlike the original method in the quantum algorithm for systems of linear equations, we employ a density matrix exponentiation method [6] herein that allows us to perform Hamiltonian simulation effectively even if the samples do not satisfy the sparsity assumption. Details are given in Appendix C. We then obtain the two hyperplanes in the form of quantum states $|\boldsymbol{w}_1, b_1\rangle$ and $|\boldsymbol{w}_2, b_2\rangle$ by calling the quantum algorithm for the systems of linear equations [3].

*Prediction process.* In Algorithm 2, given a new sample $\boldsymbol{x} \in \mathbb{R}^n$, we determine its label by comparing the distances from it to the two obtained hyperplanes. By calling the oracle to the vector $\tilde{\boldsymbol{x}}$, we can construct state $|\tilde{\boldsymbol{x}}\rangle$. Using the SWAP test [7], we then estimate the square of inner product $I = \langle \boldsymbol{w}_1, b_1|\tilde{\boldsymbol{x}}\rangle^2 = \frac{|\boldsymbol{w}_1\boldsymbol{x}+b_1|^2}{N_{\tilde{\boldsymbol{x}}}}$ by using $O(\epsilon^{-2})$ copies of $|\boldsymbol{w}_1, b_1\rangle$ and $|\tilde{\boldsymbol{x}}\rangle$ such that the error of the inner product estimation is no greater than $\epsilon$ with high probability.

Additionally, to estimate the value of $||\boldsymbol{w}_1||^2$, we employ a project measurement operations set $\{P_i\}$ to measure $|\boldsymbol{w}_1, b_1\rangle$, where $P_0 = I - |n\rangle\langle n|, P_1 = |n\rangle\langle n|$. Then, the probability of obtaining outcome 0 is $||\boldsymbol{w}_1||^2$. By measuring $O(\epsilon^{-2})$ copies of $|\boldsymbol{w}_1, b_1\rangle$ repeatedly, we can estimate the value of $||\boldsymbol{w}_1||^2$ no more than $\epsilon$ with high probability. Similarly, we can estimate the value of $||\boldsymbol{w}_2||^2$.

We can obtain the value of $|\boldsymbol{w}_1\boldsymbol{x} + b_1|^2/||\boldsymbol{w}_1||^2$ because the values of $|\boldsymbol{w}_1\boldsymbol{x} + b_1|^2$ and $||\boldsymbol{w}_1||^2$ have been estimated. Note that the value of $|\boldsymbol{w}_2\boldsymbol{x} + b_2|^2/||\boldsymbol{w}_2||^2$ can be obtained in a similar manner. If $|\boldsymbol{w}_1\boldsymbol{x} + b_1|^2/||\boldsymbol{w}_1||^2 < |\boldsymbol{w}_2\boldsymbol{x} + b_2|^2/||\boldsymbol{w}_2||^2$, then the sample is closer to the first hyperplane, and it will be labeled as a positive point; otherwise, it will be labeled negative.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (Grant No. 61772565), the Natural Science Foundation of Guangdong Province (Grant No. 2017A030313378), the Science and Technology Program of Guangzhou City (Grant No. 201707010194), and Key R & D Project of Guangdong Province (Grant No. 2018B030325001).

**Supporting information** Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

**References**

1 Jayadeva, Khemchandani R, Chandra S. Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell, 2007, 29: 905–910
2 Arun Kumar M, Gopal M. Least squares twin support vector machines for pattern classification. Expert Syst Appl, 2009, 36: 7535–7543
3 Harrow A W, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. Phys Rev Lett, 2009, 103: 150502
4 Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. Phys Rev Lett, 2014, 113: 130503
5 Giovannetti V, Lloyd S, Maccone L. Quantum random access memory. Phys Rev Lett, 2008, 100: 160501
6 Lloyd S, Mohseni M, Rebentrost P. Quantum principal component analysis. Nat Phys, 2014, 10: 631–633
7 Buhrman H, Cleve R, Watrous J, et al. Quantum fingerprinting. Phys Rev Lett, 2001, 87: 167902