

Quantum expectation-maximization algorithm

Hideyuki Miyahara,^{1,*} Kazuyuki Aihara^{2,3} and Wolfgang Lechner⁴

¹*Department of Mathematical Informatics, Graduate School of Information Science and Technology,
The University of Tokyo, Tokyo 113-8656, Japan*

²*Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan*

³*International Research Center for Neurointelligence (WPI-IRC), UTIAS, The University of Tokyo, Tokyo 113-0033, Japan*

⁴*Institute for Theoretical Physics, University of Innsbruck, A-6020 Innsbruck, Austria*



(Received 21 August 2019; published 17 January 2020)

Clustering algorithms are a cornerstone of machine learning applications. Recently, a quantum algorithm for clustering based on the k -means algorithm has been proposed by Kerenidis, Landman, Luongo, and Prakash. Based on their work, we propose a quantum expectation-maximization algorithm for Gaussian mixture models (GMMs). The robustness and quantum speedup of the algorithm are shown. We also show numerically the advantage of GMM over k -means algorithm for nontrivial cluster data.

DOI: [10.1103/PhysRevA.101.012326](https://doi.org/10.1103/PhysRevA.101.012326)

I. INTRODUCTION

Quantum computing has attracted much attention since the discovery of Shor's algorithm [1,2]. Recently, with the rapid developments in machine learning, physicists have started to consider utilizing quantum computers for machine learning applications [3–8]. As a result, quantum machine learning has emerged as an interdisciplinary field between quantum computing and machine learning. Furthermore, a quantum algorithm for the k -means algorithm [9,10] with proven quantum speedup was proposed [11].

The k -means algorithm is an essential tool in many machine learning applications [9,10]. However, the k -means algorithm is as a special case of the more general Gaussian mixture model (GMM). In the k -means algorithm, each Gaussian has the same weight and the covariance matrix of each Gaussian function is the identity. As a result, the k -means algorithm may provide poor estimates of the clusters since the assumptions of the k -means algorithm are sometimes too strong to capture all properties of complex data sets. The expectation-maximization (EM) algorithm [9,10,12] and variational Bayes (VB) inference [9,10] with the GMM are often used to improve the clustering, since the general GMM can deal with a wider class of data sets. Recently, one of the authors proposed quantum-inspired algorithms for the EM algorithm [13–15] and VB [16]. In Refs. [15,16], we have succeeded in improving the performances of the EM algorithm and VB. However, the aim of Refs. [15,16] is to make use of quantum fluctuations as a numerical tool and not to provide a quantum speedup over a classical algorithm; as a result, the computational costs are almost the same.

In this paper, we propose a quantum algorithm to estimate the parameters of the GMM, which we call the quantum EM (q-EM) algorithm. To this end, following the spirit of

Ref. [11], we first introduce a randomized variant of the EM algorithm, i.e., the δ -EM algorithm which includes nondeterministic readout of the data. Then, we formulate a quantum algorithm that realizes a speedup of the δ -EM algorithm with respect to the number of data points. The q-EM algorithm may be an important step toward quantum machine learning, since the EM algorithm is an essential algorithm in machine learning.

This paper is organized as follows. In Sec. II, we provide classical preliminaries. In particular, we review the EM algorithm and then introduce the δ -EM algorithm. In Sec. III, we present the detailed procedure of the q-EM algorithm. Then, in Sec. IV, we show its computational cost. Section V discusses the relationship between the EM algorithm and the q-EM algorithm. In Sec. VI, we show numerical simulations of the δ -EM algorithm to confirm that our starting point is valid. In Sec. VII, we address the relationship between the EM algorithm and the k -means algorithm. Finally, Sec. VIII concludes this paper.

Independently of our work, Iordanis Kerenidis, Alessandro Luongo, and Anupam Prakash proposed an extension of the q-means algorithm to Gaussian mixture models similar to this work using soft clustering [17].

II. CLASSICAL PRELIMINARIES

In this section, we first review the EM algorithm [9,10] in detail. We then introduce a randomized variant of the EM algorithm, which we call the δ -EM algorithm. The purpose of introducing the δ -EM algorithm is the need of a robust variant of the EM algorithm against noise which resembles the nondeterministic quantum measurement of a superposed state.

A. EM algorithm

The EM algorithm is a generic approach to estimate parameters of probability distributions based on maximum

*hideyuki_miyahara@mist.i.u-tokyo.ac.jp, hmiyahara512@gmail.com

Algorithm 1 EM algorithm.

```

1:  $t = 0$ 
2: assign  $y_i$  for  $i = 1, 2, \dots, N$  to clusters  $k = 1, 2, \dots, K$  randomly
3: while convergence criterion is not satisfied do
4: (E step) compute the responsibilities of cluster  $k$  on  $r_t^{i,k}$ , Eq. (2)
5: (M step) estimate  $\theta_t = \{\pi_t^k, \mu_t^k, \Sigma_t^k\}_{k=1}^K$  by Eqs. (3), (4), and (5)
6:  $t \leftarrow t + 1$ 
7: end while

```

likelihood estimation. For simplicity, we focus on the GMM and review the EM algorithm for the GMM. Let us consider a d -dimensional feature space and assume that we have N data points $\{y_i\}_{i=1}^N$. The GMM for $x \in \mathbb{R}^d$ is given by

$$p(y; \theta) = \sum_{k=1}^K \pi^k \mathcal{N}(y; \mu^k, \Sigma^k), \quad (1)$$

where $\mathcal{N}(y; \mu^k, \Sigma^k) := \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma^k|^{\frac{1}{2}}} e^{-\frac{1}{2}(y-\mu^k)(\Sigma^k)^{-1}(y-\mu^k)}$ is the d -dimensional Gaussian function with mean μ^k and covariance Σ^k and $\sum_k \pi^k = 1$. To simplify the notation, we define $\theta := \{\pi^k, \mu^k, \Sigma^k\}_{k=1}^K$.

The EM algorithm, which estimates θ , consists of the following two steps which are iterated until convergence. The first step, which is called the E step, is to compute the *responsibilities* of cluster k for each datapoint y_i :

$$r_t^{i,k} := \frac{\pi_t^k \mathcal{N}(y_i; \mu_t^k, \Sigma_t^k)}{\sum_{k'} \pi_t^{k'} \mathcal{N}(y_i; \mu_t^{k'}, \Sigma_t^{k'})}. \quad (2)$$

The second step, which called the M step, is to compute θ_t by using the responsibilities, Eq. (2):

$$\pi_{t+1}^k = \sum_i r_t^{i,k}, \quad (3)$$

$$\mu_{t+1}^k = \sum_i \frac{r_t^{i,k} y_i}{r_t^{i,k}}, \quad (4)$$

$$\Sigma_{t+1}^k = \sum_i \frac{r_t^{i,k} (y_i - \mu_t^k)(y_i - \mu_t^k)^\top}{r_t^{i,k}}. \quad (5)$$

We iterate the E and M steps by substituting Eqs. (3), (4), and (5) until convergence. Note that we can begin either of the E step or the M steps for the first iteration. The EM algorithm is summarized in Algorithm 1. Note that the procedure of the EM algorithm can be generalized for mixture models [9,10].

As explained above, the EM algorithm has two steps. On the other hand, the computational procedure of the q-EM algorithm is divided into four steps. Roughly speaking, the first two steps of the q-EM algorithm correspond to the E step of the EM algorithm and the second two steps of the q-EM algorithm correspond to the M step of the EM algorithm. This point will be mentioned in Sec. V again.

B. δ -EM algorithm

As a prerequisite to the q-EM algorithm, we need to modify the original EM algorithm, since we have to take into account randomness associated with quantum measurement.

In Sec. II A, we explained that the EM algorithm has two steps: the E and M steps. In the δ -EM algorithm, we modify the E step in the spirit of the δ - k -means algorithm in Ref. [11].

To this end, we first introduce the square GMM distance by

$$d_G^k(y_i) := (y_i - \mu^k)^\top \Sigma^k (y_i - \mu^k) + \ln |\Sigma^k| - 2 \ln(K\pi^k). \quad (6)$$

Note that when $\pi^k = 1/K$ and Σ^k is the identity matrix for $k = 1, 2, \dots, K$, $d_G^k(y_i) = d_E^k(y_i)$, where $d_E^k(\cdot)$ is the square Euclidean distance given by

$$d_E^k(y_i) := (y_i - \mu^k)^\top (y_i - \mu^k). \quad (7)$$

We then define the set of labels given by

$$L_G^\delta(y_i) := \{k \mid \|d_G^*(y_i) - d_G^k(y_i)\| \leq \delta\}, \quad (8)$$

where $d_G^*(y_i) := \min_k d_G^k(y_i)$. In the E step of the δ -EM algorithm, we take random samples from $L_G^\delta(y_i)$ in Eq. (8). We note that for soft clustering, a more precise sampling scheme may be useful, but this simple sampling works well in the approach shown in Ref. [11].

In the M step of the δ -EM algorithm, we add small noise to the estimated parameters after their estimation. As a result, the δ -EM algorithm becomes robust, and its quantum version will become implementable. In Sec. VI, we will show the validity of this algorithm numerically. Remarkably, we find that adding noise can even improve the quality of the studied benchmark examples.

III. QUANTUM ALGORITHM FOR THE EM ALGORITHM WITH THE GMM

In this section, we describe the procedure of the quantum algorithm that realizes a quantum speedup of the EM algorithm for the GMM.

To simplify the notation, we add the tilde for estimates throughout this paper; that is, we denote, e.g., \tilde{a} as the estimate of a .

A. Overview of the q-EM algorithm

We begin with the initialization of the q-EM algorithm. In the EM algorithm, we can begin either with the E step or the M step for the first iteration. For simplicity, in the case of the q-EM algorithm, we consider getting started with the E step; then we set the initial parameter set $\theta_0 = \{\pi_0, \mu_0, \Sigma_0\}$ with $\pi_0 := [\pi_0^1, \pi_0^2, \dots, \pi_0^K]$, $\mu_0 := [\mu_0^1, \mu_0^2, \dots, \mu_0^K]$, and $\Sigma_0 := [\Sigma_0^1, \Sigma_0^2, \dots, \Sigma_0^K]$.

The main procedure of the q-EM algorithm is composed of four steps. In step I, we compute the square GMM distance, and in step II, it is minimized for cluster assignment. Then, in step III, we generate quantum states of weight vectors, mean vectors, and covariance matrices, and in step IV, we apply quantum vector state tomography. By using the classical information on mean vectors and covariance matrices, we repeat the whole procedures until convergence. The output of this algorithm is $\theta_* = \{\pi_*, \mu_*, \Sigma_*\}$. The q-EM algorithm is summarized in Algorithm 2. In the rest of this section, we will explain the four steps in detail.

Algorithm 2 q-EM algorithm.

```

1:  $t = 0$ 
2: prepare for the data structures
3: while convergence criterion is not satisfied do
4:   compute the square GMM distance (step I)
5:   assign clusters (step II)
6:   generate the mean and covariance states (step III)
7:   update the parameters (step IV)
8:    $t \leftarrow t + 1$ 
9: end while
    
```

B. Step I: Computing the square GMM distance

In this step, we compute the square GMM distance, Eq. (6). Mathematically, we apply the unitary operation:

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle (\otimes_{k \in [K]} |k\rangle |0\rangle) \mapsto \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle [\otimes_{k \in [K]} |k\rangle |\tilde{d}_G^k(y_i)\rangle], \quad (9)$$

where $d_G^k(y_i)$ is the square GMM distance between y_i and the k th cluster, and $[K] := \{k\}_{k=1}^K$. For this computation, we require the precision given by $\|\tilde{d}_G^k(y_i) - d_G^k(y_i)\| \leq \epsilon_1$. In the next section, ϵ_1 will be used to analyze the runtime.

Equation (9) includes summation, multiplication, and inner products of quantum states. Among them, the computation of summation and multiplication is straightforward with quantum linear algebra while the implementation of inner products is more involved. Let us thus focus on the computation of inner products.

We assume that two unitary operations and their controlled versions are available as follows:

$$|i\rangle|0\rangle \mapsto |i\rangle\hat{G}^k|y_i\rangle, \quad (10)$$

$$|k\rangle|0\rangle \mapsto |k\rangle\hat{G}^k|\mu^k\rangle, \quad (11)$$

where $\hat{G}^k := (\hat{\Sigma}^k)^{1/2}$ for $k = 1, 2, \dots, K$. We then begin with the state

$$|\phi_{i,k}\rangle := |i\rangle|k\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle. \quad (12)$$

By using controlled versions of Eq. (10) and (11), we create the following state from $|\phi_{i,k}\rangle$:

$$|\phi_{i,k}^C\rangle := \frac{1}{\sqrt{2}}(|i\rangle|k\rangle|0\rangle\hat{G}^k|y_i\rangle + |i\rangle|k\rangle|1\rangle\hat{G}^k|\mu^k\rangle). \quad (13)$$

Then we apply the Hadamard gate on the third register of $|\phi_{i,k}^C\rangle$ and the resulting state is

$$\begin{aligned} |\phi_{i,k}^H\rangle &= \frac{1}{2}|i\rangle|k\rangle[|0\rangle(\hat{G}^k|y_i\rangle + \hat{G}^k|\mu^k\rangle) \\ &\quad + |1\rangle(\hat{G}^k|y_i\rangle - \hat{G}^k|\mu^k\rangle)]. \end{aligned} \quad (14)$$

Note that Eq. (14) is also represented as

$$|\phi_{i,k}^H\rangle = |i\rangle|k\rangle(\sqrt{p_{i,k}}|\text{tar}_{i,k}, 1\rangle + \sqrt{1-p_{i,k}}|\text{gar}_{i,k}, 0\rangle), \quad (15)$$

where $|\text{tar}_{i,k}, 1\rangle := |1\rangle\hat{G}^k(|y_i\rangle - |\mu^k\rangle)$ and $|\text{gar}_{i,k}, 0\rangle$ is a garbage state. That is, we have a unitary operator such that

$$\begin{aligned} \hat{U}_1 : |i\rangle|j\rangle|0\rangle \\ \mapsto |i\rangle|j\rangle(\sqrt{p_{i,k}}|\text{tar}_{i,k}, 1\rangle + \sqrt{1-p_{i,k}}|\text{gar}_{i,k}, 0\rangle). \end{aligned} \quad (16)$$

We also note that the probability that we get $|1\rangle$ by measuring the third register is expressed as

$$p_{i,k} = \frac{1 - \langle y_i | (\hat{G}^k)^2 | \mu^k \rangle}{2}. \quad (17)$$

We then apply all the operations except the measurement in amplitude estimation in Ref. [11,18] on \hat{U}_1 in Eq. (16). This process realizes the following unitary operation:

$$\begin{aligned} \hat{U}_2 : |i\rangle|j\rangle|0\rangle \\ \mapsto |i\rangle|k\rangle(\sqrt{\alpha}|\tilde{p}_{i,k}, 1\rangle + \sqrt{1-\alpha}|\text{gar}_{i,k}, 0\rangle), \end{aligned} \quad (18)$$

where $\|\tilde{p}_{i,k} - p_{i,k}\| < 2\pi\sqrt{p_{i,k}(1-p_{i,k})}/P_{\text{ae}} + \pi^2/P_{\text{ae}}^2$ and $\alpha > \frac{8}{\pi^2}$ [18]. Here M is a parameter to be determined (see Sec. B 1). Next, applying the mode evaluation method [19] in Lemma 8 of Ref. [7] and Theorem 2.2 of [11] to Eq. (18), we get $|\Phi_{i,k}\rangle$ such that

$$\| |\Phi_{i,k}\rangle - |0\rangle^{\otimes L} |\tilde{p}_{i,k}\rangle \|_2 \leq \sqrt{2\Delta}. \quad (19)$$

The last step is to estimate the square GMM distance of unnormalized vectors $\|y_i\|$ and $\|\mu^k\|$ and to multiply the norms of them and adding $\ln \pi^k$. A translation operator $\hat{T}(r')$ can conduct the adding operation: $\hat{T}(r')|r\rangle = |r+r'\rangle$ for $r, r' \in \mathbb{R}^N$. Note that we have assumed that we know the norms of $\{\|y_i\|\}_{i=1}^N$ and the same assumption is used in Ref. [11].

C. Step II: Assignment of clusters

The purpose of step II is cluster assignment. In this step, we utilize the following unitary operation:

$$\hat{U}_3 : (\otimes_{k \in [K]} |a^k\rangle)|0\rangle \mapsto (\otimes_{k \in [K]} |a^k\rangle) | \arg \min_{k \in [K]} a^k \rangle, \quad (20)$$

where $|a^k\rangle$ is a $(\ln p)$ -bit state for $k = 1, 2, \dots, K$. The computational cost of this operation is $O(K \ln p)$ [11].

To find cluster assignment, we perform the unitary operation given by

$$\begin{aligned} \hat{U}_4 : \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle [\otimes_{k \in [K]} |k\rangle |d_G^k(y_i)\rangle] \\ \mapsto \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle |\text{label}_t(y_i)\rangle, \end{aligned} \quad (21)$$

where $\text{label}_t(y_i)$ is the optimal label of y_i at time t .

Finally, by uncomputing the square GMM distances, we obtain

$$|\psi_t\rangle := \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle |\text{label}_t(y_i)\rangle. \quad (22)$$

This uncomputation is required to repeat iterations.

D. Step III: Generation of the mean and covariance states

In this step, we generate states that store information on the weights, mean vectors, and covariance matrices. Let us recall that Eq. (22) is also expressed as

$$|\psi_t\rangle = \sum_{k=1}^K \sqrt{\frac{N_t^k}{N}} \left(\frac{1}{\sqrt{N_t^k}} \sum_{i \in C_t^k} |i\rangle \right) |k\rangle, \quad (23)$$

$$= \sum_{k=1}^K \sqrt{\frac{N_t^k}{N}} |\chi_t^k\rangle |k\rangle. \quad (24)$$

Thus, by measuring the label register of $|\psi_t\rangle$ in Eq. (22), we obtain, with probability N_t^k/N ,

$$|\chi_t^k\rangle = \frac{1}{\sqrt{N_t^k}} \sum_{i \in C_t^k} |i\rangle, \quad (25)$$

where C_t^k is the set of labels that belong to cluster k at time t . Then, $\chi_t^k = [\dots, (\chi_t^k)_{i-1}, (\chi_t^k)_i, (\chi_t^k)_{i+1}, \dots]^T \in \mathbb{R}^N$ is

$$(\chi_t^k)_i = \begin{cases} 1/N_t^k & (i \in C_t^k) \\ 0 & (i \notin C_t^k) \end{cases}, \quad (26)$$

for $i = 1, 2, \dots, N$.

We here define $V_1 \in \mathbb{R}^{N \times d}$, $V_2 \in \mathbb{R}^{N \times d \times d}$, and $V_{0,i}$ for $i = 1, 2, \dots, N$ on a quantum random access memory (QRAM):

$$V_1 := [y_1, y_2, \dots, y_N], \quad (27)$$

$$V_2 := [y_1 \otimes y_1, y_2 \otimes y_2, \dots, y_N \otimes y_N], \quad (28)$$

$$V_{0,i} := [\underbrace{\vec{0}, \vec{0}, \dots, \vec{0}}_{i-1}, \vec{1}, \underbrace{\vec{0}, \vec{0}, \dots, \vec{0}}_{N-i-1}]. \quad (29)$$

To obtain mean vectors, we multiply V_1 to the state $|\chi_t^k\rangle$ in Eq. (25) by using quantum linear algebra [20,21]:

$$|\mu_{t+1}^k\rangle = V_1 |\chi_t^k\rangle. \quad (30)$$

The associated error is ϵ_2^μ . Similarly, we compute a state involving information on Σ^k by using quantum linear algebra [20,21]:

$$|\text{vec}[\Sigma_{t+1}^k] + \mu_{t+1}^k \otimes \mu_{t+1}^k\rangle = V_2 |\chi_t^k\rangle. \quad (31)$$

The associated error is ϵ_2^Σ . Note that ϵ_2^μ and ϵ_2^Σ appear only in logarithms; thus, we do not explicitly consider them.

We finally deal with $\{\pi^k\}_{k=1}^K$; it is relatively easy to compute the weights of the GMM, $\{\pi_t^k\}_{k=1}^K$. We here utilize Eq. (29) as follows:

$$(V_{0,i} |\chi_t^k\rangle) = \begin{cases} (N_t^k)^{-1} |\vec{1}\rangle & (i \in C_t^k) \\ |\vec{0}\rangle & (\text{otherwise}) \end{cases}. \quad (32)$$

Thus, we can estimate N_t^k similarly. Note that we assume that the sizes of all clusters are $\Omega(N/k)$.

E. Step IV: Update of the parameters

At the end of each iteration, we obtain classical information on π_{t+1} , μ_{t+1} , and Σ_{t+1} by performing the quantum state tomography algorithm for $|\chi_t^k\rangle$, $|\mu_{t+1}^k\rangle$, and $|\text{vec}[\Sigma_{t+1}^k]\rangle$. Quantum vector state tomography is explained in Ref. [11].

The quantum state tomography algorithm in Ref. [11] requires a unitary transformation $U : |0\rangle \mapsto |x\rangle$; however, the procedure to find $l(y_i)$ is not deterministic. Then, we have to devise some deterministic methods to find $l(y_i)$. One solution is to determine $l(y_i)$ by the rule

$$l(y_i) = k, \quad (33)$$

if $d_G^k(y_i) < d_G^{k'}(y_i) - 2\delta$ for $k' \neq k$, and we discard the points to which no label can be assigned.

By introducing ϵ_4^π , ϵ_3^μ , ϵ_4^μ , ϵ_3^Σ , and ϵ_4^Σ , we require the following precision in this step: $\|\pi^k - \tilde{\pi}^k\| \leq \epsilon_4^\pi$, $\|\mu^k - \tilde{\mu}^k\| \leq \epsilon_3^\mu$, $\|\mu^k\| - \|\tilde{\mu}^k\| \leq \epsilon_4^\mu \|\mu^k\|$, $\|\Sigma^k - \tilde{\Sigma}^k\| \leq \epsilon_3^\Sigma$, and $\|\Sigma^k\| - \|\tilde{\Sigma}^k\| \leq \epsilon_4^\Sigma \|\Sigma^k\|$. In the next section, ϵ_4^π , ϵ_3^μ , ϵ_4^μ , ϵ_3^Σ , and ϵ_4^Σ will be used to analyze the runtime.

IV. ANALYSIS OF ERRORS AND RUNTIME

This section is dedicated to error and runtime analysis of the q-EM algorithm. We first state the main claim and then explain it.

A. Main result

The runtime of the q-EM algorithm is represented by

$$\tilde{O} \left\{ \frac{K^2}{\epsilon_1 (\epsilon_4^\pi)^2} + K d \frac{\kappa(V_1)}{(\epsilon_4^\mu)^2} \left[\mu(V_1) + K \frac{\eta^\mu}{\epsilon_1} \right] + \frac{K^2 \eta^\mu \kappa(V_1) \mu(V_1)}{\epsilon_1 \epsilon_3^\mu} + K d^2 \frac{\kappa(V_2)}{(\epsilon_4^\Sigma)^2} \times \left[\mu(V_2) + K \frac{\eta^\Sigma}{\epsilon_1} \right] + \frac{K^2 \eta^\Sigma \kappa(V_2) \mu(V_2)}{\epsilon_1 \epsilon_3^\Sigma} \right\}, \quad (34)$$

where $\mu(\cdot)$ is given in Eq. (B8), $\kappa(\cdot)$ is the condition number, $\eta^\mu := \max_i \|y_i\|^2$, and $\eta^\Sigma := \max_i \|y_i \otimes y_i\|^2$. The definition of $\tilde{O}(\cdot)$ is given in Appendix A. Equation (34) implies that there exists a constant number k such the numerical cost of the q-EM algorithm is given by

$$O[\ln^k(N)], \quad (35)$$

where $O(\cdot)$ is the Landau's big O notation.

This result states that the runtime of each iteration of the q-EM algorithm is exponentially faster than that of the EM algorithm.

B. Error analysis

We first summarize the errors in the q-EM algorithm to analyze the total runtime of the q-EM algorithm in the following subsection. In step I, we compute $d_G^k(y_i)$; the error on this computation is

$$\|\tilde{d}_G^k(y_i) - d_G^k(y_i)\| < \epsilon_1. \quad (36)$$

For consistency between the EM algorithm and the δ -EM algorithm, we take $\epsilon_1 < \delta/2$.

In steps III and IV, we compute μ and Σ . The errors on $\|\mu^k\|$ and $\|\mu^k\|$ are $\sqrt{\eta^\mu} \epsilon_3^\mu$ and ϵ_4^μ , respectively. Then, the error on the estimation of μ takes the form

$$\|\tilde{\mu}^k - \mu^k\| \leq \sqrt{\eta^\mu} (\epsilon_3^\mu + \epsilon_4^\mu). \quad (37)$$

See also Appendix C for the above calculation. Next, we turn our attention to Σ . The errors on $\sqrt{\eta^\Sigma \epsilon_3^\Sigma}$ and $\sqrt{\eta^\Sigma \epsilon_4^\Sigma}$ are $\sqrt{\eta^\Sigma \epsilon_3^\Sigma}$ and ϵ_4^Σ , respectively. Similarly to the case of μ^k , the error on the estimation of Σ is shown as

$$\|\text{vec}[\tilde{\Sigma}^k] - \text{vec}[\Sigma^k]\| \leq \sqrt{\eta^\Sigma}(\epsilon_3^\Sigma + \epsilon_4^\Sigma). \quad (38)$$

Finally, we mention the error associated with the estimation on π^k for $k = 1, 2, \dots, K$. The error on π^k is shown as

$$\|\tilde{\pi}^k - \pi^k\| \leq \epsilon_4^\pi. \quad (39)$$

We estimate $\{\pi^k\}_{k=1}^K$ via the distribution of labels in quantum vector state tomography.

C. Runtime

In the following, Eq. (34), i.e., the runtime of each iteration of the q-EM algorithm is derived using Hoeffding's inequality. The required number of quantum vector state tomography of K mean vectors is given as follows [11]:

$$\tilde{O}\left[\frac{Kd \ln K \ln d}{(\epsilon_4^\mu)^2}\right]. \quad (40)$$

Similarly, that for covariance matrices is

$$\tilde{O}\left[\frac{Kd^2 \ln K \ln d^2}{(\epsilon_4^\Sigma)^2}\right]. \quad (41)$$

Next, we turn our attention to the runtime to prepare single copies of $|\mu^k\rangle$ and $|\text{vec}[\Sigma^k]\rangle$. The time to prepare a copy of $|\mu_i^k\rangle$ is

$$O\{\kappa(V_1)[\mu(V_1) + T_\chi^\mu] \ln(1/\epsilon_2)\}, \quad (42)$$

and that to prepare a copy of $|\text{vec}[\Sigma_i^k]\rangle$ is

$$O\{\kappa(V_2)[\mu(V_2) + T_\chi^\Sigma] \ln(1/\epsilon_2)\}, \quad (43)$$

where $\mu(V_1)$ is given in Eq. (B8), $\kappa(V_1)$ is the condition number of V_1 , $\eta^\mu := \max_i \|y_i\|^2$ and $\eta^\Sigma := \max_i \|y_i \otimes y_i\|^2$. Furthermore, T_χ^μ , which is the time to prepare $|\chi_i^k\rangle$ for estimating $\{\mu^k\}_{k=1}^K$, is given by

$$T_\chi^\mu = \tilde{O}\left[\frac{K\eta^\mu \ln(\Delta^{-1}) \ln(Nd)}{\epsilon_1}\right], \quad (44)$$

$$= \tilde{O}\left(\frac{K\eta^\mu}{\epsilon_1}\right). \quad (45)$$

Similarly, T_χ^Σ is given by

$$T_\chi^\Sigma = \tilde{O}\left[\frac{K\eta^\Sigma \ln(\Delta^{-1}) \ln(Nd)}{\epsilon_1}\right], \quad (46)$$

$$= \tilde{O}\left(\frac{K\eta^\Sigma}{\epsilon_1}\right). \quad (47)$$

In addition, T_χ^π is given by

$$T_\chi^\pi = \tilde{O}\left[\frac{K \ln(\Delta^{-1}) \ln(Nd)}{\epsilon_1}\right], \quad (48)$$

$$= \tilde{O}\left(\frac{K}{\epsilon_1}\right). \quad (49)$$

We also need to estimate the norms of $|\mu^k\rangle$ and $|\text{vec}[\Sigma^k]\rangle$. The time for the norm estimation of $|\mu^k\rangle$ is

$$\tilde{O}\left[\frac{KT_\chi^\mu \kappa(V_1) \mu(V_1)}{\epsilon_3^\mu}\right], \quad (50)$$

and that of $|\text{vec}[\Sigma^k]\rangle$ is

$$\tilde{O}\left[\frac{KT_\chi^\Sigma \kappa(V_2) \mu(V_2)}{\epsilon_3^\Sigma}\right]. \quad (51)$$

We then estimate the runtime for estimating $\{\pi^k\}_{k=1}^K$. Due to Hoeffding's inequality [22], we need to perform sampling $\frac{2K}{(\epsilon_4^\pi)^2} \ln \frac{2}{\Delta_{\text{each}}^\pi}$ times to realize $\|\tilde{p}_k - p_k\| \leq \epsilon_4^\pi$ for $k = 1, 2, \dots, K$ with probability $(1 - \Delta_{\text{each}}^\pi)^K$ since the distribution of π is the K -state discrete distribution. By setting $1 - \Delta^\pi := (1 - \Delta_{\text{each}}^\pi)^K$, we have $\Delta_{\text{each}}^\pi = 1 - (1 - \Delta^\pi)^{1/K}$. Thus, we have

$$N_\pi = \frac{2K}{(\epsilon_4^\pi)^2} \ln \frac{2}{1 - (1 - \Delta^\pi)^{1/K}}, \quad (52)$$

$$= \tilde{O}\left[\frac{K}{(\epsilon_4^\pi)^2}\right]. \quad (53)$$

Furthermore, we have to repeat the estimation process K times for estimation of π compared to those of μ and Σ , since we have to sample i from C^k , K times, for $k = 1, 2, \dots, K$. Thus, the runtime for estimating π has an additional multiple of K .

Thus, the total runtimes for estimating $\{\pi^k\}$, $\{\mu^k\}$, and $\{\Sigma^k\}$ are, respectively,

$$\tilde{O}\left[\frac{K^3}{\epsilon_1(\epsilon_4^\pi)^2}\right], \quad (54)$$

$$\tilde{O}\left\{Kd \frac{\kappa(V_1)}{(\epsilon_4^\mu)^2} \left[\mu(V_1) + K \frac{\eta^\mu}{\epsilon_1}\right] + \frac{K^2}{\epsilon_1} \frac{\eta^\mu \kappa(V_1) \mu(V_1)}{\epsilon_3^\mu}\right\}, \quad (55)$$

$$\tilde{O}\left\{Kd^2 \frac{\kappa(V_2)}{(\epsilon_4^\Sigma)^2} \left[\mu(V_2) + K \frac{\eta^\Sigma}{\epsilon_1}\right] + \frac{K^2}{\epsilon_1} \frac{\eta^\Sigma \kappa(V_2) \mu(V_2)}{\epsilon_3^\Sigma}\right\}. \quad (56)$$

In total, we have obtained Eq. (34).

V. CORRESPONDENCE BETWEEN THE EM ALGORITHM AND THE Q-EM ALGORITHM

In this section, we discuss the correspondence between the EM algorithm and the q-EM algorithm. As explained in Sec. II A, the EM algorithm has two steps called the E and M steps. In the E step, we compute the responsibilities. Roughly speaking, the first and second steps of the q-EM algorithm correspond to this step since cluster assignment is computed in the first and second step in the case of the q-EM algorithm. On the other hand, the M step of the EM algorithm computes parameters. Similarly, the third and fourth steps of the q-EM algorithm correspond to this step since parameters are updated in the third and fourth steps of the q-EM algorithm.

We also add some remarks on the similar and different points between the EM algorithm and the q-EM algorithm. The similar point is that, at the end of each iteration, both algorithms have updated parameters. The different point is the

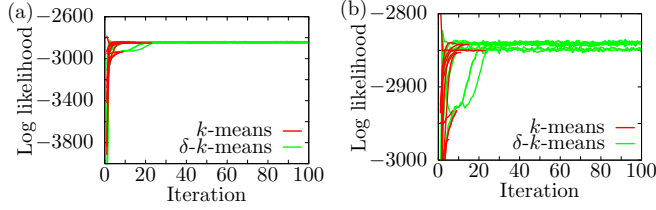


FIG. 1. Log-likelihood of the k -means algorithm (red lines) and the δ - k -means algorithm (green lines). We perform the simulation 10 times, respectively. The difference of (a) and (b) is the scale of the vertical axis.

data structure on cluster assignment. In the EM algorithm, the responsibilities are computed; on the other hand, the superposed state is stored in the q-EM algorithm. Roughly speaking, this leads to the advantage of the q-EM algorithm over the EM algorithm.

VI. NUMERICAL SIMULATION

To devise a quantum version of the EM algorithm, we proposed the δ -EM algorithm in Sec. II. In this section, to see that the EM and δ -EM algorithms are equivalent when δ is sufficiently small and that the δ -EM algorithm improves upon the δ - k -means algorithm, we show numerical simulations of the EM algorithm, the δ -EM algorithm, the k -means algorithm, and the δ - k -means algorithm.

In Ref. [23], the comparison between the k -means algorithm and the EM algorithm with GMM is shown. Then we use similar synthetic data sets used in Ref. [23]. In the numerical simulations, we set $\delta = 0.2$ except the numerical simulation for the δ dependence of the δ -EM algorithm. For simplicity, we add Gaussian noise to the parameters estimated in the M step of the δ -EM algorithm and the centroids estimated in the δ - k -means algorithm [24].

A. Example I

We begin with the explanation of the data set used in this subsection. We generated by drawing 1000 data points from the mixture of two Gaussian functions. The means of the two Gaussian functions are $\mu^1 = [0.3, 0.0]^T$ and $\mu^2 = [-0.3, 0.0]^T$, respectively, and the covariances are, respectively,

$$\Sigma^1 = \begin{bmatrix} 1.0 & 0.98 \\ 0.98 & 1.0 \end{bmatrix}, \quad (57)$$

$$\Sigma^2 = \begin{bmatrix} 1.0 & -0.98 \\ -0.98 & 1.0 \end{bmatrix}. \quad (58)$$

We also put $\pi^1 = \pi^2 = 0.5$.

In Figs. 1 and 2, we show the log-likelihood of the k -means algorithm and the δ - k -means algorithm and that of the EM algorithm and the δ -EM algorithm, respectively. We plot 10 trials for each algorithm. These figures show that the k -means and δ - k -means algorithms have similar performance and that the EM and δ -EM algorithms have similar performance. For clarity, we graphically show the parameters estimated by the δ - k -means algorithm and the δ -EM algorithm in Fig. 3 [25]. We have chosen the best estimates of the δ - k -means algorithm

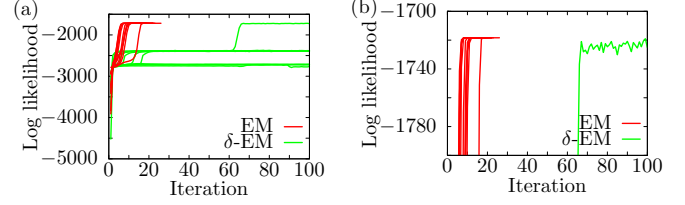


FIG. 2. Log-likelihood of the EM algorithm (red lines) and the δ -EM algorithm (green lines). We perform the simulation 10 times, respectively. The difference of (a) and (b) is the scale of the vertical axis.

and the δ -EM algorithm in one hundred trials. These figures demonstrate that the δ -EM algorithm outperforms the δ - k -means algorithm.

In Table I, we summarize the success rates of the EM algorithm, the δ -EM algorithm, the k -means algorithm, and the δ - k -means algorithm. Here, the success rate means the ratio of the number of the successfully predicted hidden variables [26] to the number of the total data points. This table shows that the δ -EM algorithm works better than the δ - k -means algorithm. Thus, we insist that it is meaningful to devise a quantum version of the δ -EM algorithm.

In Fig. 4, we show the δ dependence of the best success rates of the δ -EM algorithm in 100 trials. This figure shows that the δ -EM algorithm is robust for small δ , but the performance decreases rapidly for large values of δ . We need to set δ small, since the critical value depends on data sets.

B. Example II

We again start with the data set used in this subsection. The data points are also generated by the mixture of two Gaussian functions, but the parameters are different. We set $[\pi^1, \pi^2] = [0.7, 0.3]$, $\mu^1 = [0.0, -0.5]^T$, $\mu^2 = [0.0, 0.0]^T$, and

$$\Sigma^1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}, \quad (59)$$

$$\Sigma^2 = \begin{bmatrix} 10.0 & 0.0 \\ 0.0 & 0.10 \end{bmatrix}. \quad (60)$$

Furthermore, we draw 1000 data points from the mixture of two Gaussian functions.

We first show the parameters estimated by the δ - k -means algorithm and the δ -EM algorithm in Fig. 5. We have chosen the best estimates of the δ - k -means algorithm and the δ -EM algorithm in 100 trials. These figures represent that the δ -EM algorithm outperforms the δ - k -means algorithm. In particular,

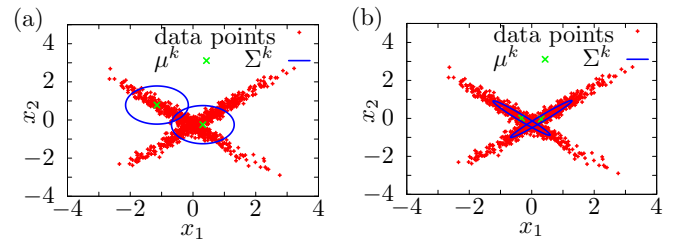


FIG. 3. Pictures of estimated functions by (a) the δ - k -means algorithm and (b) the δ -EM algorithm.

TABLE I. Success rates of the EM algorithm, the δ -EM algorithm, the k -means algorithm, and the δ - k -means algorithm. These scores are best ones in 100 trials with randomized initial inputs.

EM	δ -EM	k -means	δ - k -means
93.9%	94.3%	72.4%	72.5%

in the case of the δ - k -means algorithm, the covariances are fixed at the identity matrix; then each cluster tries to exclude each other.

In Table II, we summarize the success rates of the EM algorithm, the δ -EM algorithm, the k -means algorithm, and the δ - k -means algorithm. Here the success rate means the ratio of the number of the successfully predicted labels to the number of the total data points. This table shows that the δ -EM algorithm works better than the δ - k -means algorithm.

VII. DISCUSSIONS

We here discuss the relationship between the EM algorithm with the GMM and the k -means algorithm. The EM algorithm with the GMM is an extension of the k -means algorithm; thus we explain the two conditions that the EM algorithm with the GMM becomes identical to the k -means algorithm.

The first condition is that $r_i^{i,k}$ takes 1 for a certain k and 0 otherwise. This implies that the k -means algorithm is an algorithm for hard clustering, while the EM algorithm is one for soft clustering. The second condition is that $\pi^k = 1/K$ and $\Sigma^k = I_d$, where I_d is the d -dimensional identity matrix for $k = 1, 2, \dots, K$. This is the reason why the k -means algorithm does not explicitly deal with weights and covariance matrices. From the viewpoint of a probability distribution, the k -means algorithm is an algorithm to estimate $\{\mu_i^k\}_{k=1}^K$ in

$$p(x; \{\mu_i^k\}_{k=1}^K) = \frac{1}{K} \frac{1}{(2\pi)^{\frac{d}{2}}} \sum_{k=1}^K e^{-\frac{1}{2} \|x - \mu_i^k\|^2}. \quad (61)$$

As shown in Sec. VI, the weights and the covariances of the GMM play an important role; thus, we also insist that the δ -EM algorithm is a meaningful extension of the δ - k -means algorithm.

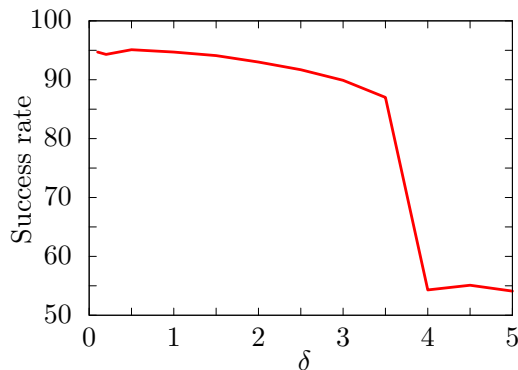


FIG. 4. δ Dependence of success rates. Each success rate is the best one in 100 trials.

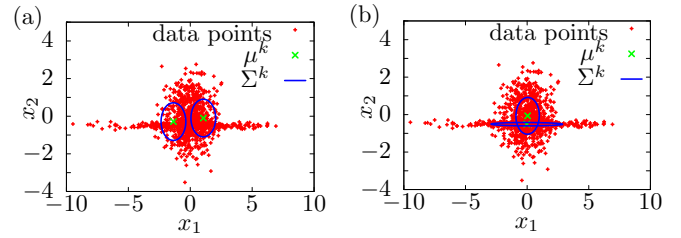


FIG. 5. Pictures of estimated functions by (a) the δ - k -means algorithm and (b) the δ -EM algorithm.

VIII. CONCLUSION

In this paper, we have proposed a quantum algorithm for the EM algorithm and showed that it realize a quantum speedup compared to the classical EM algorithm. The key idea is to generalize the distance that is minimized in the k -means algorithm by considering also weights and covariances. Though we have focused on the GMM, we can generalize this condition to other mixture models. In machine learning, the EM algorithm with the GMM is more often used than the k -means algorithm; thus, this work is an important step toward quantum machine learning. The algorithm requires a QRAM oracle which has so far not been implemented in experiments yet. As a future direction we will investigate the applicability of superposition designs as proposed in Refs. [27,28].

ACKNOWLEDGMENTS

H.M. thanks Clemens Daska for fruitful discussions. H.M. is supported by JSPS KAKENHI Grant No. JP18J12175. The work is supported by the Austrian Science Fund (FWF) through a START grant under Project No. Y1067-N27 and the SFB BeyondC Project No. F7108-N38, the Hauser-Raspe foundation, and the European Union's Horizon 2020 research and innovation program under Grant No. 817482 PasQuanS.

APPENDIX A: BIG O NOTATION

We here introduce the big O notation, which is often used in computer science. For functions $f(x)$ and $g(x)$, one writes

$$f(x) = O[g(x)], \quad (A1)$$

if and only if

$$\exists x_0, \exists M > 0, \text{ s.t. } x > x_0 \Rightarrow \|f(x)\| < M\|g(x)\|. \quad (A2)$$

Similarly, we say

$$f(x) = \tilde{O}[g(x)] \quad (A3)$$

TABLE II. Success rates of the EM algorithm, the δ -EM algorithm, the k -means algorithm, and the δ - k -means algorithm. These scores are best ones in 100 trials with randomized initial inputs.

EM	δ -EM	k -means	δ - k -means
88.8%	89.2%	57.9%	55.4%

if and only if

$$\exists k, f(x) = O[g(x) \ln^k(x)]. \quad (\text{A4})$$

These definitions will be utilized to describe the q-EM algorithm and to perform error analysis.

APPENDIX B: QUANTUM PRELIMINARIES

We provide some tools that are required for the q -means algorithm [11] in this section. These tools are also utilized in a quantum algorithm for the EM algorithm.

1. Amplitude estimation

Here we summarize the amplitude estimation algorithm that was proposed in Ref. [18]. Assume that we have U_A given by

$$U_A : |0\rangle \mapsto \sqrt{p}|tar_{i,k}, 1\rangle + \sqrt{1-p}|gar_{i,k}, 0\rangle. \quad (\text{B1})$$

Then there exists an amplitude estimation algorithm that outputs \tilde{p} such that

$$\|\tilde{p} - p\| \leq 2\pi \frac{\sqrt{p(1-p)}}{P_{ae}} + \left(\frac{\pi}{P_{ae}}\right)^2, \quad (\text{B2})$$

with probability at least $8/\pi^2$. The algorithm perform U_A P_{ae} times. Note that if $p = 0$, then $\tilde{p} = 0$, and if $p = 1$ and P_{ae} is even, then $\tilde{p} = 1$.

Furthermore, to raise the probability to obtain a good estimate on distances, we utilize a tool in Ref. [7]. We make multiple copies of the amplitude estimates, apply the quantum mode evaluation algorithm proposed in Lemma 8 of Ref. [7] in Sec. B 2, and reverse the circuit to remove the garbage state. We note that very recently an amplitude estimation algorithm without phase estimation was introduced [29].

2. Median evaluation

The time complexity of the mode evaluation algorithm is given in Lemma 8 of Ref. [7]. Let us summarize the main idea of this Lemma. Let U be a unitary operation given by

$$U : |0^{\otimes n}\rangle \mapsto \sqrt{a}|x, 1\rangle + \sqrt{1-a}|gar, 0\rangle \quad (\text{B3})$$

for $1/2 < a \leq 1$ in time T . Then there exists a quantum algorithm that produces a state $|\Phi\rangle$ such that

$$\|\Phi_{i,k}\rangle - |0\rangle^{\otimes nL}|x\rangle\|_2 \leq \sqrt{2\Delta} \quad (\text{B4})$$

for $\Delta > 0$, $1/2 < a_0 < a$, and integer L in time $2T \lceil \frac{\ln(1/\Delta)}{2(a_0 - 1/2)^2} \rceil$.

3. Quantum random access memory

In the q -means and q -EM algorithms, it is crucial to prepare data as a quantum state efficiently. To this end, we exploit

the QRAM introduced in Refs. [30,31]. Here, we consider a device that performs the operation

$$\sum_j \psi_j |j\rangle_a \xrightarrow{\text{QRAM}} \sum_j \psi_j |j\rangle_a |D_j\rangle_d. \quad (\text{B5})$$

We follow the application of the QRAM as in Ref. [11]. Let $V_1 \in \mathbb{R}^{N \times d}$; then there is a data structure to store the rows of V_1 such that the time to insert, update, or delete a single entry $v_{i,j}$ is $O(\ln^2 N)$ and a quantum algorithm on the data structure can be performed in time $O(\ln^2 N)$ that realizes the following unitaries:

$$|i\rangle|0\rangle \mapsto |i\rangle|v_i\rangle \text{ for } i \in [N], \quad (\text{B6})$$

$$|0\rangle \mapsto \sum_{i \in [N]} \|v_i\| |i\rangle. \quad (\text{B7})$$

4. Quantum linear algebra

Some useful subroutines that are used in q -means and q -EM are given as follows:

Theorem 1. Let $M \in \mathbb{R}^{d \times d}$ that satisfies $\|M\|_2 = 1$ and $x \in \mathbb{R}^d$. If M is stored in QRAM and the time to prepare $|x\rangle$ is T_x , then there exist quantum algorithms that return

(i) a state $|z\rangle$ such that $\| |z\rangle - |Mx\rangle \| \leq \epsilon$ in time $\tilde{O}[\{\kappa(M)\mu(M) + T_x\kappa(M)\} \ln(\epsilon^{-1})]$,

(ii) a state $|z\rangle$ such that $\| |z\rangle - |M^{-1}x\rangle \| \leq \epsilon$ in time $\tilde{O}[\{\kappa(M)\mu(M) + T_x\kappa(M)\} \ln(\epsilon^{-1})]$,

(iii) the norm $z \in (1 + \delta)\|Mx\|$ with relative error δ in time $\tilde{O}[T_x\kappa(M)\mu(M)\delta^{-1} \ln(\epsilon^{-1})]$,

where

$$\mu(M) := \min_{p \in [0,1]} [\|M\|_F, \sqrt{s_{2p}(M)s_{1-2p}(M^T)}], \quad (\text{B8})$$

$$s_p(M) := \max_{i \in [n]} \sum_{j \in [d]} M_{i,j}^p, \quad (\text{B9})$$

and $\kappa(M)$ is the condition number of M .

APPENDIX C: INEQUALITY FOR ERROR ANALYSIS

The following holds for general \vec{a} and \vec{b} :

$$\begin{aligned} & \| \|\vec{a}\| \cdot |\vec{a}\rangle - \|\vec{b}\| \cdot |\vec{b}\rangle \| \\ & \leq \| \|\vec{a}\| \cdot |\vec{a}\rangle - \|\vec{a}\| \cdot |\vec{b}\rangle \| + \| \|\vec{a}\| \cdot |\vec{b}\rangle - \|\vec{b}\| \cdot |\vec{b}\rangle \| \quad (\text{C1}) \end{aligned}$$

$$= \| \|\vec{a}\| \cdot \|\vec{a}\rangle - \|\vec{b}\rangle \| + (\|\vec{a}\| - \|\vec{b}\|) \|\vec{b}\rangle. \quad (\text{C2})$$

We used the above equation to derive Eq. (37).

[1] P. W. Shor, *SIAM Rev.* **41**, 303 (1999).

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. (Cambridge University Press, Cambridge, UK, 2010).

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Nature* **549**, 195 (2017).

[4] S. Lloyd, M. Mohseni, and P. Rebentrost, [arXiv:1307.0411](https://arxiv.org/abs/1307.0411) (2013).

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

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

- [7] N. Wiebe, A. Kapoor, and K. Svore, [arXiv:1401.2142](https://arxiv.org/abs/1401.2142) (2014).
- [8] V. Dunjko, J. M. Taylor, and H. J. Briegel, *Phys. Rev. Lett.* **117**, 130501 (2016).
- [9] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics (Springer, New York, 2006), p. 738.
- [10] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, MA, 2012).
- [11] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, [arXiv:1812.03584](https://arxiv.org/abs/1812.03584) (2018).
- [12] A. P. Dempster, N. M. Laird, and D. B. Rubin, *J. Roy. Stat. Soc. Ser. B* **39**, 1 (1977).
- [13] H. Miyahara and K. Tsumura, in *Proc. of the American Control Conference (ACC'16), Boston, MA* (IEEE, Piscataway, NJ, 2016).
- [14] H. Miyahara, K. Tsumura, and Y. Sughiyama, in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC'16)* (IEEE, Los Alamitos, CA, 2016), pp. 4674–4679.
- [15] H. Miyahara, K. Tsumura, and Y. Sughiyama, *J. Stat. Mech.: Theory Exp.* (2017) 113404.
- [16] H. Miyahara and Y. Sughiyama, *Phys. Rev. A* **98**, 022330 (2018).
- [17] I. Kerenidis, A. Luongo, and A. Prakash, [arXiv:1908.06657](https://arxiv.org/abs/1908.06657) [quant-ph] (2019).
- [18] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, *Contemp. Math.* **305**, 53 (2002).
- [19] This algorithm has no specific name in Ref. [7], and it is called median evaluation in Ref. [11]. But this algorithm realizes majority voting, so we call it mode evaluation.
- [20] S. Chakraborty, A. Gilyén, and S. Jeffery, [arXiv:1804.01973](https://arxiv.org/abs/1804.01973) (2018).
- [21] I. Kerenidis and A. Luongo, [arXiv:1805.08837](https://arxiv.org/abs/1805.08837) (2018).
- [22] W. Hoeffding, in *The Collected Works of Wassily Hoeffding* (Springer, Berlin, 1994), pp. 409–426.
- [23] D. Cornell and S. Sastry, Performance Comparison of K-Means and Expectation Maximization with Gaussian Mixture Models for Clustering EE6540 Final Project (2015), see <https://pdfs.semanticscholar.org/0c23/163053af3eab73fc49c5ad4d6612b76a0259.pdf>.
- [24] In the δ - k -means algorithm, we added Gaussian noise whose mean is 0 and variance is 0.01 to each element of all the centroids. In the M step of the δ -EM algorithm, we added Gaussian noise whose mean is 0 and variance is 0.01 to μ^k and each element of μ^k and add Gaussian noise whose mean is 0 and variance is 0.001 to each element of Σ^k . When π is not normalized, we normalized π . When Σ^k is not symmetric, we symmetrize Σ^k by $[\Sigma^k + (\Sigma^k)^\top]/2$. When the estimated Σ^k has a negative eigenvalue σ^* , we make all eigenvalues positive by adding $|\sigma^*|I^k$, where I^k is the $k \times k$ identity matrix.
- [25] In Fig. 3, we draw the blue ellipses by moving θ ($0 \leq \theta < 2\pi$) in $f(\theta) := \mu^k + \sqrt{\sigma_1^k} e_1^k \cos(\theta) + \sqrt{\sigma_2^k} e_2^k \sin \theta$, where σ_1^k and σ_2^k are the eigenvalues of Σ^k and e_1^k and e_2^k are the corresponding eigenvectors of Σ^k , respectively, for $k = 1, 2$. In Fig. 5, we also draw blue ellipses by the same procedure.
- [26] The hidden variable in the GMM means the index of the Gaussian functions. Note that the GMM has the symmetry on swapping the indices; then we use the maximum value on swapping.
- [27] L. M. Sieberer and W. Lechner, *Phys. Rev. A* **97**, 052329 (2018).
- [28] C. Dłaska, L. M. Sieberer, and W. Lechner, *Phys. Rev. A* **99**, 032342 (2019).
- [29] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, [arXiv:1904.10246](https://arxiv.org/abs/1904.10246) (2019).
- [30] V. Giovannetti, S. Lloyd, and L. Maccone, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [31] V. Giovannetti, S. Lloyd, and L. Maccone, *Phys. Rev. A* **78**, 052310 (2008).