Learning memory kernels in Generalized Langevin Equations

Quanjun Lang, Bowen Li, Jianfeng Lu

Contents

1 Introduction

We study the problem of learning in open quantum systems. The goal is to learn the dynamics in the Lindblad quantum master equation (QME)

$$\frac{d}{dt}\rho = -i[H,\rho] + \sum_{k=1}^{r}(C_k\rho C_k^\dagger - \frac{1}{2}C_k^\dagger C_k\rho - \frac{1}{2}C_k^\dagger C_k\rho) \tag{1.1}$$ {eq_QME_main}

from observations of the trajectories. Here H is the system Hamiltonian and C_k's are jump operators. The density matrix $\rho \in \mathbb{R}^{n\times n}$ here characterizes the distribution of a quantum $|\psi(t)\rangle$ in the sense that $\mathbb{E}[|\psi(t)\rangle\langle\psi(t)|]$. Here $|\psi(t)\rangle \in \mathbb{R}^n$ itself satisfies the stochastic Schrödinger equation

$$d|\psi(t)\rangle = \left(-iH|\psi(t)\rangle - \frac{1}{2}\sum_{k=1}^{r}C_k^\dagger C_k|\psi(t)\rangle\right)dt + \sum_{k=1}^{r}C_k|\psi(t)\rangle\, dW_t^k \tag{1.2}$$

where W_t^k's are independent Brownian motions.

The above solution of the QME can be written as

$$\frac{d}{dt}\rho := \mathcal{L}\rho = \mathcal{L}_H\rho + \mathcal{L}_L\rho, \tag{1.3}$$ {eq_QME_Lindb

where $\mathcal{L}_H\rho = -i[H,\rho]$ and $\mathcal{L}_L\rho = \sum_{k=1}^{r}(C_k\rho C_k^\dagger - \frac{1}{2}C_k^\dagger C_k\rho - \frac{1}{2}C_k^\dagger C_k\rho)$. The operator $\mathcal{L}$ is called the Lindbladian superoperators, and solutions of (1.3) can be expressed using semigroup generated by $\mathcal{L}$, namely

$$\rho(t) = e^{\mathcal{L}t}\rho(0). \tag{1.4}$$

Then the map $\mathcal{V}(t) = e^{\mathcal{L}t}$ is a complete positive trace-preserving map for arbitrary time t. By the Choi-Kraus' Theorem, a mapping $\mathcal{V} : \mathcal{B}(\mathbb{R}^n) \to \mathcal{B}(\mathbb{R}^n)$ is completely positive and trace-preserving if and only if it can be expressed as

$$\mathcal{V}\rho = \sum_k C_k^\dagger \rho C_k \tag{1.5}$$

where $\sum_k C_k C_k^\dagger = I_{\mathcal{H}}$.

1.1 Difficulty

Firstly, the evaluation of the density requires taking the expectation with respect to an observable, namely

$$\langle A, \rho \rangle_F = \mathrm{tr}(A\rho) = \langle A \rangle$$

where $A = \sum_i a_i \left|a_i\right\rangle\!\left\langle a_i\right|$. When observe using A, we obtain state $\left|a_i\right\rangle$ with probability a_i. In practice, we take the empirical distribution of the observed states as the estimated value of $\mathrm{tr}(A, \rho)$. Therefore, the accuracy suffers from the law of large numbers with order $1/\sqrt{N}$, where N is the number of independent trials. With given observable A, the original equation (1.1) can be written as

$$\frac{d}{dt}\langle A \rangle(t) = -i\langle [A, H] \rangle(t) + \sum_{k=1}^{r} \left(\left\langle C_k^\dagger A C_k \right\rangle(t) - \frac{1}{2}\left\langle A C_k^\dagger C_k \right\rangle(t) - \frac{1}{2}\left\langle C_k^\dagger C_k A \right\rangle(t) \right) \tag{1.6}$$

Due to the expense of evaluating the states, the derivative term in (1.1) is unrealistic to obtain. Since the finite difference

$$\frac{\langle A \rangle(t + \Delta t) - \langle A \rangle(t)}{\Delta t}$$

will amplify the error in $\langle A \rangle(t)$ and $\langle A \rangle(t + \Delta t)$. The error in the derivative is of order $\Delta t^2 + \frac{\delta}{\Delta t}$, where the first Δt^2 is the error in finite difference using midpoint method, and the δ is the accuracy in the expectation. In order to achieve ε accuracy in the derivative, we need $\Delta t \le \sqrt{\varepsilon}$ and both $\langle A \rangle(t)$ and $\langle A \rangle(t + \Delta t)$ to have $\delta = \varepsilon^{3/2}$ accuracy, therefore requires $1/\varepsilon^{3/2}$ number of independent trails in evaluating the expectation.

1.2 Vectorization and reshaping

1.2.1 Vectorization

Using the fact that

$$\mathrm{vec}(AXB) = (B^\top \otimes A)\mathrm{vec}(X) \tag{1.7}$$

where vec is the vectorization of a matrix in $\mathbb{R}^{n\times n}$ to a vector in $\mathbb{R}^{n^2}$, we can write the vectorized version of the equation (1.1).

$$\mathrm{vec}(\rho)'(t) = \left\{ -i(I \otimes H - H^\top \otimes I) + \sum_{k=1}^{r} \left[\overline{C_k} \otimes C_k - \frac{1}{2} I \otimes (C_k^\dagger C_k) - \frac{1}{2}\overline{(C_k^\dagger C_k)} \otimes I \right] \right\} \mathrm{vec}(\rho)(t) := \mathbf{L}\,\mathrm{vec}(\rho)(t). \tag{1.8}$$

where the super operator is denoted as $\mathbf{L}$. Note that $\mathbf{L}$ has only $2+3r$ Kronecker products in the summation. We can write $\mathbf{L}$ as

$$\mathbf{L} = \sum_{k=1}^{\widetilde{r}} \mathbf{V}_k \otimes \mathbf{U}_k \tag{1.9}$$

where $\mathbf{V}_k$ and $\mathbf{U}_k$ are matrices of size $n \times n$. And the corresponding operator $\mathcal{L}$ is written as

$$\mathcal{L}\rho = \sum_{k=1}^{\widetilde{r}} \mathbf{U}_k \rho \mathbf{V}_k^\top \tag{1.10}$$

Note that $\widetilde{r}$ and r does not have explicit relationships? According to Bowen, there is a one-to-one relationship between $\{H, C_k\}$ and $\{\mathbf{V}_k, \mathbf{U}_k\}$.

1.2.2 Reshaping

We apply the reshaping operator $\mathcal{R}$, defined as in [VLP93]. If matrix $A \in \mathbb{R}^{n^2\times n^2}$, define the rearrangement of A (relative to the blocking parameters n) by

$$A = \begin{bmatrix} A_{11} \dots A_{1n} \\ \vdots \ddots \vdots \\ A_{n1} \dots A_{nn} \end{bmatrix}, \quad \mathcal{R}(A) = \begin{bmatrix} \widetilde{A}_1 \\ \vdots \\ \widetilde{A}_n \end{bmatrix}, \quad \widetilde{A}_j = \begin{bmatrix} \mathrm{vec}(A_{1j})^\top \\ \vdots \\ \mathrm{vec}(A_{nj})^\top \end{bmatrix} \tag{1.11}$$

This rearrangement operator $\mathcal{R}$ gives

$$\mathcal{R}(A \otimes B) = \mathrm{vec}(A)\mathrm{vec}^\top(B), \tag{1.12}$$

Therefore,

$$\mathbf{E} := \mathcal{R}(\mathbf{L}) = \sum_{k=1}^{\widetilde{r}} \mathrm{vec}(\mathbf{V}_k)\mathrm{vec}^\top(\mathbf{U}_k). \tag{1.13}$$

That is to say, the matrix $\mathbf{E}$ has low rank $\widetilde{r}$ compared to n^2.

1.3 Loss function

1.3.1 Trajectory based settings

Assume that we have M independent density trajectory observed on a uniform discrete mesh $\{t_l\}_{l=1}^L$, $t_l = l\Delta t$, namely the following data

$$\textbf{Data:}\{\rho_{t_l}^m\}_{m,l}^{M,L} \tag{1.14}$$

We also assume a Gaussian random noise for each observation point. Our goal is to estimate the Hamiltonian H and the jump operators C_k. By the correspondence between $\{H, C_k\}$ and $\{\mathbf{V}_k, \mathbf{U}_k\}$, it is equivalent to estimate $\mathbf{L}$, and therefore, $\mathcal{R}(\mathbf{L})$ from the data.

We can assume that the initial density $\rho(0)$ follows a distribution μ_0, and then we can define the loss function as

$$\mathcal{E}(\widehat{\mathcal{L}}) = \mathbb{E}_{\mu_0}\left[\int_0^T \left\|\dot{\rho}_t - \widehat{\mathcal{L}}\rho_t\right\|_F^2 dt\right] \tag{1.15}$$

where $\|A\|_F$ represents the Frobenius norm of A. Given the observation data, it is natural to define the loss that approximates the above integral since it represents the negative log-likelihood of the unknown matrix $\mathbf{L}$. Namely, we have

$$\mathcal{E}(\widehat{\mathcal{L}}) = \frac{1}{ML}\sum_{m=1}^{M}\sum_{l=1}^{L}\left\|\dot{\rho}_{t_l}^m - \widehat{\mathcal{L}}\rho_{t_l}^m\right\|_F^2 \tag{1.16}$$

In order to use the rearranged low-rank property, we vectorize the density matrix ρ and decompose the Frobenius norm as

$$\mathcal{E}(\widehat{\mathcal{L}}) = \frac{1}{ML}\sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{i,j=1}^{n}\left|(\dot{\rho}_{t_l}^m)_{i,j} - (\widehat{\mathcal{L}}\rho_{t_l}^m)_{i,j}\right|^2. \tag{1.17}$$

Notice that

$$(\mathbf{U}_k\rho\mathbf{V}_k^\top)_{i,j} = \mathrm{vec}(\mathbf{U}_k)^\top P_{ij}\mathrm{vec}(\mathbf{V}_k) = \langle P_{ij}, \mathrm{vec}(\mathbf{U}_k)\mathrm{vec}(\mathbf{V}_k)^\top\rangle_F \tag{1.18}$$

where $P_{ij} = e_{ij} \otimes \rho$, and e_{ij} is the $n \times n$ matrix that are all zeros except for the (i,j) entry equals to 1, and $\langle\cdot,\cdot\rangle_F$ represents the Frobenius inner product of $\mathbb{R}^{n^2\times n^2}$. Hence

$$(\widehat{\mathcal{L}}\rho_{t_l}^m)_{i,j} = \left(\sum_{k=1}^{\widetilde{r}}\mathbf{U}_k\rho_{t_l}^m\mathbf{V}_k\right)_{i,j} = \sum_{k=1}^{\widetilde{r}}\langle P_{t_l,ij}^m, \mathrm{vec}(\mathbf{U}_k)\mathrm{vec}(\mathbf{V}_k)^\top\rangle_F = \langle P_{t_l,ij}^m, \widehat{\mathbf{E}}\rangle_F \tag{1.19}$$

and $P_{t_l,ij}^m = e_{ij} \otimes \rho_{t_l}^m$. Therefore the loss function can be written with respect to the reshaped unknown matrix $\widehat{\mathbf{E}}$

$$\mathcal{E}(\widehat{\mathcal{L}}) = \mathcal{E}(\widehat{\mathbf{E}}) = \frac{1}{ML}\sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{i,j=1}^{n}\left|(\rho_{t_l}^m)_{ij} - \langle P_{t_l,ij}^m, \widehat{\mathbf{E}}\rangle_F\right|^2. \tag{1.20}$$

1.3.2 Connection to Matrix sensing

In the language of matrix sensing, the matrix $P_{t_l,ij}^m$ can be considered as the measurement matrices, with the corresponding output $(\rho_{t_l}^m)_{ij}$. Now we define the operator $\mathcal{A}: \mathbb{R}^{n^2\times n^2} \to \mathbb{R}^{MLn^2}$ as

$$\mathcal{A}(\widehat{\mathbf{E}}) = \mathrm{vec}\left[(\langle P_{t_l,ij}^m, \widehat{\mathbf{E}}\rangle_F)_{m,l,i,j}\right] \tag{1.21}$$

and the vector $\mathbf{y} \in \mathbb{R}^{MLn^2}$ as $\mathrm{vec}\left[(\rho_{t_l}^m)_{ij}\right]$, we have the loss function written as

$$\mathcal{E}(\widehat{E}) = \left\|\mathcal{A}(\widehat{\mathbf{E}}) - \mathbf{y}\right\|^2 \tag{1.22}$$

up to a constant.

1.4 Matrix sensing

1.4.1 Problem settings

Suppose we observe data from

$$y = \mathcal{A}(M) + z \tag{1.23}$$

where M is an unknown $n_1 \times n_2$ matrix, $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ is a linear mapping, and z is an m-dimensional noise term. The goal is to recover a good approximation of M while requiring as few measurements as possible. Each measurement is interpreted as

$$[\mathcal{A}(M)]_i = \langle A_i, M \rangle_F = y_i \tag{1.24}$$

where A_i is a $n_1 \times n_2$ matrix.

We first introduce the isometry constants of the linear map $\mathcal{A}$.

Definition 1.1. *For each integer $r = 1, 2, \ldots, n$, the isometry constant δ_r of $\mathcal{A}$ is the smallest quantity such that*

$$(1 - \delta_r)\|X\|_F^2 \le \|\mathcal{A}(X)\|^2 \le (1 + \delta_r)\|X\|_F^2 \tag{1.25}$$

holds for all matrices X of rank at most r.

We say that $\mathcal{A}$ satisfies the RIP at rank r if δ_r is bounded by a sufficiently small constant between 0 and 1.

1.4.2 Gaussian Measurement

An essential example shows that the Gaussian measurement has RIP.

Definition 1.2. *$\mathcal{A}$ is a Gaussian measurement operator if each measurement matrix A_i, $1 \le i \le m$, contains i.i.d. $\mathcal{N}(0, 1/m)$ entries, and A_i are independent from each other.*

Theorem 1.1. *[CP, Theorem 2.3] Fix $0 \le \delta < q$ and let $\mathcal{A}$ be a random measurement ensemble obeying the following condition: for any given $X \in \mathbb{R}^{n_2 \times n_2}$ and any fixed $0 < t < 1$,*

$$\mathbb{P}\left(\left| \|A(X)\|^2 - \|X\|_F^2 \right| > t\|X\|_F^2 \right) \le C \exp(-cm) \tag{1.26}$$

for fixed constants $C, c > 0$ (which may depend on t). Then if $m \ge Dnr$, $\mathcal{A}$ satisfies the RIP with isometry constant $\delta_r \le \delta$ with probability exceeding $1 - Ce^{-dm}$ for fixed constants $D, d > 0$.

If $\mathcal{A}$ is a Gaussian random measurement ensemble, $\|\mathcal{A}(X)\|^2$ is distributed as $m^{-1}\|X\|_F^2$ times a chi-squared random variable with m degrees of freedom and from standard concentration inequalities,

$$\mathbb{P}\left(\left| \|A(X)\|^2 - \|X\|_F^2 \right| > t\|X\|_F^2 \right) \le 2\exp\Big(-\frac{m}{2}(t^2/2 - t^3/3) \Big) \tag{1.27}$$

Same result holds if $\mathcal{A}$ is a random projection, or A_i contains sub-Gaussian entries.

1.4.3 Pauli Measurement

Let $M \in \mathbb{C}^{n \times n}$ be an unknown matrix of rank at most r. Let $W_1, \ldots, W_{n^2}$ be an orthonormal basis for $\mathbb{C}^{n \times n}$, with respect to the Frobenius inner product. WE choose m basis elements, $S_1, \ldots, S_m$, i.i.d. uniformly at random from $\{W_1, \ldots, W_{n^2}\}$. We then observe the coefficients $\langle S_i, M \rangle$.

Definition 1.3. *We say the basis $\{W_1, \ldots, W_{n^2}\}$ is **incoherent** if the W_i all have small operator norm,*

$$\|W_i\| \le K/\sqrt{n}. \tag{1.28}$$

where K is a constant.

In the special case that W_i is given by Pauli matrices $P_1 \otimes \cdots \otimes P_n/\sqrt{d}$, they are incoherent with $\|W_i\| \le K/\sqrt{d}$ with $K = 1$.

The linear operator $\mathcal{A}$ is constructed by taking the i-th measurement as S_i.

1.4.4 Using observables

Assume that we have M independent density trajectory, a set of observables $\{A_q\}_{q=1}^{Q}$ so that the density trajectory is observed on a uniform discrete mesh $\{t_l\}_{l=1}^{L}$, $t_l = l\Delta t$, namely the following data

$$\textbf{Data:}\{\langle A_q\rangle^m(t_l)\}_{m,q,l}^{M,Q,L} \tag{1.29}$$

We also assume a Gaussian random noise for each observation point. Our goal is to estimate the Hamiltonian H and the jump operators C_k. By the correspondence between $\{H, C_k\}$ and $\{\mathbf{V}_k, \mathbf{U}_k\}$, it is equivalent to estimate $\mathbf{L}$, and therefore, $\mathcal{R}(\mathbf{L})$ from the data.

It is natural to define the loss function as

$$\mathcal{E}(\widehat{\mathbf{L}}) = \frac{1}{Q}\sum_{q=1}^{Q}\int_0^T \left\|\langle A_q\rangle'(t) - \right\| \tag{1.30}$$

2 Proposed method

We first use the trajectories using the Prony method, then take explicit derivatives and use alternating least squares to minimize the loss.

2.1 Prony fitting of the trajectory

To increase the accuracy of the estimation of the derivatives, we use the Prony method to fit the trajectories $\langle A\rangle$. Since the solution can be written as $\rho(t) = e^{t\mathbf{L}}\rho(0)$, and $\mathbf{L}$ contains the jump operators, it converges to a steady state exponentially. In the case the $\mathbf{L}$ is diagonalizable, $\mathbf{L} = U\Sigma U^{-1}$, where $\Sigma = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{n^2}\}$(possibly repeated). Then

$$\rho(t) = Ue^{t\Sigma}U^{-1}\rho(0) \tag{2.1}$$

2.2 Alternating least squares

3 Literature review

3.1 Quantum Chanel tomography

3.2 Lindbladian learning

3.3 Prony method for system eigenvalues

3.4 Matrix completion

4 Numerical Details

Given a discrete time mesh $t_n = n\Delta t$, generate data $\rho(t_n)$ using the Python package mesolve. Try to learn the quantum channels of $e^{t_n\mathbf{L}}$ using the ALS code, and then compare the eigenvalues of each result.

References

[CP] Emmanuel J Candes and Yaniv Plan. ``Tight Oracle Bounds for Low-rank Matrix Recovery from a Minimal Number of Random Measurements''. In: ().

[VLP93] C. F. Van Loan and N. Pitsianis. ``Approximation with Kronecker Products''. In: *Linear Algebra for Large Scale and Real-Time Applications*. Ed. by Marc S. Moonen, Gene H. Golub, and Bart L. R. De Moor. Dordrecht: Springer Netherlands, 1993, pp. 293–314. (Visited on 03/20/2024).