Revisiting Online Quantum State Learning

Feidiao Yang, Jiaqing Jiang, Jialin Zhang, Xiaoming Sun

Institute of Computing Technology, Chinese Academy of Sciences, Bejing, China

University of Chinese Academy of Sciences, Beijing, China

{yangfeidiao, jiangjiaqing, zhangjialin, sunxiaoming}@ict.ac.cn

Abstract

In this paper, we study the online quantum state learning problem which is recently proposed by Aaronson et al. (2018). In this problem, the learning algorithm sequentially predicts quantum states based on observed measurements and losses and the goal is to minimize the regret. In the previous work, the existing algorithms may output mixed quantum states. However, in many scenarios, the prediction of a pure quantum state is required. In this paper, we first propose a Follow-the-Perturbed-Leader (FTPL) algorithm that can guarantee to predict pure quantum states. Theoretical analysis shows that our algorithm can achieve an $O(\sqrt{T})$ expected regret under some reasonable settings. In the case that the pure state prediction is not mandatory, we propose another deterministic learning algorithm which is simpler and more efficient. The algorithm is based on the online gradient descent (OGD) method and can also achieve an $O(\sqrt{T})$ regret bound. The main technical contribution of this result is an algorithm of projecting an arbitrary Hermitian matrix onto the set of density matrices with respect to the Frobenius norm. We think this subroutine is of independent interest and can be widely used in many other problems in the quantum computing area. In addition to the theoretical analysis, we evaluate the algorithms with a series of simulation experiments. The experimental results show that our FTPL method and OGD method outperform the existing RFTL approach proposed by Aaronson et al. (2018) in almost all settings. In the implementation of the RFTL approach, we give a closed-form solution to the algorithm. This provides an efficient, accurate, and completely executable solution to the RFTL method.

Introduction

The interdisciplinary research between quantum computing and machine learning is becoming an attractive area in recent years (Biamonte et al. 2017; Lloyd, Mohseni, and Rebentrost 2014). On one hand, people expect to take advantage of the great power of quantum computers to improve the efficiency of the algorithms in big data processing and machine learning. One representative example following this idea is the HHL algorithm (Harrow, Hassidim, and Lloyd 2009). On the other hand, machine learning algorithms and theories may also help solve some interesting problems in quantum computation and quantum information (Carleo and Troyer 2017). In this paper, we apply the online learning theory to solve an interesting problem of learning an unknown quantum state.

Learning an unknown quantum state is a fundamental problem in quantum computation and quantum information. The basic version is the **quantum state tomography** problem (Vogel and Risken 1989), which aims to fully recover the classical description of an unknown quantum state. Although quantum state tomography gives a complete characterization of the target state, it is quite costly. Recent advancement showed that fully reconstructing an unknown quantum state in the worst case needs exponential copies of the state (Haah et al. 2016; Odonnell and Wright 2016).

However, in some applications, it is unnecessary to fully reconstruct an unknown quantum state. Some side information is sufficient. Therefore, some learning tasks move on to learn the success probabilities of applying a collection of two-outcome measurements to an unknown state, with respect to some metrics. Of which, the **shadow tomography** problem (Aaronson 2018) requires to estimate the success probabilities uniformly over all measurements in the collection. Aaronson (2018) showed that the required number of copies of the unknown state in the shadow tomography is nearly linear to the number of qubits and poly-logarithmic in terms of the number of the measurements.

More generally, it may not need to estimate the success probabilities within an error uniformly over all two-outcome measurements. Following the idea of the statistical learning theory, we may assume that there is a distribution over some possible two-outcome measurements. And our goal is to learn a quantum state such that the expected difference between the success probabilities of applying a measurement sampled from the distribution to the learned state and the target state respectively is within a specific error. This is called the statistical learning model or the PAC-learning model of quantum states. Aaronson (2007) proved that the number of samples for the PAC-learning of quantum states only grows linearly with the number of qubits of the state, which is surprisingly an exponential reduction compared with the full quantum state tomography.

However, the assumption that there is a distribution over some two-outcome measurements and the data are i.i.d. samples from this distribution does not always hold. The Environment may change over time or it is even adversarial. Complementary to the statistical learning theory, the online learning theory is good at coping with arbitrary or even adversarial sequential data. Therefore, Aaronson et al. (2018) further proposed the model of online quantum states learning. In this model, the data, such as measurements and losses, are provided sequentially. The learning algorithm is to predict a series of quantum states interactively. Its goal is to minimize the regret, which is the difference in the total loss between the learning algorithm and the best fixed quantum state in hindsight.

Although the existing theory has provided helpful ideas for this problem, it is still a challenge to design and analyze algorithms for online quantum state learning. For example, a feasible solution in conventional online learning is often a vector in the real Euclidean space, but a feasible solution in quantum state learning is a complex matrix with special constraints. Besides, a direct adaption of the existing techniques can not utilize the properties of the quantum setting. If we can take advantage of these unique features or leverage techniques from quantum computing, we may get better results or different solutions.

In (Aaronson et al. 2018), the authors proposed three very different approaches. They evaluated the algorithms with two metrics, the regret in online learning and the number of errors. First, they adapted the Regularized Follow-the-Leader (RFTL) algorithm (Abernethy, Hazan, and Rakhlin 2008; Shalev-Shwartz and Singer 2007) to the online quantum state learning. Particularly, they employed the negative von Neumann entropy as the regularization. The RFTL method can achieve an $O(L\sqrt{nT})$ regret, where L is the Lipschitz coefficient of the loss functions, n is the number of qubits of a state, and T is the time horizon of the learning process. Its number of errors is $O(\frac{n}{\varepsilon^2})$ under some assumptions. The RFTL method has the best theoretical guarantee over the other two methods. Their second method employs the technique of postselection-based learning procedure (Aaronson 2005). Its error number is bounded as $O(\frac{n}{\varepsilon^3}\log\frac{n}{\varepsilon})$ and the regret bound is not available. Their third method is based on an upper-bound on the sequential fat-shattering dimension of quantum states (Nayak 1999; Ambainis et al. 2002; Rakhlin, Sridharan, and Tewari 2015). It achieves a regret bound of $O(L\sqrt{nT}\log^{3/2}T)$. What weird is that this result is non-explicit and non-constructive, that is, we actually do not have any specific algorithm corresponding to this result. Seeing from this discussion, although the latter two methods highly utilize the techniques of quantum computing and they do not rely on the theory of online learning, they actually do not outperform the typical RFTL method. This shows the great power of machine learning and optimization theories on solving interesting problems in quantum computation and quantum information.

In this paper, we revisit the problem of online quantum state learning from the perspective of online convex optimization.

First, predicting a pure quantum state is of special interest in quantum state learning (Lee, Lee, and Bang 2018; Benedetti et al. 2019) since a pure state has its unique value in theory and practice. However, the existing RFTL method cannot make such a guarantee since its prediction is always a full rank matrix, which corresponds to a mixed state. It is still a challenge to predict pure states in online quantum state learning. In this paper, we propose a Follow-the-Perturbed-Leader (FTPL) algorithm (Kalai and Vempala 2005) that can guarantee to predict pure quantum states every round. The key idea is to formulate the optimization objective with a stochastic linear regularization to be a special semi-definite programming (SDP), and we show this SDP always has a rank-1 solution which is corresponding to a pure quantum state. Our analysis shows that the regret with respect to the expected prediction is bounded as $O(\sqrt{T})$. We further adapt the FTPL method to a typical and reasonable setting with L_1 loss. In this case, our FTPL method can achieve an $O(\sqrt{T})$ expected regret.

Second, if pure states are not mandatory, the online gradient descent (OGD) method (Zinkevich 2003) is a simple and efficient approach for this problem. Actually, it is widely used in practice for online learning. However, the OGD method relies on a subroutine of projection if the feasible solutions are constrained. In this paper, we propose an algorithm of projecting an arbitrary Hermitian matrix onto the set of quantum states with respect to the Frobenius norm. The key idea is to reduce the problem to project a vector onto the probability simplex. Our algorithm is exact and efficient. It could be widely used as a subroutine in many other problems in quantum computing. We apply our method to the projected online gradient descent algorithm for quantum state learning and we show that this method can achieve an $O(\sqrt{T})$ regret.

Third, the RFTL method due to Aaronson et al. (2018) relies on an offline oracle of solving a linear optimization with the negative von Neumann entropy regularization, which is not fully discussed in the original work. In this paper, we give a closed-form solution to this offline convex optimization problem. Our result provides an efficient, accurate, and completely executable solution to the RFTL method. We also implement this solution in our experiments.

Last but not least, we conduct a series of simulation experiments to evaluate the algorithms. In our experiments, the FTPL method and the OGD method outperforms the RFTL approach in almost all settings.

Preliminaries

For readers unfamiliar with the field of quantum computing and making this paper self-contained, in this section, we introduce some basic concepts in quantum computation and quantum information that are necessary for this paper. Interested readers are recommended to the celebrated textbook by Nielsen and Chuang (2002).

Quantum state   Unlike in classical computers that a bit is in a deterministic state of either 0 or 1, in quantum computing, a **quantum bit** (**qubit** henceforth) could be in a superposition of 0 and 1. The state of a qubit could be de-

scribed by a unit vector $(\alpha_0, \alpha_1)^\mathsf{T} \in \mathbb{C}^2$, where $|\alpha_0|^2 + |\alpha_1|^2 = 1$, meaning that when measuring the qubit, we may observe the result 0 with probability $|\alpha_0|^2$ or the result 1 with probability $|\alpha_1|^2$. Similarly, the state of an n-qubit system could be described by a unit vector $(\alpha_0, \dots, \alpha_{2^n-1})^\mathsf{T} \in \mathbb{C}^{2^n}$. When measuring this system, we may observe the result $(i)_2$ with probability $|\alpha_i|^2$ for $i = 0, \dots, 2^n - 1$, where $(i)_2$ is the n-bit binary representation of integer i.

Pure state, mixed state, and density matrix An n-qubit system could be in a more complicated state than the state introduced above. It can be a distribution over those "simple" states. More precisely, it could be in a "simple" state $\Psi_i \in \mathbb{C}^{2^n}$ with probability p_i for some i. For distinguishing these cases, we call a state corresponding to a unit vector a **pure state** and a state that is a distribution over multiple pure states a **mixed state**.

The representation of pure states and mixed states can be unified with a mathematical tool called **density matrix** (or **density operator**). Specifically, if a quantum state is in a state $\Psi_i \in \mathbb{C}^{2^n}$ with probability p_i, its density matrix is a $d \times d$ complex matrix ρ, where $d = 2^n$, defined as

$$\rho = \sum_i p_i \Psi_i \Psi_i^\dagger,$$

in which $\Psi_i^\dagger$ is the adjoint conjugate of Ψ_i.

It can be showed that a $2^n \times 2^n$ complex matrix ρ is a density matrix (quantum state) if and only if it is Hermitian, positive semi-define (PSD), and with trace 1. Further, a density matrix ρ represents a pure state if and only if $\mathrm{rank}(\rho) = 1$.

The set of density matrices of n-qubit states is denoted as

$$\mathcal{K}_n = \left\{ \rho \in \mathbb{C}^{2^n \times 2^n} \mid \rho = \rho^\dagger, \rho \succeq 0, \mathrm{Tr}(\rho) = 1 \right\}.$$

We abbreviate $\mathcal{K}_n$ as $\mathcal{K}$ when n is clear in the context. It is straightforward to verify that $\mathcal{K}_n$ is a convex set.

Two-outcome measurement Quantum measurement is a means to extract classical (observable) information from a quantum state. While quantum measurement is a broad topic, in this paper we only consider a special case called **two-outcome measurement** (Aaronson 2018; Aaronson et al. 2018). When applying a two-outcome measurement to a quantum state, it succeeds with a specific probability. Mathematically, a two-outcome measurement could be described by a Hermitian PSD matrix $E \in \mathbb{C}^{d \times d}$ with eigenvalues in $[0, 1]$. When applying E to a quantum state ρ, it gets a successful result (Yes or 1) with probability $\mathrm{Tr}(E\rho)$, or a failed result (No or 0) with probability $1 - \mathrm{Tr}(E\rho)$.

Problem Description and Settings

Online quantum state learning is a sequential prediction process with interactions between a player (algorithm) and an adversary over T rounds. In round t, the player predicts a quantum state $\omega_t \in \mathcal{K}$. Simultaneously the adversary chooses and reveals a two-outcome measurement E_t and a loss function $\ell_t : \mathbb{R} \to \mathbb{R}$. At the end of this round the player suffers a loss $\ell_t(\mathrm{Tr}(E_t\omega_t))$.

As a convention in online learning, we assume the loss function ℓ_t is convex and L-Lipschitz.

We assume that the adversary can access the player's strategy, without knowing its random numbers if it is a randomized algorithm. Generally, the adversary can select E_t and ℓ_t in an arbitrary way. We call it the oblivious setting if E_t and ℓ_t are chosen before the game playing, and the adaptive setting is that E_t and ℓ_t may depend on the player's predictions in previous rounds.

It is called the realizable case if there is an underlying unknown quantum state ρ to be learned and the loss function ℓ_t is relevant to ρ, while in the non-realizable case the loss functions need not be consistent with any quantum state (Aaronson et al. 2018).

Common loss functions are the L_1 loss (absolute loss) defined as

$$\ell_t(x) := |x - b_t|,$$

and the L_2 loss (square loss) given by

$$\ell_t(x) := (x - b_t)^2.$$

In the realizable case, b_t may be an approximation to the success probability of applying E_t to ρ, or a Bernoulli random variable corresponding to the measurement result.

While we generally discuss the generic settings in this paper, we pay particular attention to a case with L_1 loss and $b_t \in \{0, 1\}$. We argue that this is a typical, realistic, and reasonable setting since in physics we can only observe the measurement results of the unknown state.

In online learning, we use the **regret** to evaluate the performance of an algorithm. In our problem, it is defined as

$$\mathrm{regret}_T = \sum_{t=1}^T \ell_t(\mathrm{Tr}(E_t\omega_t)) - \min_{\omega \in \mathcal{K}} \sum_{t=1}^T \ell_t(\mathrm{Tr}(E_t\omega)).$$

The regret is the difference in the total loss between the learning algorithm and the best fixed prediction in hindsight. A good algorithm should have a sublinear regret in terms of the time horizon T, that is, the average regret per round is approaching to 0 when T trends to infinity.

Many online learning algorithms utilize the gradients of the loss functions. In our setting, the gradient of the loss ℓ_t with respect to the point ω_t is denoted as

$$\nabla_t = \frac{\mathrm{d}}{\mathrm{d}\omega_t} \ell_t(\mathrm{Tr}(E_t\omega_t)) = \ell_t'(\mathrm{Tr}(E_t\omega_t))E_t.$$

Our Results

We present our work in this section. In the first subsection, we discuss the FTPL method of predicting pure quantum states. The second subsection proposes a projection algorithm onto the set of density matrices and applies it to the OGD method. In the third subsection, we provide a closed-form solution to the RFTL method.

Follow-the-Perturbed-Leader Algorithm for Pure Quantum State Prediction

Predicting a pure quantum state is of special interest in quantum state learning (Lee, Lee, and Bang 2018; Benedetti et al. 2019) since a pure quantum state has its unique value in theory and practice. For example, if we want to utilize the

learning results by preparing or transporting them, the device or resource for pure states are more economical than that for mixed states.

However, although the RFTL method has a good regret guarantee, it always predicts mixed quantum states since we can show that ω_t is full-rank. This can also be implied from Claim 14 in (Aaronson et al. 2018). Similarly, the OGD method which will be introduced shortly cannot make such promise as well. Therefore, it is still a challenge to predict ω_t as a pure quantum state.

We realize that by adding a stochastic linear regularization instead of the deterministic convex regularization as in the RFTL method, we can always guarantee pure state prediction, leading to a Follow-the-Perturbed-Leader algorithm depicted in Algorithm 1.

Algorithm 1 FTPL algorithm for pure state prediction

1: **for** $t = 1$ to T **do**
2:     Sample a random Hermitian matrix Z_t
3:     Predict

$$\omega_t = \arg\min_{\omega \in \mathcal{K}} \left\{ \eta \sum_{s=1}^{t-1} \mathrm{Tr}(\nabla_s \omega) - \mathrm{Tr}(Z_t \omega) \right\}. \quad (1)$$

4: **end for**

As claimed above, Algorithm 1 can always predict pure states, which is formally stated as the following theorem.

Theorem 1. *The mathematical programming (1) in Algorithm 1 always has a rank-1 solution. In this case, ω_t is a pure quantum state.*

An explicit algorithm to solve the mathematical programming in (1) naturally gives a constructive proof to this theorem, which we present as follows.

Proof. The optimization problem in (1) is actually a semi-definite programming problem. While SDP is difficult in general, this special one is easy to solve.

Since ∇_s and Z_t are Hermitian, we have the following diagonal decomposition

$$\eta \sum_{s=1}^{t-1} \nabla_s - Z_t = U \Lambda U^\dagger,$$

where Λ is a diagonal matrix and U is a unitary matrix. The SDP is transformed to a diagonal form as

$$\min_{\omega' \in \mathcal{K}} \mathrm{Tr}(\Lambda \omega'),$$

in which $\omega' = U^\dagger \omega U$. This can be further translated to a linear programming problem

$$\min_{x \in \Delta_d} \lambda^\mathsf{T} x,$$

where $\lambda = (\Lambda_1, \ldots, \Lambda_d)^\mathsf{T}$ is the diagonal vector of Λ. While solving linear programming is costly in general, this one is quite trivial since the feasible set is the probability simplex. The solution x could be set in such a way that it has only one

element with value 1 corresponding to the smallest element in λ and other elements are 0. Finally the solution to the original SDP $\omega_t = U \mathrm{diag}(x) U^\dagger$ is a rank-1 density matrix, corresponding to a pure quantum state. □

Before discussing the regret of the FTPL method, we turn to the construction of the random Hermitian matrix Z_t. While the exponential distribution is commonly used in conventional FTPL method (Kalai and Vempala 2005; Neu and Bartók 2016), we employ the normal distribution in our algorithm since the former depends on the assumption that the elements in the gradients are all non-negative which does not hold in our setting.

A random Hermitian matrix $Z \in \mathbb{C}^{d \times d}$ can be sampled as follows. First its diagonal elements $Z_{i,i} \sim \mathcal{N}(0, 1)$ for $i = 1, \ldots, d$. And then its upper triangle is filled as $Z_{i,j} \sim \mathcal{N}(0, \frac{1}{2}) + i\mathcal{N}(0, \frac{1}{2})$ for $1 \leqslant i < j \leqslant d$. Finally its lower triangle is completed by the definition of Hermitian that $Z_{j,i} = Z_{i,j}^*$ for $1 \leqslant i < j \leqslant d$, where $Z_{i,j}^*$ is the conjugate of the complex number $Z_{i,j}$. All coefficients in Z are independent. We denote such a distribution over Hermitian matrices as $\mathcal{D}_\mathcal{N}$.

Now we turn back to the regret analysis of the FTPL method. For simplicity, we only consider the oblivious setting in which Z_t needs not to be re-sampled every round. Since it is a randomized algorithm, here we provide a result in terms of the regret with respect to the expected prediction, that is, if we predict $\overline{\omega}_t = \mathbb{E}[\omega_t]$ in round t, we have the following statement.[1]

Theorem 2. *Suppose $Z \sim \mathcal{D}_\mathcal{N}$ and $Z_1 = \cdots = Z_T = Z$. By setting $\eta = \frac{1}{L\sqrt{T}}$, the regret of algorithm 1 with respect to the expected prediction is bounded as*

$$\sum_{t=1}^{T} \ell_t(\mathrm{Tr}(E_t \mathbb{E}[\omega_t])) - \min_{\omega \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(\mathrm{Tr}(E_t \omega)) = O(Ld\sqrt{T}).$$

Algorithm 1 could be seen as a variant of the algorithm implied from Theorem 2 that it takes only one sample instead of the expectation as the prediction. Since the exact expectation is usually inaccessible, it is reasonable in practice to take samples to replace the expectation as in many Monte Carlo methods. It is an interesting problem to analyze the relationship between the regret gap and the number of samples.

In a special case that the loss function is L_1 loss and $b_t \in \{0, 1\}$, we can adapt the FTPL method to achieve a better theoretical result that it provides an expected regret guarantee. As we argue before, this is a typical and reasonable setting and it has practical value.

In this case, the L_1 loss becomes a linear loss as

$$\ell_t(\mathrm{Tr}(E_t \omega_t)) = (1 - b_t)\mathrm{Tr}(E_t \omega_t) + b_t \mathrm{Tr}((I_d - E_t)\omega_t)$$
$$= \mathrm{Tr}\left(((1 - 2b_t)E_t + b_t I_d)\omega_t\right),$$

where I_d is the $d \times d$ identity matrix. The corresponding gradient is modified to be

$$\nabla_t = (1 - 2b_t)E_t + b_t I_d. \quad (2)$$

---

[1]Due to the space limitation, the detailed proofs to our main theorems are provided in an extended version of this paper, which will be available on the Internet.

By linearity we have

$$\ell_t(\text{Tr}(E_t\mathbb{E}[\omega_t])) = \mathbb{E}\left[\ell_t(\text{Tr}(E_t\omega_t))\right].$$

Applying this observation to Algorithm 1 and Theorem 2, consequently we have the following expected regret bound.

Theorem 3. *For the setting with L_1 loss and $b_t \in \{0, 1\}$, the FTPL algorithm (Algorithm 1) running with the gradients defined in (2) and $Z_1 = \ldots = Z_d \sim \mathcal{D}_\mathcal{N}$ achieves an expected regret bound as*

$$\mathbb{E}\left[\text{regret}_T\right] = \mathbb{E}\left[\sum_{t=1}^{T}\ell(\text{Tr}(E_t\omega_t)) - \min_{\omega\in\mathcal{K}}\sum_{t=1}^{T}\ell(\text{Tr}(E_t\omega))\right]$$
$$= O(Ld\sqrt{T}).$$

This is a more direct result than Theorem 2 since it is completely consistent with the algorithm.

Projection onto Density Matrices and Its Application to Online Gradient Descent

Online gradient descent is a simple and efficient method in online learning and therefore it is widely used in practice. Besides, the OGD method has other advantages. For example, it can achieve a logarithmic regret for strongly convex loss functions (Hazan, Agarwal, and Kale 2007), which is a significant improvement compared with the square-root regret in general cases. However, the OGD method relies on a subroutine of projection for constrained optimization since one step of an update may lead to a solution outside of the feasible region so we have to pull it back.

Projection is an operation of finding the nearest point in a set $\mathcal{S}$ to a given point y. Formally, it is defined as

$$\Pi_\mathcal{S}(y) = \arg\min_{x\in\mathcal{S}}\|x - y\|.$$

It is well known that projection is an important subroutine in many optimization and machine learning algorithms. However, projection is a tricky operation since there is no generic approach and the algorithm, as well as the computational cost, heavily relies on the properties of the feasible set. For example, it is quite easy to project a point onto a ball, but projecting a point onto a general polytope is a linear programming problem and it is quite costly.

In this section, we devise an algorithm that projects an arbitrary Hermitian matrix onto the set of density matrices with respect to the Frobenius norm. This operation is an analogue of the projection in vector space with respect to the Euclidean norm.

Our approach is described in Algorithm 2.[2] The main idea is to project the spectrum vector of the input matrix onto the probability simplex with respect to the Euclidean norm, which is a well-studied and efficient process (Chen and Ye 2011; Wang and Carreiraperpinan 2013).

The soundness of Algorithm 2 is stated as the following theorem.

---

[2]We are grateful to the reviewer for pointing out that some similar result was independently developed recently, see (Gonçalves, Gomes-Ruggiero, and Lavor 2016). In addition, some relevant projection methods were developed under special assumptions such as sparsity, see (Bolduc et al. 2017; Kyrillidis et al. 2013).

Algorithm 2 Projection onto the set of density matrices

1: **Input**: a $d \times d$ Hermitian matrix A (no matter PSD or not)
2: Decompose A into its diagonal form $A = UQU^\dagger$
3: Let vector $q = (Q_{1,1}, \ldots, Q_{d,d})^\mathsf{T}$
4: Project vector q onto the d-dimensional probability simplex Δ_d and get $\lambda = (\lambda_1, \ldots, \lambda_d)^\mathsf{T}$, that is

$$\lambda = \arg\min_{x\in\Delta_d}\|x - q\|_2$$

5: Let diagonal matrix $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$
6: **return** $A' = U\Lambda U^\dagger$

Theorem 4. *Suppose A is an arbitrary $d \times d$ Hermitian matrix, then the matrix A' produced by Algorithm 2 is a density matrix and it satisfies*

$$\|A - A'\|_F \leqslant \|A - \rho\|_F$$

for any $d \times d$ density matrix ρ.

Our algorithm is computationally efficient. Since the process in step 4 of projecting a vector onto the probability simplex takes $O(d\log d)$ time, the major part in the computational cost is from the process of spectrum decomposition in step 2 and the matrix multiplication in step 6. We think this problem is of independent interest and our method could be widely used as a subroutine in many other problems in quantum computing.

With the projection method we propose above, we can design a projected online gradient descent algorithm for online quantum state learning, which is depicted in Algorithm 3.

Algorithm 3 OGD method for online quantum state learning

1: **Input**: T, $\omega_1 = I_d/d$, step size $\{\eta_t\}$
2: **for** $t = 1$ to T **do**
3:     Predict ω_t and observe measurement E_t as well as loss function ℓ_t.
4:     Update and project:

$$\omega_{t+1} = \Pi_\mathcal{K}(\omega_t - \eta_t\ell_t'(\text{Tr}(E_t\omega_t))E_t).$$

5: **end for**

By adapting the standard technique of analyzing the OGD method (Hazan 2016) to our setting of learning quantum states, Algorithm 3 achieves an $O(\sqrt{T})$ regret bound, which is stated formally in Theorem 5.

Theorem 5. *By setting $\eta_t = \eta = \frac{1}{Ld}\sqrt{\frac{d-1}{T}}$, the regret of Algorithm 3 is bounded as*

$$\text{regret}_T \leqslant L\sqrt{(d-1)T}.$$

Closed-form Solution to the RFTL Algorithm

The basic idea of the RFTL method due to Aaronson et al. (2018) is to predict a density matrix that minimizes a linear

loss with an additional negative von Neumann entropy regularization. Specifically, in round t the algorithm predicts

$$\omega_t := \arg\min_{\omega \in \mathcal{K}} \left\{ \eta \sum_{s=1}^{t-1} \mathrm{Tr}\left(\nabla_s \omega\right) + \sum_{i=1}^{2^n} \lambda_i(\omega) \log \lambda_i(\omega) \right\}. \quad (3)$$

Aaronson et al. (2018) showed that the regret of the RFTL algorithm can be upper-bounded as $2L\sqrt{(2\log 2)nT}$.

Although Aaronson et al. (2018) stated that the mathematical programming in (3) is convex and it can be solved efficiently, there is still something to do with this offline convex optimization problem to provide an end-to-end solution to the RFTL method. First, it is a constrained optimization problem and the feasible set is the set of density matrices. By utilizing some typical convex optimization algorithms such as the projected gradient descent, we have to design a subroutine of projecting a matrix onto the set of density matrices, as we just did. Second, the gradient of the objective function in (3) at density matrix ω is $\eta \sum_{s=1}^{t-1} \nabla_s + I + \log \omega$, whose norm is unbounded since ω is an arbitrary density matrix and its eigenvalues can be arbitrarily close to 0. This is inconsistent with the bounded-gradient assumption (Lipschitz condition) of some convex optimization algorithms and their analysis.[3] Third, generic convex optimization algorithms are often iterative processes. Only an approximate solution is found after finite steps and the optimal solution cannot be guaranteed. There is a theoretical gap between an approximate solution and the optimal assumption in the analysis of the RFTL method, not to mention that it may take much time to find an acceptable solution.

Fortunately, we can circumvent all the issues by providing a closed-form solution to the convex optimization in (3), as

$$\omega_t := \frac{\exp\left(-\eta \sum_{s=1}^{t-1} \nabla_s\right)}{\mathrm{Tr}\left(\exp\left(-\eta \sum_{s=1}^{t-1} \nabla_s\right)\right)}. \quad (4)$$

We formally state the result in the following theorem.

Theorem 6. *Suppose ∇_s are Hermitian matrices for $s = 1, \ldots, t-1$, and $\mathcal{K}$ is the set of density matrices. ω_t given in equation (4) is the optimal solution of the mathematical programming problem in (3).*

Our result provides an efficient, accurate, and completely executable solution to the RFTL algorithm. Our solution has several advantages over using a generic offline convex optimization oracle to solve (3). First, it is not iterative and it produces an exact optimal solution, while generic iterative optimization algorithms do not guarantee. Second, it is computationally efficient since it only takes one step of diagonal decomposition, while every step in generic optimization algorithms may require such operation. Besides, our solution does not depend on the subroutine of projection.

Experiments

In addition to the theoretical analysis discussed above, in this section, we evaluate the algorithms of online quantum state learning with a series of simulation experiments.

Experimental Settings

We implement and compare the three algorithms we discussed, the FTPL method, the OGD method, and the RFTL method with our closed-form solution.

In this experiment, we consider a typical and reasonable setting described as follows. First, it is a realizable setting, that is, there is an underlying unknown quantum state ρ, pure or mixed, to be learned. The loss functions ℓ_t are determined by the measurements applied to ρ, although the measurements could be chosen randomly or adversarially. Specifically, we consider the absolute loss $\ell_t(z) = |z - b_t|$, in which b_t is a Bernoulli random variable with expectation $\mathbb{E}[b_t] = \mathrm{Tr}(E_t \rho)$, corresponding to the result of measuring ρ with E_t.

Evaluating online learning algorithms without live data is a challenge, especially for the adversarial settings, since it is quite difficult to find out the worst case to fool an algorithm. For this purpose, we propose an adaptively adversarial data generation policy to select E_t. Specifically, for a given algorithm and a time step t, the adversary can predict the output of the algorithm, denoted as $\widetilde{\omega}_t$, since the adversary can access the code of the algorithm. And then the adversary chooses a two-outcome measurement E_t that maximizes the difference between $\mathrm{Tr}(E_t \widetilde{\omega}_t)$ and $\mathrm{Tr}(E_t \rho)$, that is, $E_t = \arg\max_E |\mathrm{Tr}(E(\widetilde{\omega}_t - \rho))|$. We call such data generation policy an immediate-punishment adversary. In addition, we also consider the stochastic data generation policy in which E_t is generated randomly. This is the most natural way to evaluate the algorithms. In short, we implement four different data generation policy, the stochastic policy and the immediate-punishment adversarial policies against the FTPL method, the OGD method, and the RFTL method, respectively.

Regarding other experimental parameters, we take the number of qubits $n = 4$, so the dimension of the density matrices is 16×16. For each experiment, we run for 100 trials with randomly generated target states ρ and report the average curves.

Experimental Results

The experimental results are illustrated in Figure 1. Each sub-figure corresponds to a specific experimental setting, labeled with its sub-caption. In each sub-caption, the first term corresponds to the type of the underlying quantum state, "pure" for a pure state and "mixed" for a mixed state. The second term denotes the data generation policy, "stochastic" for the stochastic data policy, "FTPL" for the adversarial data policy against the FTPL method, and so on.

---

[3]Some recent advancement pointed out that the bounded gradient assumption is unnecessary in some special cases in stochastic programming (Nguyen et al. 2018). Thanks the reviewer for providing this reference.

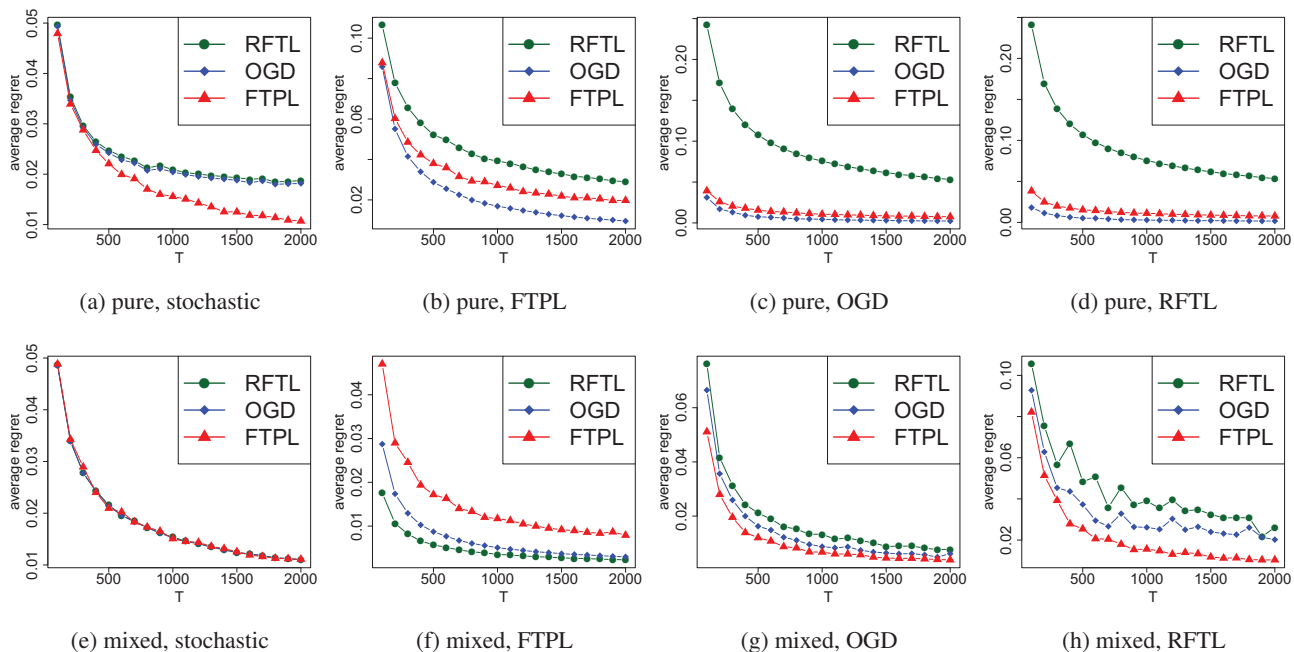| (a) pure, stochastic | (b) pure, FTPL | (c) pure, OGD | (d) pure, RFTL |
| (e) mixed, stochastic | (f) mixed, FTPL | (g) mixed, OGD | (h) mixed, RFTL |

Figure 1: The experimental results with the metric of average regret. Each sub-figure corresponds to an adversarial setting. The first term is the type of the target state and the second term is what algorithm the adversary is against for.

We report the experimental results in terms of the **average regret**, that is, regret_T/T, which should converge to 0 for algorithms with a sublinear regret. In a sub-figure, the horizontal axis corresponds to the time horizon T and the vertical axis denotes the average regret. A red curve is for the FTPL method, a blue curve is for the OGD method, and a green curve is for the RFTL method.

Seeing from the experimental results, we can observe that our FTPL method outperforms the RFTL approach in almost all settings, not to mention its unique merit of predicting pure states. Specifically, when the target state is pure, the FTPL method performs much better than the RFTL method, even though the data policy is adversarial against it. What surprising is that the FTPL method performs the best even though the target state is mixed, as long as the adversary is not against it.

Besides, the performance of the OGD method is quite good and stable, even though the adversary is against it. It also outperforms the RFTL method in almost all settings, which demonstrates the value of the simple and efficient OGD method in practice.

Summary and Future Work

In this paper, we revisit the problem of online quantum state learning in the perspective of online convex optimization.

First, we propose a Follow-the-Perturbed-Leader algorithm that can guarantee to predict pure states, which is of special value in quantum state learning. Our analysis shows that the regret with respect to the expected prediction is bounded as $O(\sqrt{T})$. We further adapt the FTPL method to a

typical and reasonable setting with L_1 loss. In this case, the FTPL method can achieve an $O(\sqrt{T})$ expected regret.

Second, we propose a simple and efficient online gradient descent algorithm for online quantum state learning which can achieve an $O(\sqrt{T})$ regret bound. The OGD method is based on an algorithm of projecting an arbitrary Hermitian matrix onto the set of density matrices, which could be widely used as a subroutine in many other problems in quantum computing.

Third, we give a closed-form solution to the existing RFTL method. Our result provides an efficient, accurate, and completely executable solution to the RFTL method.

In addition to the theoretical analysis, we also conduct a series of simulation experiments to evaluate the algorithms. The experimental results show that our FTPL and OGD method outperforms the RFTL method in almost all settings, not to mention the unique merit of the FTPL method of predicting pure states. These results demonstrate the value of our methods in practice.

As for the future work, there are several challenges and interesting problems in this topic. Compared with the RFTL method, although our methods have some merit such as pure state prediction, the regret bounds in our results are worse in terms of the number of qubits n, since $d = 2^n$. We think it is due to the loose analysis. Tighter analysis to the FTPL method and the OGD method in terms of the number of the qubits n is a challenge. Besides, finding the lower bounds in different settings is also helpful to understand the problem of online quantum state learning.

Acknowledgments

References

Aaronson, S.; Chen, X.; Hazan, E.; Kale, S.; and Nayak, A. 2018. Online learning of quantum states. *neural information processing systems* 8962–8972.

Aaronson, S. 2005. Limitations of quantum advice and one-way communication. *Theory of Computing* 1(1):1–28.

Aaronson, S. 2007. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463(2088):3089–3114.

Aaronson, S. 2018. Shadow tomography of quantum states. *symposium on the theory of computing* 325–338.

Abernethy, J. D.; Hazan, E.; and Rakhlin, A. 2008. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 263–274.

Ambainis, A.; Nayak, A.; Ta-Shma, A.; and Vazirani, U. V. 2002. Dense quantum coding and quantum finite automata. *J. ACM* 49(4):496–511.

Benedetti, M.; Grant, E.; Wossnig, L.; and Severini, S. 2019. Adversarial quantum circuit learning for pure state approximation. *New Journal of Physics* 21(4):043023.

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; and Lloyd, S. 2017. Quantum machine learning. *Nature* 549(7671):195.

Bolduc, E.; Knee, G. C.; Gauger, E. M.; and Leach, J. 2017. Projected gradient descent algorithms for quantum state tomography. *npj Quantum Information* 3(1):44.

Carleo, G., and Troyer, M. 2017. Solving the quantum many-body problem with artificial neural networks. *Science* 355(6325):602–606.

Chen, Y., and Ye, X. 2011. Projection onto a simplex. *arXiv: Optimization and Control*.

Gonçalves, D. S.; Gomes-Ruggiero, M. A.; and Lavor, C. 2016. A projected gradient method for optimization over density matrices. *Optimization Methods and Software* 31(2):328–341.

Haah, J.; Harrow, A. W.; Ji, Z.; Wu, X.; and Yu, N. 2016. Sample-optimal tomography of quantum states. *symposium on the theory of computing* 913–925.

Harrow, A. W.; Hassidim, A.; and Lloyd, S. 2009. Quantum algorithm for linear systems of equations. *Physical review letters* 103(15):150502.

Hazan, E.; Agarwal, A.; and Kale, S. 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69(2-3):169–192.

Hazan, E. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2(3-4):157–325.

Kalai, A., and Vempala, S. 2005. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences* 71(3):291–307.

Kyrillidis, A.; Becker, S.; Cevher, V.; and Koch, C. 2013. Sparse projections onto the simplex. In *International Conference on Machine Learning*, 235–243.

Lee, S. M.; Lee, J.; and Bang, J. 2018. Learning unknown pure quantum states. *Physical Review A* 98(5).

Lloyd, S.; Mohseni, M.; and Rebentrost, P. 2014. Quantum principal component analysis. *Nature Physics* 10(9):631.

Nayak, A. 1999. Optimal lower bounds for quantum automata and random access codes. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, 369–377.

Neu, G., and Bartók, G. 2016. Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits. *The Journal of Machine Learning Research* 17(1):5355–5375.

Nguyen, L. M.; Nguyen, P. H.; van Dijk, M.; Richtárik, P.; Scheinberg, K.; and Takáč, M. 2018. Sgd and hogwild! convergence without the bounded gradients assumption. *arXiv preprint arXiv:1802.03801*.

Nielsen, M. A., and Chuang, I. 2002. Quantum computation and quantum information.

Odonnell, R., and Wright, J. 2016. Efficient quantum tomography ii. *symposium on the theory of computing* 899–912.

Rakhlin, A.; Sridharan, K.; and Tewari, A. 2015. Online learning via sequential complexities. *J. Mach. Learn. Res.* 16:155–186.

Shalev-Shwartz, S., and Singer, Y. 2007. A primal-dual perspective of online learning algorithms. *Machine Learning* 69(2-3):115–142.

Vogel, K., and Risken, H. 1989. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. *Physical Review A* 40(5):2847.

Wang, W., and Carreiraperpinan, M. A. 2013. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv: Learning*.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 928–936.