Quantum Fidelity Estimation Methods

Çetin İlhan KAYA

February 6, 2022

Contents

1 Introduction

In the era of quantum hype, one of the most important properties that experimentalist try to improve is the *fidelity*. Fidelity measures the distance between desired and actual states. Computationally speaking, it measures how realiable a signal or information in the system is. Eventhough physical interpretation of fidelity is yet to be discovered, it can be understood by the physical point of view after defining *infidelity* as

$$infidelity = 1 - fidelity \tag{1}$$

By this definition, infidelity is the probability of the transition occuring between desired and actual states. So, experimental physicist are overcoming this infidelity in the pathway of reliable universal quantum computers. To doing so, one needs to measure the fidelity for benchmarking but precise methods like *quantum state tomography* or *quantum process tomography* are very energy intensive and this energy requirement is growing with the number of qubits in the system. Instead of measuring fidelity by tomography, it is also possible to estimate its value in some interval. By this way, we both save energy and time since estimation methods use less quantum operation and those operations are usually require less time to operate. In this paper, we will discuss three quantum estimation methods all of which are faster and cheaper than tomograpies. Note that only quantum state fidelity is considered in this paper, not quantum process fidelity.

2 Direct Fidelity Estimation Method

Direct fidelity estimation (DFE) method allows us to predict fidelity between desired state ρ and arbitrary state σ up to a constant additive error. For this method, we use randomly selected Pauli operators which prevents that the number of operations growing with the size of the system.

To make estimation, we need to have two assumptions: ρ is pure and we can measure n-qubit Pauli observables (n-fold tensor products of Pauli operators $\mathbb{1}$, X, Y and Z). Since all we need to apply only Pauli operators which are single qubit meaning we don't need 2-qubit gates or entanglement measurements.

We use *square fidelity* for this method;

$$F(\rho,\sigma) = (tr(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}))^2 = tr(\rho\sigma) \tag{2}$$

Notice that we can write this trace in terms of Pauli expectation values of ideal state ρ and actual state σ. Let W_k denote all possible Pauli operators and define characteristic function;

$$\chi_\rho = tr(\frac{\rho W_k}{\sqrt{d}}) \tag{3}$$

where d is the dimension of the Hilbert space such that $d = 2^n$. Note that

$$tr(\rho\sigma) = \sum_k \chi_\rho(k)\chi_\sigma(k) \tag{4}$$

Here, what we are looking for is an estimation for this trace.

2.1 DFE algorithm

To estimate $tr(\rho,\sigma)$, method is done by following steps;

(i) At first, we select a random number k such that $k \in \{1, ..., d^2\}$ with probability

$$\mathcal{P}(k) = (\chi_\rho(k))^2 \tag{5}$$

Note that, here we choose k due to an importance weighting rule that is we choose Pauli operators in such a way that they detect deviation most probably. Although we need infinite numbers of copies of σ in order to measure $\chi_\sigma(k)$ perfectly, but for now, assume we can perfectly measure it. Then estimator is

$$\Omega = \frac{\chi_\sigma(k)}{\chi_\rho(k)} \tag{6}$$

notice that

$$\langle\Omega\rangle_k = tr(\rho\sigma) \tag{7}$$

(ii) To estimate $tr(\rho\sigma)$ with some fixed additive error ϵ and fail probability δ, we iterate this estimator for

$$\ell = \frac{1}{\epsilon^2 \delta} \tag{8}$$

times with independently chosen $k_1, ..., k_\ell$ which give us independent estimates $\Omega_1, ..., \Omega_\ell$

(iii) Now let *infinite-precision* estimator be:

$$\Lambda = \frac{1}{\ell} \sum_{i=1}^{\ell} \Omega_i \tag{9}$$

By Cherbyshev's inequality we have that

$$\mathcal{P}(|\Lambda - tr(\rho\sigma)| \geq \epsilon) \leq \delta \tag{10}$$

(iv) At last, we estimate the infinite-precision estimator using estimator $\tilde{\Lambda}$. Note that $\tilde{\Lambda}$ uses finite copies of σ.

For a given choise of $k_1, ..., k_\ell$ and for each $i = 1, ..., \ell$, we use m_i copies of σ such that

$$m_i(k_i) = \frac{2\log(2/d)}{(\chi_\rho(k_i))^2 \ell \epsilon^2 d} \tag{11}$$

Then we measure Pauli observable W_k on each of these copies of σ and get measurement outcomes $A_{ij} \in \{1, -1\} \forall j = 1, .., m_i$. Also note that

$$\langle A_{ij} \rangle_k = \chi_\sigma(k_i)\sqrt{d} \tag{12}$$

Finally letting

$$\tilde{\Omega}_i = \frac{1}{m_i \sqrt{d} \chi_\rho(k_i)} \sum_{j=1}^{m_i} A_{ij} \tag{13}$$

gives estimation for fidelity as

$$\tilde{\Lambda} = \frac{1}{\ell} \sum_{i=1}^{\ell} \tilde{\Omega}_i \tag{14}$$

with error probability by Hoeffding's inequality:

$$\mathcal{P}(\left|\tilde{\Lambda} - \Lambda\right| \geq \epsilon) \leq \delta \tag{15}$$

Therefore we conclude that $F(\rho, \sigma)$ lies in between $[\tilde{\Lambda} - 2\epsilon, \tilde{\Lambda} + 2\epsilon]$ with a probability $\geq 1 - 2\delta$.

This method uses $\ell = 1/(\epsilon^2\delta)$ Pauli observables independent of the size of the system and m copies of σ where

$$m = \sum_{i=1}^{\ell} m_i \tag{16}$$

Notice that m depends on the random choices k_i but we have,

$$\langle m_i \rangle_k = \sum_{k_i} (\chi_\rho(k_i))^2 m_i \leq 1 + \frac{2d}{\ell\epsilon^2} \log(2/\delta) \tag{17}$$

$$\therefore \langle m \rangle_k \leq 1 + \frac{1}{\epsilon^2\delta} + \frac{2d}{\epsilon^2} \log(2/\delta) \tag{18}$$

Notice that required number of copies is in the order of d.

So by Markov's inequality we have;

$$\mathcal{P}(m \geq t \langle m \rangle) \leq \frac{1}{t} \quad \forall t \geq 1 \tag{19}$$

meaning that the probability of m exceeding $t \langle m \rangle$ is so low, offering reliable number of required actual state σ copies.

2.2 implementation

In the implementation, I applied the method on IBMQ-Belem processor with qiskit. For theoretical part, or ideal ρ, QuTiP library was pretty useful. Note that due to limitation on the job requests in IBMQ, constant error ϵ and error probability δ are so big. During implementations, I realized that maximum number of experiments we can carry was 24. For iteration number lower than this, we can choose $\epsilon = 0.5$ and $\delta = 0.5$. (since $m = 2$, $\ell = 8$ makes the maximum possible iteration number) The full code is in the final_cikaya.py file.

Note that due to equation.6 we need to implement this part so that the $\chi_\rho \neq 0$ since it is in the denominator. And this loop causes a shrink in the Pauli operator set that we can use to $\{\mathbb{1}, Z\}$.

```
while len(omega_tilde) < l:
    chi_rho_k = 0
    while chi_rho_k == 0:
        rand=np.random.randint(0,4,size=n)
        #construct W_k
        Wk = tensor(pauliset[rand[0]],pauliset[rand[1]],
                pauliset[rand[2]],pauliset[rand[3]],pauliset[rand[4]])
        chi_rho = Wk*rho/math.sqrt(d)
        chi_rho_k = chi_rho.tr()
```

After this discovery, note that during studying this method, i was using Ξ instead of Λ but in the presentation, i had to change it in order to prevent disgust that most people experiences after encountring with it, so in the codes, there is xhi instead of lambda.

After running the code on IBMQ-Belem, i get fidelity interval as $[0.23, 1]$ which is not a good estimation thanks to permission of limited experiments. In fact, the result was somewhere in between 0.23... and 1.23... but thanks to following part of the code, we don't see 1.23 type of fidelity estimates;

```
Fmax = xhi_tilde + 2*epsilon
if Fmax > 1:
    Fmax = 1
Fmin = xhi_tilde - 2*epsilon
print("fidelity lies between [",Fmin,Fmax,"]")
```

Also note that in calculating the infinite-precision estimator, i had to make an adjustment to normalize the result with dividing by $m\sqrt{d}$ since the loop runs for ℓm times.

```
#infinite-precision estimator
xhi = 0
stop = 0
for i in range(len(omega)):
    xhi = xhi + omega[i]/(l*m*math.sqrt(d))
```

3 Variational Quantum Fidelity Estimation

This method is designed for low ranked states. For this reason, it can be useful in condensed matter physics (marginals of ground states) and data science (covariance matrices). And this method doesn't need to access to the purifications of ρ and σ.

$$F(\rho, \sigma) = tr(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}) \tag{20}$$

Due to this fidelity definition, we want to calculate trace but this operation requires exponential amount of time as the dimensions grow. And since fidelity contains non-integer powers of ρ and σ, we can't construct an exact quantum algorithm that uses probability measurement results on finite number of ρ and σ copies and outputs fidelity. And since the problem of deciding the trace distance is large or small is a *quantum statistical zero-knowledge (QSZK)*-complete and this class contains *bounded-error quantum poly-time (BQP)*, it is also hard to compute for quantum computers.

For *low-rank fidelity estimation*, we use *the truncated fidelity*. For this, consider subnormalized states:

$$\begin{aligned} \rho_m \equiv \rho_m^\rho = \Pi_m^\rho \rho \Pi_m^\rho = \sum_{i=1}^{m} r_i \left|r_i\right\rangle\!\left\langle r_i\right|, \\ \sigma_m^\rho = \Pi_m^\rho \sigma \Pi_m^\rho \end{aligned} \tag{21}$$

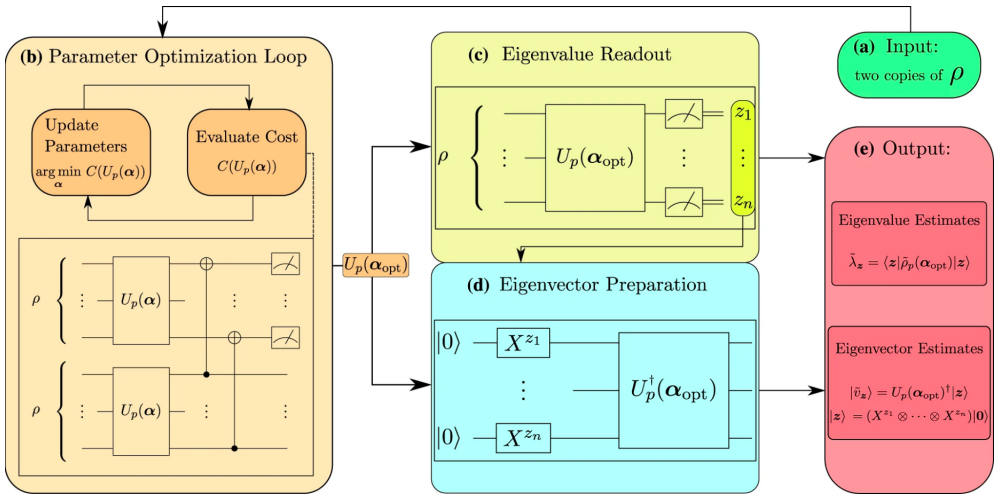

Figure 1: schematics of VQSD algorithm [5]

where Π_m^ρ is the projection onto the subspace spanned by the m-largest eigenvalues of the eigenvectors of ρ, and the eigenvalues of ρ are in decreasing order ($r_i \geq r_{i+1}$). Then truncated fidelity,

$$F(\rho_m, \sigma_m^\rho) = F(\rho_m, \sigma) = Tr\sqrt{\sqrt{\rho_m}\sqrt{\sigma}} \quad (22)$$

and *the generalized truncated fidelity*,

$$\begin{aligned} F_*(\rho_m, \sigma_m^\rho) =& Tr\sqrt{\sqrt{\rho_m}\sqrt{\sigma}} \\ &+ \sqrt{(1 - Tr\rho_m)(1 - Tr\sigma_m^\rho)} \end{aligned} \quad (23)$$

Considering the truncated fidelity as lower and the generalized truncated fidelity as higher bounds, then we have *truncated fidelity bounds (TFB)*, such that

$$F(\rho_m, \sigma_m^\rho) \leq F(\rho, \sigma) \leq F_*(\rho_m, \sigma_m^\rho) \quad (24)$$

TFB gets tighter as m increases and eventually when $m = rank(\rho)$ bounds are equal to the fidelity.

3.1 VQFE algorithm

This algorithm has three steps, approximately diagonalizing ρ with a variational quantum-classical hybrid algorithm, computing matrix elements of σ in the eigenbasis of ρ and classically processing these matrix elements to produce the certified bounds on $F(\rho, \sigma)$.

(i) First, we apply *variational quantum state diagonalization (VQSD)* method.

VQSD is a variational quantum-classical algorithm that uses two copies of ρ, and outputs approximations of the m-largest eigenvalues $\{r_i\}$ and a gate sequence U which prepares the asociated eigenvectors $\{|r_i\rangle\}$. This algorithm contains a quantum-classical optimization loop that minimzes cost function C. (Cost function C measures how $\tilde{\rho} = U\rho U^\dagger$ far from diagonality)

$$C = D_{HS}(\tilde{\rho}, \mathcal{Z}(\tilde{\rho})) \quad (25)$$

where $D_{HS}(\rho, \sigma) = Tr((\rho - \sigma)^2)$ is the Hilbert-Schmidt distance and $\mathcal{Z}$ is the dephasing channel in computational basis.

If we get $C = 0$, we have exact diagonalization. However if we get $C \neq 0$, VQSD algorithm gives

$$\rho' = U^\dagger \mathcal{Z}(\tilde{\rho}) U = \sum_i r_i' \, |r_i'\rangle\langle r_i'| \quad (26)$$

such that $C = D_{HS}(\rho, \rho')$

(ii) Second, we measure the matrix elements $\sigma_{ij}' = \langle r_i'|\sigma|r_j'\rangle$ if $C \neq 0$ if $C \neq 0$.
To do so, we prepare the eigenfunctions in the form $\frac{1}{\sqrt{2}}(|r_i'\rangle + |r_j'\rangle)$ by the eigenvector preparation circuit.

Then we compute the inner products of this superposition state with σ by SWAP test. Note that for fixed m, we need to measure $\frac{1}{2}(m(m+1))$ matrix elements while if we increase m by 1, we only need to measure $2m+1$ new matrix elements.

(iii) At final step, we only need to do classical computation to combine outputs of the previous steps and produce certified bounds on $F(\rho, \sigma)$. What we classically compute is

$$Tr\sqrt{\sqrt{\rho_m}\sqrt{\sigma}} = Tr\sqrt{\sum_{i,j} T_{ij} |r_i\rangle\langle r_j|} \tag{27}$$

where T is $m \times m$ matrix with elements $T_{ij} = \sqrt{r_i r_j} \langle r_i|\sigma|r_j\rangle = \sqrt{r_i r_j}\sigma_{ij} \ni T \geq 0$ Note that T_{ij} can be computed directly from previous steps. To calculate TFB, notice that

$$\begin{aligned} tr(\rho_m) &= \sum_{i=1}^{m} r_i \\ \& \;\; tr(\sigma_m^\rho) &= \sum_{i=1}^{m} \sigma_{ii} \end{aligned} \tag{28}$$

Finally, we compute TFB by

$$F(\rho_m, \sigma_m^\rho) = \sum_i \sqrt{\lambda_i} \tag{29}$$

$$F_*(\rho_m, \sigma_m^\rho) = \sum_i \sqrt{\lambda_i} + \sqrt{(1 - \sum_i r_i)(1 - \sum_i \sigma_{ii}} \tag{30}$$

where λ_i are the eigenvalues of T with $i = 1, ..., m$

4 Direct Fidelity Estimation via Machine Learning

In this method, we solve quantum fidelity estimation problem using classical machine learning techniques. This method transform quantum state fidelity problem into a classification problem. To do that we divide the quantum state space into different subspaces based on their fidelity values. Then we obtain an estimate of the fidelity using a neural network to predict which subspace the quantum state is in.

This method reduces the required measurement settings and the number doesn't increase with the size of the system. As a bonus, this method is applicable to arbitrary quantum states. However, the method is useful when the quantum devices are trusted and the imperfections are caused by noise and device defects. Note that this method gets more accurate as the actual fidelity increases (2) and has an upper limit on the fidelity estimation accuracy.

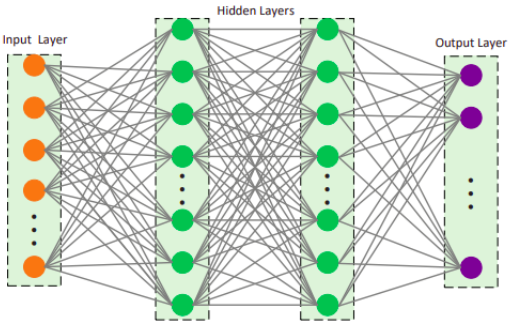

Figure 2: neural network [4]

Like in DFE, let W_k denote all possible Pauli operators. Then we have,

$$tr(\rho\sigma) = \sum_k \frac{1}{d} \chi_\rho(k)\chi_\sigma(k) \tag{31}$$

where characteristic function defined as

$$\chi_\rho(k) = tr(\rho W_k) \tag{32}$$

with the weight function defined for target pure state ρ as

$$\mathcal{P}(k) = (\chi_\rho(k))^2 \tag{33}$$

We choose k measurement settings for n-qubit quantum system which gives 2^n possible outcomes. Using the 2^n outcomes, expectation values of $2^n - 1$ nontrivial Pauli operators can be calculated. Then we select M of these $2^n - 1$ expectation values as neuron inputs. As a result, input layer has a total of $k \times M$ neurons, where $M \leq 2^n - 1$.

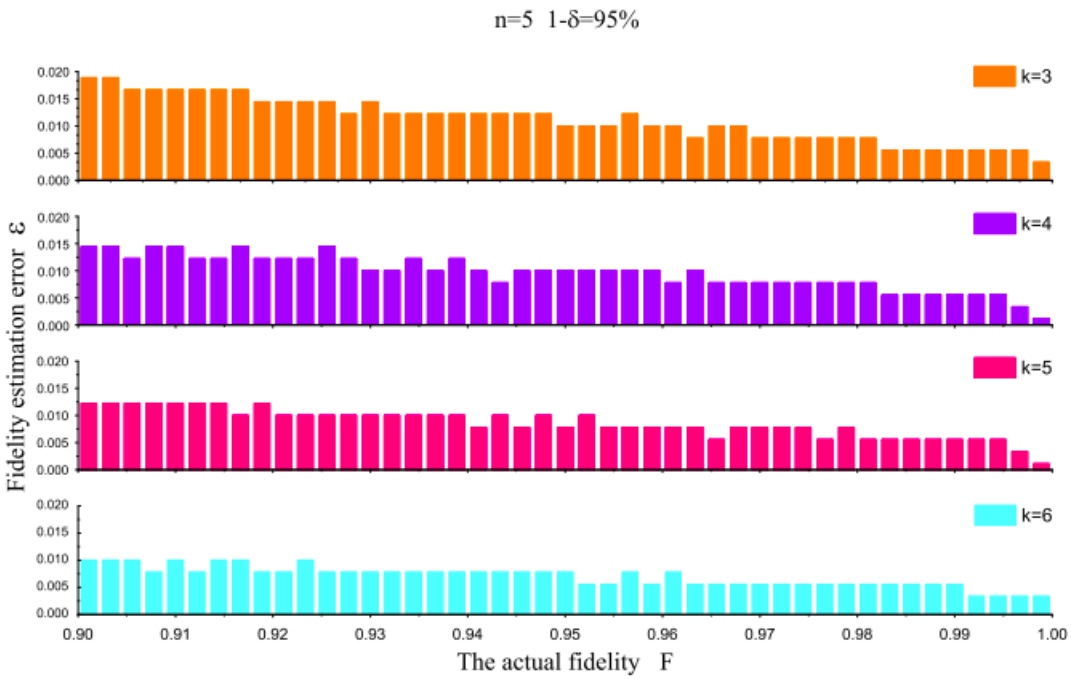

Figure 3: esimation error vs. actual fidelity [4]

4.1 Neural network

To train database of neural network for an arbitrary n-qubit system, we first generate a database for pure state $|0\rangle^{\otimes n}\langle 0|$. Then we divide fidelity of $|0\rangle^{\otimes n}\langle 0|$ into 122 intervals (the numbers are from [4]) and generate 20000 quantum states satisfying randomness and uniformity in each interval. (16000 of them are used for training neural network and 4000 of them are used for validation of neural network) For an arbitrary n-qubit quantum pure state ρ_0, we know that there is a unitary U satisfying

$$\rho_0 = U|0\rangle^{\otimes n}\langle 0|U^\dagger \tag{34}$$

If we apply this U on each quantum state in the original database, we get a new database for σ. Since unitary transform is invariant under inner product, the relation between new database with respect to σ is identical to the original database with respect to $|0\rangle^{\otimes n}\langle 0|$

5 Conclusion

Compared to QST, all of those methods are both cheaper and faster but less accurate. In DFE, algorithm is faster than tomography by a factor of d but compared to DFE via machine learning, DFE requires more measurement settings. Using less measurement settings and being even faster makes DFE via machine learning more suitable but in the meantime it has an upper bound for estimation accuracy and can be used only when the fidelity is high and for small range of error sources. On the other hand, VQFE technique is suitable since it doesn't require any access to the purifications of ρ and σ, in the meantime due to TFB, it can be used for low-ranked states only. Therefore fidelity estimation techniques are still suitable for specific situations, overcome eachother for different expectations and there are still so much work to do.

As you may realized so far, I could only implement DFE algorithm in IBMQ and other two remained unfinished. In addition, in the proposal paper, it was stated that the methods will be compared to eachother and publicly available processors will be compared to eachother but so many things happened in the meantime. Maybe it was a bad idea to implement 6 codes.

References

[1] B. Schumacher, *Quantum coding*, 1995, Phys. Rev. A 51, 2738

[2] Flammia, Steven T. and Liu, Yi-Kai, *Direct Fidelity Estimation from Few Pauli Measurements*, 2011. Physical Review Letters, 106 (23). Art. No. 230501. ISSN 0031-9007. doi:10.1103/PhysRevLett.106.230501

[3] Cerezo, M., et.al. *Variational Quantum Fidelity Estimation*, 2020. DOI:10.22331/q-2020-03-26-248

[4] Zhang, X., Luo, M., et.al., *Direct Fidelity Estimation of Quantum States Using Machine Learning*, 2021. Phys. Rev. Lett. 127, 130503 (2021). DOI: 10.1103/PhysRevLett.127.130503

[5] Ryan LaRose, Arkin Tikku, Étude O'Neel-Judy, *Lukasz Cincio, and Patrick J Coles. Variational quantum state diagonalization*, 2019, npj Quantum Information, DOI: https://doi.org/10.1038/s41534-019-0167-6

[6] Nielsen, M., Chuang, I., *Quantum Computation and Quantum Information*, 2010.

•