Master's thesis: Numerical comparison of MCMC methods for Quantum tomography

Danila Mokeev
Supervisors: Estelle Massart, Andrew Thompson and Tameem Adel
21st of June 2024

Ecole Polytechnique de Louvain

Scope of this thesis

Topic: Markov chain Monte Carlo (MCMC) methods in Quantum tomography

Research questions:

1. How do these methods perform in different experimental setups?
2. Why do some methods perform better than others?

Purpose:

- Enable new directions of research
- Help researchers make an informed choice for their use case

Thesis contributions

1. Numerically compare 2 MCMC algorithms, the prob-estimator and the Projected Langevin algorithm
2. Propose 2 new algorithms to understand the impact of the prior and the algorithm on the accuracy

Table of Contents

Table of Contents

Motivation behind Quantum tomography

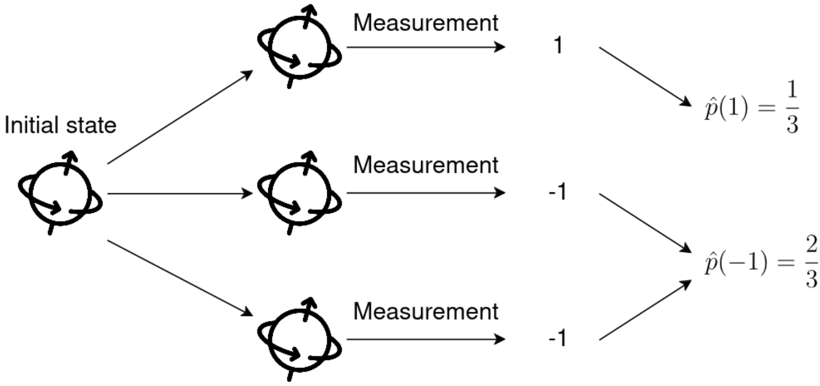
Quantum tomography is a process to reconstruct the quantum state of a system.

There are some challenges to consider:

- Quantum systems are inherently probabilistic
- A measurement can only be made once
- We can only measure the position or momentum, but not both

Quantum tomography: a diagram

Quantum tomography allows to address the existing challenges



Quantum tomography: mathematical description (1)

The Born rule states that

$$p(m) = \mathrm{tr}(\rho P_m) \tag{1}$$

with

- $p(m)$ the probability of occurence of m
- P_m the projector matrix associated to the eigenvalue m of an *observable* O
- ρ the *density matrix* representing the quantum state

The size of ρ is $2^n \times 2^n$ with n the number of qubits.

Quantum tomography: mathematical description (2)

If we flatten the matrices

$$A = \begin{bmatrix} P_{11} & P_{12} & P_{13} \cdots \\ P_{21} & P_{22} & P_{23} \cdots \\ \vdots & \vdots & \vdots \\ P_{m1} & P_{m2} & P_{m3} \cdots \end{bmatrix} \quad \vec{\rho} = \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \rho_{13} \\ \vdots \end{bmatrix} \tag{2}$$

then we can estimate ρ by solving the resulting system of equations

$$A\vec{\rho} = \hat{p} \tag{3}$$

Most common methods

- Direct methods:

$$\hat{\rho} = (A^T A)^{-1} A^T \hat{p} \tag{4}$$

- Optimization-based methods:

$$\hat{\rho} = \mathsf{argmin}_{\vec{\rho}} ||A\vec{\rho} - \hat{p}|| \tag{5}$$

- Pauli basis expansion:

$$\hat{\rho} = \sum_{b \in \{I,x,y,z\}^n} \rho_b \sigma_b \tag{6}$$

- Bayesian methods, and in particular MCMC methods

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \rho_i \quad \text{with } \rho_i \sim \pi(\rho|\mathbf{D}) \tag{7}$$

Existing methods: our focus in this thesis

- Direct methods:

$$\hat{\rho} = (A^T A)^{-1} A^T \hat{p} \tag{8}$$

- Optimization-based methods:

$$\hat{\rho} = \text{argmin}_{\vec{\rho}} ||A\vec{\rho} - \hat{p}|| \tag{9}$$

- Pauli basis expansion:

$$\hat{\rho} = \sum_{b \in \{I,x,y,z\}^n} \rho_b \sigma_b \tag{10}$$

- Bayesian methods, and in particular MCMC methods

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \rho_i \quad \text{with } \rho_i \sim \pi(\rho|\mathbf{D}) \tag{11}$$

Table of Contents

Bayesian framework

In the Bayesian framework:

$$\underbrace{\pi(\rho|\mathbf{D})}_{\text{Posterior}} \propto \underbrace{\pi(\mathbf{D}|\rho)}_{\text{Likelihood}} \underbrace{\pi(\rho)}_{\text{Prior}} \tag{12}$$

Recall that each term is a distribution!

In the context of Quantum tomography:

- Likelihood $\pi(\mathbf{D}|\rho) = \exp(-||A\vec{\rho} - \hat{p}||)$
- Prior $\pi(\rho)$ is method specific

Markov chain Monte Carlo methods

- Markov chain Monte Carlo (MCMC) methods *sample* from $\pi(\rho|\mathbf{D})$.
- They build a Markov chain of samples $\rho_1, \rho_2, \ldots$ such that

$$f(x) = \pi(\rho|\mathbf{D}) \tag{13}$$

  with the equilibrium distribution $f(x)$ of the chain
- The density matrix is then calculated as

$$\tilde{\rho} = \mathbb{E}[\rho] = \int \rho\pi(\rho|\mathbf{D})d\rho \tag{14}$$

$$\Leftrightarrow \hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} \rho_i \quad \text{with } \rho_i \sim \pi(\rho|\mathbf{D}) \tag{15}$$

An example: Metropolis-Hastings algorithm

mcmc.gif

Advantages of MCMC algorithms

Why are we interested in MCMC methods?

- Prior $\pi(\rho)$: additional information about the density matrix - low-rank for example
- Uncertainty quantification: working with distributions instead of point estimates

Table of Contents

Prob-estimator: prior

- Sum of rank-1 matrices:

$$\rho = \sum_{i=1}^{d} \gamma_i V_i {V_i}^{\dagger}$$

- The prior $\pi_1(\gamma_1 \ldots \gamma_d)$ is a Dirichlet distribution. A typical draw leads to a sparse vector.

$$\gamma = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}$$

- The prior $\pi_2(V_i)$ is a unit sphere distribution

$$||V_i|| = 1$$

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

```
// Iterate over each dimension i
```

for $i \leftarrow 1 : d$ **do**

1. Sample γ_i^* from $\pi_1(\gamma_i)$
2. Update $\gamma^{(t)}$ with accept/reject step

end

for $i \leftarrow 1 : d$ **do**

1. Sample V_i^* from $\pi_2(V_i)$
2. Update $V^{(t)}$ with an accept/reject step

end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

    // Iterate over each dimension i

    for $i \leftarrow 1 : d$ **do**

        1. Sample γ_i^* from $\pi_1(\gamma_i)$
        2. Update $\gamma^{(t)}$ with accept/reject step

    end

    for $i \leftarrow 1 : d$ **do**

        1. Sample V_i^* from $\pi_2(V_i)$
        2. Update $V^{(t)}$ with an accept/reject step

    end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

    `// Iterate over each dimension` i

    for $i \leftarrow 1 : d$ **do**

        1. Sample γ_i^* from $\pi_1(\gamma_i)$

        2. Update $\gamma^{(t)}$ with accept/reject step

    end

    for $i \leftarrow 1 : d$ **do**

        1. Sample V_i^* from $\pi_2(V_i)$

        2. Update $V^{(t)}$ with an accept/reject step

    end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

```
// Iterate over each dimension i
```

for $i \leftarrow 1 : d$ **do**

1. Sample γ_i^* from $\pi_1(\gamma_i)$
2. Update $\gamma^{(t)}$ with accept/reject step

end

for $i \leftarrow 1 : d$ **do**

1. Sample V_i^* from $\pi_2(V_i)$
2. Update $V^{(t)}$ with an accept/reject step

end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

```
// Iterate over each dimension i
```

for $i \leftarrow 1 : d$ **do**

1. Sample γ_i^* from $\pi_1(\gamma_i)$
2. Update $\gamma^{(t)}$ with accept/reject step

end

for $i \leftarrow 1 : d$ **do**

1. Sample V_i^* from $\pi_2(V_i)$
2. Update $V^{(t)}$ with an accept/reject step

end

end

Projected Langevin: prior

- Burer-Monteiro factorization: $\rho = YY^\dagger$, with $\mathrm{rank}(Y) = r$
- Low-rank prior: spectral scaled Student-t distribution

$$\pi(Y) = \prod_{j=1}^{r} (\theta^2 + \underbrace{s_j(Y)^2}_{j\text{th eigenvalue of } Y})^{-(2d+r+2)/2} \tag{16}$$

- Promotes sparsity among the eigenvalues leading to a low rank
- Very similar to the Student-t distribution

Projected Langevin: algorithm

Algorithm: Projected Langevin algorithm

for $t \leftarrow 1 : T$ **do**

1. Sample $\tilde{w}^{(t)} \sim N(\mathbf{0}, \mathbf{I})$
2. $\tilde{Y}^{(t)} \leftarrow \tilde{Y}^{(t-1)} - \eta^{(t)} \underbrace{\nabla \pi(\tilde{Y}^{(t-1)} | \mathbf{D})}_{\textbf{gradient}} + \frac{\sqrt{2\eta^{(t)}}}{\beta} \tilde{w}^{(t)}$

end

The gradient allows us to explore the regions of high density faster.

Projected Langevin: algorithm

Algorithm: Projected Langevin algorithm

for $t \leftarrow 1 : T$ **do**

1. Sample $\tilde{w}^{(t)} \sim N(\mathbf{0}, \mathbf{I})$
2. $\tilde{Y}^{(t)} \leftarrow \tilde{Y}^{(t-1)} - \eta^{(t)} \underbrace{\nabla \pi(\tilde{Y}^{(t-1)} | \mathbf{D})}_{\textbf{gradient}} + \frac{\sqrt{2\eta^{(t)}}}{\beta} \tilde{w}^{(t)}$

end

The gradient allows us to explore the regions of high density faster.

Projected Langevin: algorithm

Algorithm: Projected Langevin algorithm

for $t \leftarrow 1 : T$ **do**

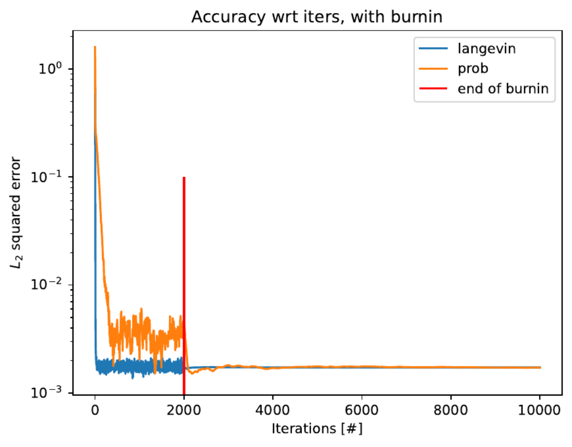
1. Sample $\tilde{w}^{(t)} \sim N(\mathbf{0}, \mathbf{I})$
2. $\tilde{Y}^{(t)} \leftarrow \tilde{Y}^{(t-1)} - \eta^{(t)} \underbrace{\nabla \pi(\tilde{Y}^{(t-1)} | \mathbf{D})}_{\textbf{gradient}} + \frac{\sqrt{2\eta^{(t)}}}{\beta} \tilde{w}^{(t)}$

end

The gradient allows us to explore the regions of high density faster.

Table of Contents

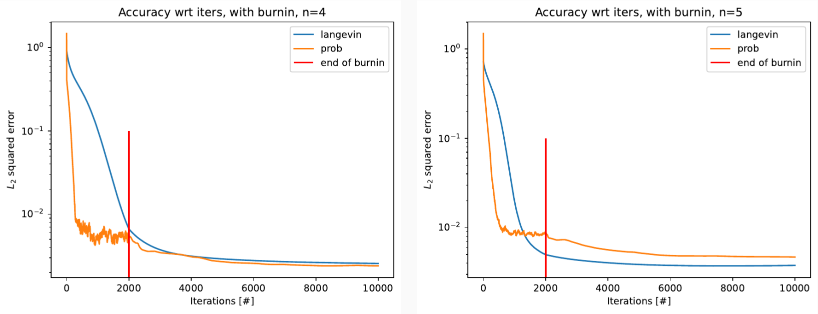
Convergence plot



$\Longrightarrow$ Projected Langevin converges faster for $n = 3$ qubits

Convergence speed for $n = 4, 5$

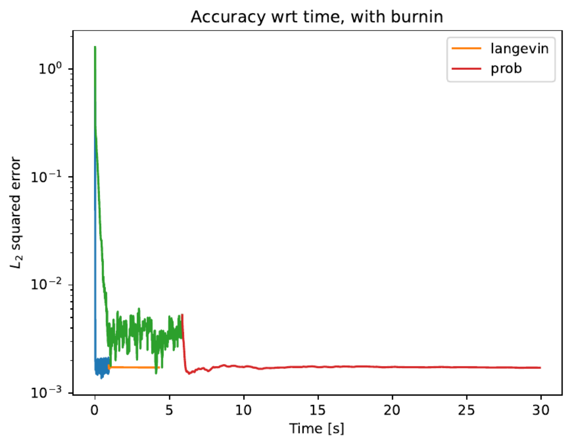
Reminder: a larger n means a larger density matrix



(a) $n = 4$ **(b)** $n = 5$

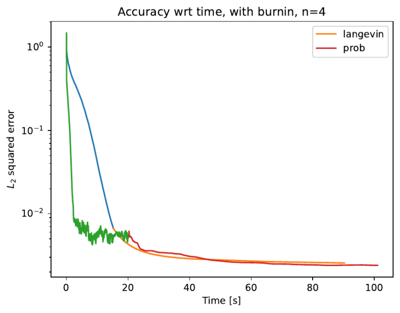
$\Longrightarrow$ Projected Langevin converges slower than previously

Computation time for $n = 3$

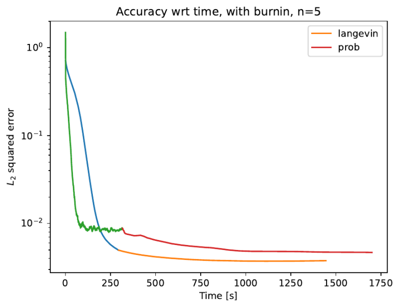


$\Longrightarrow$ For $n = 3$, Projected Langevin takes much less time

Computation time for $n = 4, 5$



(a) $n = 4$



(b) $n = 5$

$\Longrightarrow$ When n increases, Projected Langevin becomes as slow as the prob-estimator due to the gradient cost.

Introducing 2 new methods

What makes Projected Langevin perform better ?

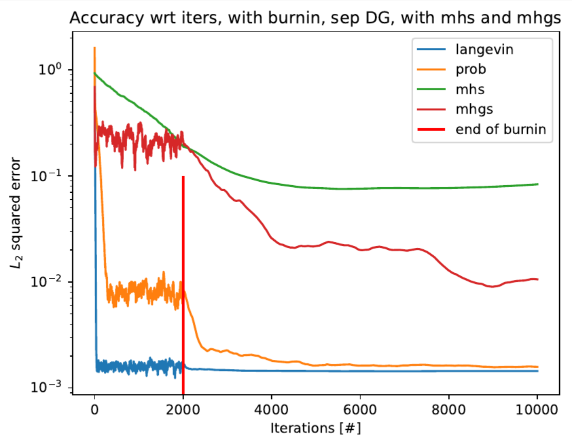
To answer this question, we introduce 2 new algorithms:

1. Metropolis-Hastings with Student-t prior (MHS)
2. Metropolis-Hastings with Gibbs with Student-t prior (MHGS)

They combine:

- The algorithm from the prob-estimator
- The prior from the Projected Langevin algorithm

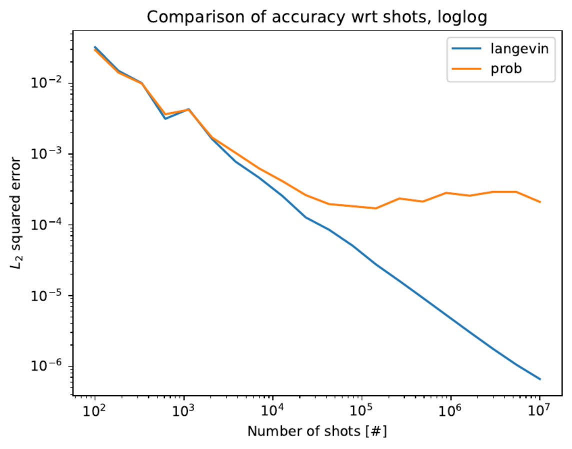
Convergence comparison



$\Longrightarrow$ The prior itself is not a solution, and must be paired with the right algorithm to be fast and accurate

Impact of the number of shots

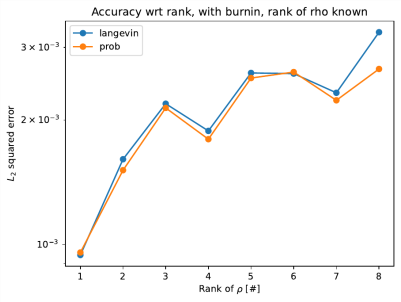
Shot: measurement we perform on a clone of the state



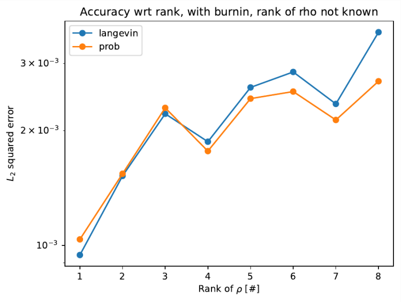
$\Longrightarrow$ The prob-estimator does not scale!

Impact of knowing the rank of ρ

Reminder: for Projected Langevin, $\rho = YY^\dagger$, with $\text{rank}(Y) = r$



(a) Rank of ρ known



(b) Rank of ρ not known

$\Longrightarrow$ The information about the rank only marginally affects the accuracy

Summary

- Quantum tomography is not yet a solved problem, especially for large systems
- MCMC methods are a promising direction of research, thanks to uncertainty quantification and prior information
- The choice of the algorithm might have more impact on the convergence speed and accuracy than the prior

Future work

- Extend tests to larger n to check for robustness of conclusions
- Try other gradient-based algorithms to see the impact on convergence speed (for example HMC)
- Try other priors to see the impact on computation time (the gradient may be faster to compute) and accuracy

References

Prob-estimator: full

It can be seen as eigendecomposition, without the orthogonality property:

$$\rho = U \Lambda U^{\dagger} \tag{17}$$

Prior:

$$\pi(\rho) = \pi_1(\gamma_1, \ldots, \gamma_d) \prod_{i=1}^{d} \pi_{2,i}(V_i) \tag{18}$$

Likelihood:

$$\pi(\mathbf{D}|\rho) = \pi(\rho, \mathbf{D}) = \exp(-\lambda \ell(\rho, \mathbf{D})) \tag{19}$$

with:

$$\ell(\rho, \mathbf{D}) = \sum_{\mathbf{a} \in \mathcal{E}^n} \sum_{\mathbf{s} \in \mathcal{R}^n} \left[\mathrm{tr}(\rho P_{\mathbf{s}}^{\mathbf{a}}) - \hat{p}_{\mathbf{a},\mathbf{s}} \right]^2 \tag{20}$$

Posterior:

$$\pi(\nu|\mathbf{D}) \propto \exp(-\lambda \ell(\nu, \mathbf{D})) \pi(\nu) \tag{21}$$

Projected Langevin: full

Prior:

$$\nu_\theta(Y) = C_\theta \det(\theta^2 I_d + YY^\dagger)^{-(2d+r+2)/2} \tag{22}$$

Likelihood:

$$L(Y, \mathbf{D}) = \sum_{i=1}^{M} (\hat{p}_m - \mathrm{tr}(A_m Y Y^\dagger))^2 \tag{23}$$

Posterior:

$$\hat{\nu}_{\lambda,\theta}(Y, \mathbf{D}) = \exp(-f_{\lambda,\theta}(Y, \mathbf{D})) \tag{24}$$

with

$$f_{\lambda,\theta}(Y, \mathbf{D}) = \lambda \sum_{i=1}^{M} (\hat{p}_m - \mathrm{tr}(A_m Y Y^\dagger))^2 + \frac{2d + r + 2}{2} \log\det(\theta^2 I_d + YY^\dagger) \tag{25}$$

Potential future experiments

- Try experiemnts with more qubits to draw more robust conclusions
- Test with other algorithms and priors to see if its a property of this prior in particular, or it generalizes (The calculation of the graidient is going to be more costly in all cases)
- Try to use HMC to see if it still converges as fast for higher dimensions