Statistical Methods for Quantum State Estimation

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Bachelor of Science
in
Mathematics and Physics

Author :	Pascal van der Vaart
Student ID :	s1861972
Supervisor :	dr. S.L. van der Pas, dr. T.E. O'Brien
2^{nd} corrector :	prof. dr. R.D. Gill

Leiden, The Netherlands, August 15, 2019

Statistical Methods for Quantum State Estimation

Pascal van der Vaart

Huygens-Kamerlingh Onnes Laboratory, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

August 15, 2019

Contents

Chapter 1

Introduction

Quantum computing is currently a widely researched topic with many potential applications. A particular application is the variational quantum eigensolver [1], which can solve large eigenvalue problems that are intractable on classical computers. This requires the estimation of a quantum state from data generated by measurements. This process is known as quantum state estimation. There are various statistical methods for this task. There has been success with frequentist methods such as linear inversion [2], and maximum likelihood [3]. Bayesian methods have also been proposed and have desirable properties, such as providing error bars on the estimate [4]. A problem in an experimental setting is that preparing and measuring quantum states does not always go without failure, and in this thesis the problem of state estimation is explored when there is uncertainty in the validity of measurement results. In particular, to explore methods of handling these errors, a methods of moments estimator, a maximum likelihood estimator, and a Bayesian mean estimator are implemented and compared in the cases with errors and without errors.

In Chapter 2 several concepts from quantum mechanics and quantum computing are introduced. Chapter 3 will introduce the two frequentist estimation methods of this thesis, and Chapter 4 discusses the Bayesian method and how the Metropolis-Hastings algorithm can be used to implement the estimation method. In Chapter 5 the problem of experimental errors is introduced, and Chapter 6 contains figures comparing the estimators in the case where experimental errors are present, and the case where they are not. The results are discussed in Chapter 7.

Chapter 2

Preliminary Quantum Mechanics

This chapter will provide some basic concepts from quantum mechanics needed to understand the origin of the problem in this thesis, and explain what quantum state estimation is. Most of these concepts can be found in a textbook on quantum computing. For more information I can recommend chapter 2 of *Quantum Computation and Quantum Information* by Nielsen and Chuang [5]. At the end of this chapter it will be clear that quantum state estimation boils down to estimating multiple Bernoulli parameters, which are linked by constraints arising due to physical properties of quantum states.

2.1 Bra-ket Notation

In quantum mechanics elements in a separable Hilbert space $\mathcal{H}$ are usually denoted using the bra-ket notation. Elements of $\mathcal{H}$ are denoted as "kets" $|\phi\rangle \in \mathcal{H}$ which are seen as column vectors, and the corresponding conjugate transpose of $|\phi\rangle$ is denoted by a "bra" in the dual space: $(|\phi\rangle)^\dagger = \langle\phi| \in \mathcal{H}'$ which is seen as a row vector. The standard inner product can then be written as a matrix multiplication $< |\phi\rangle, |\psi\rangle >= (|\phi\rangle)^\dagger|\psi\rangle = \langle\phi|\psi\rangle \in \mathbb{C}$ and the outer product is denoted by the matrix product $|\phi\rangle\langle\psi|$.

2.2 Quantum States

Quantum states can be represented by certain elements in Hilbert spaces known as state vectors.

Definition 1 *An element $|\phi\rangle \in \mathcal{H}$ is called a* ***state vector*** *if $\langle\phi|\phi\rangle = 1$.*
The set $\mathcal{S}(\mathcal{H})$ is the set of state vectors in $\mathcal{H}$.

In this thesis the quantum states are states of qubits, which are represented by elements of a two dimensional complex Hilbert space, which is isomorphic to $\mathbb{C}^2$. It is common to take $\{|0\rangle, |1\rangle\}$ as an orthonormal basis for a single qubit. This means that states can be written as $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ such that $|\alpha|^2 + |\beta|^2 = 1$. The state of multiple qubits can be represented in the tensor product of their respective Hilbert spaces. A system with n qubits can thus be represented by a space which is isomorphic to $\mathbb{C}^{2n}$.

A problem with state vectors is that they fail to capture classical uncertainty of the state. If a system is in state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ with probability p and in state $|\phi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ with probability p', one may expect to represent the combined state by $\sqrt{p}|\phi\rangle + \sqrt{p'}|\phi'\rangle$. But in general this is not a state, as

$$\begin{aligned}(\sqrt{p}\langle\phi| + \sqrt{p'}\langle\phi'|)(\sqrt{p}|\phi\rangle + \sqrt{p'}|\phi'\rangle) &= \\ p(\alpha^2 + \beta^2) + p'(\alpha'^2 + \beta'^2) + \sqrt{pp'}(\alpha\alpha' + \beta\beta') &= \\ p + p' + \sqrt{pp'}(\alpha\alpha' + \beta\beta') &= \\ 1 + \sqrt{pp'}(\alpha\alpha' + \beta\beta'),\end{aligned}$$

which is in general not equal to 1. The correct way to capture both classical and quantum uncertainty in a state is to use density matrices.

2.3 Density Matrices

Definition 2 *A density matrix is a Hermitian positive semidefinite matrix $\rho \in Mat(\mathbb{C}^{2^n}, \mathbb{C}^{2^n})$ such that $Tr(\rho) = 1$.*

A density matrix ρ represents a pure state if $\rho = |\phi\rangle\langle\phi|$ for some $|\phi\rangle \in \mathcal{S}(\mathcal{H})$. Indeed, in this case the state could just be written as a state vector $|\phi\rangle$. Density matrices can also represent mixed states. This becomes clear after spectral decomposition, $\rho = \sum_i p_i|\phi_i\rangle\langle\phi_i|$, in which the p_i should be interpreted as classical probabilities, i.e., there is a p_i chance that the state is $|\phi_i\rangle$. Density matrices are thus capable of incorporating classical uncertainty of states in a system. The fact that ρ is positive semidefinite and has trace 1 ensures that $p_i \geq 0$ and $\sum_i p_i = 1$. A measure of how pure or mixed a density matrix is is its purity, which is defined to be $\operatorname{Tr}\left(\rho^2\right)$. Pure states have purity 1, since $\operatorname{Tr}\left(|\phi\rangle\langle\phi||\phi\rangle\langle\phi|\right) = \operatorname{Tr}\left(|\phi\rangle\langle\phi|\right) = \langle\phi|\phi\rangle = 1$. For a mixed state of n qubits the purity lies in the interval $[\frac{1}{2^n}, 1)$. It will turn out that the performance of the methods for quantum state estimation is severely impacted by the purity of the to be estimated state.

In conclusion, when considering mixed states, quantum states can be represented by a subset of a space of Hermitian matrices. A particularly useful basis for this space is discussed in the next section.

2.4 Pauli Matrices

Definition 3 *The Pauli matrices are the Hermitian matrices*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

.

They have the following properties:

- $X^2 = Y^2 = Z^2 = \mathbb{I}$
- $XY = iZ$, $YZ = iX$, and $ZX = iY$
- $\operatorname{Tr}(X) = \operatorname{Tr}(Y) = \operatorname{Tr}(Z) = 0$

Lemma 1 *The set $\{\mathbb{I}, X, Y, Z\}$ is a basis for the 2×2 Hermitian matrices over $\mathbb{R}$. Furthermore, $\{\mathbb{I}_2, X, Y, Z\}^{\otimes n}$ is a basis for the $2^n \times 2^n$ Hermitian matrices over $\mathbb{R}$. This basis is known as the Pauli basis.*[*]

Proof 1 *To prove the first part, note that 2×2 Hermitian matrices are of the form*

$$H = \begin{pmatrix} a & b + ci \\ b - ci & d \end{pmatrix},$$

where $a, b, c, d \in \mathbb{R}$, so writing $H = \frac{1}{2}(a+d)\mathbb{I} + bX + cY + \frac{1}{2}(a-d)$ shows that $\{\mathbb{I}, X, Y, Z\}$ spans the space of 2×2 Hermitian matrices. Furthermore, solving $a\mathbb{I} + bX + cY + dZ = 0$ results in $b = 0$, $c = 0$, $a + d = 0$, and $a - d = 0$, which has $a, b, c, d = 0$ as its only solution, proving that $\{\mathbb{I}, X, Y, Z\}$ is indeed a basis. The second part follows directly from the definition of the tensor and Kronecker product.

[*] Usually the dimension of $\mathbb{I}$ is implicit, but I have explicitly written $\mathbb{I}_2$ to point out that this is the 2×2 identity matrix that appears in the tensor product.

Since the Pauli matrices have trace 0, a result of Lemma 1 is that every 2×2 density matrix can be written as $\frac{1}{2}(\mathbb{I} + c_x X + c_y Y + c_z Z)$. The factor $\frac{1}{2}$ comes from that fact that Tr $(\mathbb{I}) = 2$ and Tr $(\rho) = 1$. In this 2×2 case a density matrix can be represented by the three real scalars c_x, c_y, c_z. This easily generalizes to the case of larger systems, in which a $2^n \times 2^n$ density matrix can be written as $\rho = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1} c_i B_i)$, where $B_i \in \{\mathbb{I}_2, X, Y, Z\}^{\otimes n} \setminus \{\mathbb{I}\}$. The numbering of these B_i does not really matter, and for the sake of this thesis should not be worried about.

The normalizing factor $\frac{1}{2^n}$ is used for the following nice property.

Lemma 2 *Any density matrix* $\rho = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1} c_i B_i)$ *is such that* $c_i \in [-1, 1]$ *for all* $i \in \{1, \dots, 4^n - 1\}$.

Proof 2 *Let* $\rho = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1} c_i B_i)$ *be a density matrix, and let* $j \in \{1, \dots, 4^n - 1\}$. *Note that*

$$Tr\left(\rho B_j\right) = Tr\left(\frac{1}{2^n}(B_j + \sum_{i=1}^{4^n-1} c_i B_i B_j)\right) = \frac{1}{2^n} Tr\left(c_j B_j^2\right) = c_j,$$

since for all $i \in \{1, \dots, 4^n - 1\} \setminus \{j\}$ *it holds that* $Tr\left(B_i B_j\right) = 0$, *and* $Tr\left(B_j\right) = 0$.

Let p_i *and* $|\phi_i\rangle$ *be the eigenvalues and corresponding eigenvectors of* ρ, *so that* $\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|$. *Writing* ρ *this way results in*

$$\begin{aligned} |c_j| =& |Tr\left(\rho B_j\right)| = |\sum_i p_i Tr\left(|\phi_i\rangle\langle\phi_i|B_j\right)| = \\ & |\sum_i p_i \langle\phi_i|B_j|\phi_i\rangle| \leqslant \sum_i p_i |\langle\phi_i|B_j|\phi_i\rangle| \leqslant \sum_i p_i = 1, \end{aligned}$$

where in the last step the Cauchy-Schwarz inequality is used to bound

$$|\langle\phi_i|B_j|\phi_i\rangle| \leqslant \sqrt{\langle\phi_i|\phi_i\rangle\langle\phi_i|B_j^2|\phi_i\rangle} = 1.$$

2.5 Measurements

Definition 4 *Let* I *be a non empty set. A positive-operator valued measure is a set* $\{P_i \mid i \in I\} \subseteq Mat(\mathbb{C}^{2n}, \mathbb{C}^{2n})$ *such that*

- $\sum_{i \in I} P_i = \mathbb{I}$
- $P_i \geqslant 0$ *for all* $i \in I$

These will be called measurements for short. Applying a measurement $\{P_i \mid i \in I\}$ on a quantum state ρ results in an outcome $i \in I$ with probability $\mathrm{Tr}\,(\rho P_i)$. One can check that indeed $\sum_{i \in I} \mathrm{Tr}\,(\rho P_i) = \mathrm{Tr}\,(\rho \sum_{i \in I} P_i) = \mathrm{Tr}\,(\rho) = 1$. And $\mathrm{Tr}\,(\rho P_i) \geqslant 0$. The state also collapses to a specific state depending on the outcome according to the Born rule. In this thesis only the outcome and the fact that the state collapses to something other than ρ are important, because the outcomes of the measurements will be used to estimate the quantum state, and the fact that the state collapses means that the to be estimated state ρ has to be prepared again for each measurement.

In this thesis, only a small subset of measurements will be used. The set I will always be $\{0, 1\}$, meaning that each measurement only has two possible outcomes. The measurements will be based on the Pauli matrices. Other measurements do exist and can be done experimentally, but it is unclear whether doing these provides a real advantage over the simpler ones presented here.

Lemma 3 *For each $B \in \{I, X, Y, Z\}^{\otimes n}$ the set $\{B_0, B_1\} = \{\frac{1}{2}(\mathbb{I} - B), \frac{1}{2}(\mathbb{I} + B)\}$ is a measurement.*

Proof 3 *Clearly, $B_0 + B_1 = \mathbb{I}$. Furthermore, $\frac{1}{2}(\mathbb{I} \pm B)\frac{1}{2}(\mathbb{I} \pm B) = \frac{1}{4}(\mathbb{I} \pm 2B + B^2) = \frac{1}{2}(\mathbb{I} \pm B)$, so B_0 and B_1 are projections and hence positive semidefinite.*

This is the measurement that is meant when the phrase "measuring B" is used in this thesis.

The following measurements can be done on a single qubit[†]:

$$\begin{aligned}
\{X_0, X_1\} &= \{\frac{1}{2}(\mathbb{I} - X),\ \frac{1}{2}(\mathbb{I} + X)\} \\
\{Y_0, Y_1\} &= \{\frac{1}{2}(\mathbb{I} - Y),\ \frac{1}{2}(\mathbb{I} + Y)\} \\
\{Z_0, Z_1\} &= \{\frac{1}{2}(\mathbb{I} - Z),\ \frac{1}{2}(\mathbb{I} + Z)\}
\end{aligned}$$

As an example, consider the state $\rho = \frac{1}{2}(\mathbb{I} + c_x X + c_y Y + c_z Z)$, where $c_x, c_y, c_z \in [-1, 1]$ according to Lemma 2. It will be clear from this example that from a mathematical point of view, ρ defines probability distributions

[†] Technically a measurement of $\mathbb{I}$ could also be considered, but there is no point since it always results in 1.

over the outcomes of the above measurements. Doing a measurement results in a sample from its respective distribution defined by ρ. The probabilities of finding the possible outcomes can be found by a direct calculation. By definition, the probability of finding 1 when measuring X is given by

$$\begin{aligned}
\mathrm{Tr}\,(\rho X_1) &= \mathrm{Tr}\left(\frac{1}{2}\rho(\mathbb{I}+X)\right)\\
&= \mathrm{Tr}\left(\frac{1}{2}\rho+\frac{1}{4}\mathbb{I}X+\frac{1}{4}c_xXX+\frac{1}{4}c_yYX+\frac{1}{4}c_zZX\right)\\
&= \mathrm{Tr}\left(\frac{1}{2}\rho\right)+\mathrm{Tr}\left(\frac{1}{4}\mathbb{I}X+\frac{1}{4}c_xXX+\frac{1}{4}c_yYX+\frac{1}{4}c_zZX\right)\\
&= \mathrm{Tr}\left(\frac{1}{2}\rho\right)+\mathrm{Tr}\left(\frac{1}{4}c_x\mathbb{I}\right)=\frac{1}{2}+\frac{1}{2}c_x,
\end{aligned}$$

where the mentioned properties of the Pauli matrices and the fact that they have trace 0 is used to obtain the last line.

Evidently when calculating the probability to find 0 when measuring X the probability $\mathrm{Tr}\,(\rho X_0)=\frac{1}{2}-\frac{1}{2}c_x$ is found. These probabilities can analogously be calculated for the Y and Z measurements which will depend on their respective coefficients c_y and c_z. These are valid probabilities since $c_x, c_y, c_z \in [-1,1]$. Doing a measurement on X, Y, or Z and only considering the outcome is therefore equivalent to sampling a Bernoulli distribution with parameter $\frac{1}{2}+\frac{1}{2}c_x$, $\frac{1}{2}+\frac{1}{2}c_y$, or $\frac{1}{2}+\frac{1}{2}c_z$ respectively.

2.6 Quantum State Estimation

In quantum state estimation the goal is to reconstruct the density matrix that represents the state of a particular system. The necessity of quantum state estimation comes up in experimental quantum computing, such as variational quantum eigensolvers [1]. The output of a quantum algorithm is contained in the state of the system, which is hidden to the experimentalist. The state has to be estimated by doing measurements and using some estimation method to recover the desired information. This section will show that doing quantum state estimation essentially boils down to estimating Bernoulli parameters of independent samples from multiple Bernoulli distributions, where the Bernoulli parameters are linked to each other by constraints on the parameter space.

It is assumed here that a device is able to consistently produce a state ρ, after which any of the measurements mentioned can be done. After this the state collapses, and the device will create ρ again from scratch so that a new measurement can be done. Since it takes time to prepare a state, it is desirable that estimation methods are efficient so that the state does not have to be prepared too many times for a good estimate.

As seen in last section, the probabilities of finding an outcome 0 or 1 when doing different measurements on ρ depend on the coefficients of ρ in the Pauli basis elements corresponding to the measurements. Information on ρ can therefore be gained by repeating experiments to estimate the coefficients. For example, again consider the state $\rho = \frac{1}{2}(\mathbb{I} + c_x X + c_y Y + c_z Z)$. In the previous paragraph it was calculated that measuring X, Y and Z results in outcome 1 with probability $p_x := \frac{1}{2} + \frac{1}{2}c_x, p_y := \frac{1}{2} + \frac{1}{2}c_y$ and $p_z := \frac{1}{2} + \frac{1}{2}c_z$ respectively. Doing these measurements on the state corresponds to sampling from Bernoulli distributions, where each different measurement X, Y, or Z, has its own parameter p_x, p_y or p_z. Quantum state estimation then seems to be equivalent to estimating Bernoulli parameters, but this is not entirely true. There is some hidden matrix $\rho = \frac{1}{2}(\mathbb{I} + c_x X + c_y Y + c_z Z)$, which corresponds to three Bernoulli distributions with parameters p_x, p_y and p_z. The goal is to estimate the Bernoulli parameters from independent samples from their corresponding distribution so that ρ can be recovered. The problem is that there is no guarantee that for estimates of the Bernoulli parameters $\hat{p}_x, \hat{p}_y$ and $\hat{p}_z$ that the resulting matrix $\hat{\rho} = \frac{1}{2}(\mathbb{I} + (2\hat{p}_x - 1)X + (2\hat{p}_y - 1)Y + (2\hat{p}_z - 1)Z)$, is positive semidefinite, and for larger systems it is very unlikely that this naive approach will lead to a positive semidefinite estimate $\hat{\rho}$. Furthermore, it will be clear later that there is also no "nice" way of writing the positive semidefiniteness restriction in terms of the coefficients in the Pauli basis.

When considering larger systems, if the system has n qubits then ρ is a $2^n \times 2^n$ matrix and can be written as $\frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1} c_i B_i)$, where $c_i \in [-1, 1]$ and $B_i \in \{I, X, Y, Z\}^{\otimes n}$ by Lemma 1 and Lemma 2. Lemma 3 ensures that for each $i \in \{1, \ldots 4^n - 1\}$ there is a measurement which finds outcome 1 with probability $\frac{1}{2} + \frac{1}{2}c_i$ and outcome 0 with probability $\frac{1}{2} - \frac{1}{2}c_i$, since $\mathrm{Tr}\left(\frac{1}{2}\rho(\mathbb{I} + B_i)\right) = \frac{1}{2}\mathrm{Tr}\,(\rho) + \frac{1}{2}\mathrm{Tr}\,(\rho B_i) = \frac{1}{2} + \frac{1}{2}c_i$, where the identity $\mathrm{Tr}\,(\rho B_i) = c_i$ is taken from the proof of Lemma 2. The important observation here is that for an n qubit system, $4^n - 1$ parameters have to be estimated in order to reconstruct the density matrix. The problem of reconstructing ρ for n qubit systems thus essentially seems the same as for the 1 qubit case, apart from the fact that there are $4^n - 1$ measurements and corresponding Bernoulli distributions instead of just 3. The high dimensionality however causes the constraint of positive semidefiniteness to be a lot more complicated. It turns out that for 1 qubit, the coefficients have to lie in a sphere, but for more qubits this shape gets very complicated. In fact, satisfying these constraints, known as the N-representability constraints in quantum chemistry, is proven to be QMA-complete, a quantum analogue to NP-complete [6].

In conclusion, quantum state estimation is the statistical problem in which there is a $2^n \times 2^n$ density matrix ρ, which corresponds to a set of Bernoulli parameters $\{p_i \mid i \in \{1, \ldots, 4^n - 1\}\}$ through the fact that ρ can be written as $\rho = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1}(2p_i - 1)B_i)$, where $B_i \in \{I, X, Y, Z\}^{\otimes n}$. Given independent samples $o_1 \ldots o_{N_i} \sim \text{Ber}(p_i)$ from each Bernoulli distribution, the goal is to create estimates $\hat{p}_i$ such that the reconstructed matrix $\hat{\rho} = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1}(2\hat{p}_i - 1)B_i)$ is a good estimate to ρ, and ideally also positive semidefinite.

Estimates will be evaluated by two metrics, the trace distance between ρ and $\hat{\rho}$, and the mean squared error on the estimates $\hat{p}_i$ of p_i.

Definition 5 *The trace distance between $\hat{\rho}$ and ρ is given by*

$$T(\hat{\rho}, \rho) = \frac{1}{2} Tr\left(\sqrt{(\hat{\rho} - \rho)^\dagger (\hat{\rho} - \rho)}\right).$$

This is a metric also used in other literature [2].

Definition 6 *The mean squared error between the Bernoulli parameters of $\hat{\rho} = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1}(2\hat{p}_i - 1)B_i)$ and $\rho = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1}(2p_i - 1)B_i)$ is given by*

$$MSE(\hat{\rho}, \rho) = \frac{1}{4^n - 1} \sum_{i=1}^{4^n-1} (\hat{p}_i - p_i)^2.$$

This metric provides information on how well a method can estimate the individual parameters.

The following two chapters will present several methods for quantum state estimation.

Chapter 3

Frequentist Methods for Quantum State Estimation

This chapter will discuss two frequentist methods of quantum state estimation. For the first method there will be some simulation experiments to show certain shortcomings in order to motivate the direction towards more complex methods. Here it is assumed that quantum states can consistently be prepared and measured without error. In Chapter 5 it is discussed how each method can be adapted to provide estimates in a setting where each measurement result has an individual probability to be wrong.

3.1 Method of Moments Estimator

Assuming there are no measurement errors, the most straightforward way of estimating a density matrix $\rho = \frac{1}{2^n}(\mathbb{I} + \sum_{i=1}^{4^n-1} c_i B_i)$ would be to do measurements for each B_i. This corresponds to sampling a Bernoulli distribution with parameter $p_i := \frac{1}{2} + \frac{1}{2}c_i$ so that the estimator $\hat{p}_i = \frac{N_{1,i}}{N_i}$ can be used, where $N_{1,i}$ is the amount of outcomes 1 when measuring B_i, and N_i is the total amount of measurements of B_i. As stated before $\hat{\rho} = \frac{1}{2^n}(\mathbb{I} + \sum i = 1^{4^n-1} \hat{c}_i B_i)$ is not guaranteed to be positive semidefinite. The gravity of this issue can be seen in the results from the following simulation.

The goal of this experiment is to convince the reader that the method of moments estimator is not a good estimator if positive semidefinite estimates are desired. In this simulation the probability of successfully obtaining a positive semidefinite estimate is computed for different sizes of systems. For each $n \in \{1,2,3,4\}$, 10000 random density matrices of size $2^n \times 2^n$ are generated*. The moment estimator $\hat{\rho}$ is then calculated based on $4^n \cdot 100$ random measurements, and it is counted how many times $\hat{\rho}$ is positive semidefinite. The result of the simulation is shown in Figure 3.1

n	Probability of PSD estimate
1	0.8952
2	0.1157
3	0
4	0

Figure 3.1: *This table shows the estimated probability of the method of moments being positive semidefinite. It shows that as n, the amount of qubits, increases, the method of moments estimator becomes worse at producing positive semidefinite results. For $n \geqslant 3$ it is so dramatic that out of 10000 estimates 0 were positive semidefinite.*

The result of the simulation is that for larger density matrices it gets increasingly less likely that the moment estimator produces a positive semidefinite estimate, so much so that for $n \geqslant 3$ qubits none of the 10000 estimates were positive definite. When generating the matrices for larger n they tend to become less pure, but this is not what causes the difficulty in reconstructing them. In the next simulation it is shown that density matrices of lower purity are actually easier to reconstruct.

In this simulation, density matrices for $n = 2$ are generated and their purity is calculated. Again, based on $4^n \cdot 100$ random measurements the method of moments estimator is calculated, and whether the estimate is positive semidefinite or not. Afterwards bins of purity are made by dividing the difference between the highest purity and lowest purity into 6 equally sized parts. Then for each bin it is calculated how many times a matrix in this bin was positive semidefinite divided by how many times a matrix fell in this bin. This will provide information on approximately what the probability is of successfully obtaining a positive semidefinite estimate given the purity of the real state. The result can be found in Figure 3.2

*These are generating by creating a matrix A that has complex standard normally distributed entries. The matrix $A^*A/\mathrm{Tr}\,(A^*A)$ is then a density matrix. This fact will be proven in Lemma 6

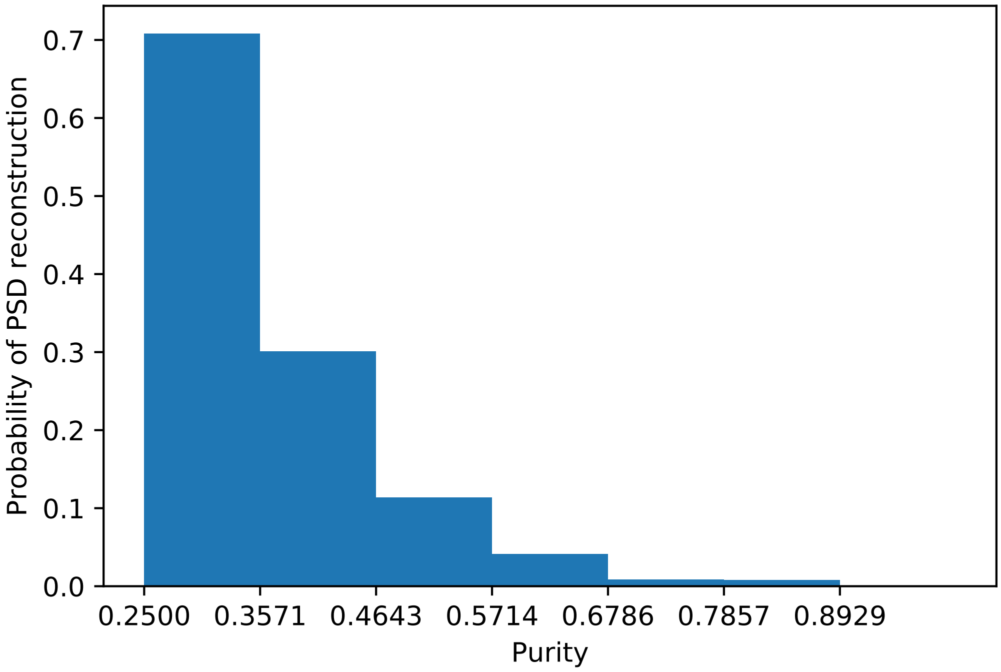


Figure 3.2: *The result of the experiment described above. It shows that the higher the purity of a state, the lower the probability is that the moment estimator produces a positive semidefinite estimation. This figure should not be read as a histogram, it states that for 70% of the matrices with purity between 0.25 and 0.36 the moment estimator produced a positive semidefinite estimate*

In conclusion, this method does not work particularly well if positive semidefinite estimates are desired. However, it is definitely debatable if an estimator is “bad” if it does not result in positive semidefinite matrices. In some applications, an experimentalist is only interested in a few of the coefficients of ρ in the Pauli basis, meaning that only a few of the Bernoulli parameters in this model are important. More can be read on this in Chapter 7.

3.2 Maximum Likelihood Estimation

Maximum likelihood estimation is a common method in statistics to estimate some parameter. Given a data set, it uses the parameter at which the likelihood is maximized as an estimator for the true value. This thesis only contains discrete random variables, so a simpler definition of likelihood can be used.

Definition 7 *Let X be a discrete random variable with probability mass function p_θ dependent on a parameter θ. The likelihood is a function $\theta \to L(\theta; x)$, where $L(\theta; x) = p_\theta(x) = \mathbb{P}(X = x|\theta)$.*

The likelihood is a function that maps a parameter θ to the probability of it producing data x. Usually, X is a vector of i.i.d random variables $(X_1, \ldots X_N)$, in which case the likelihood can be written as

$$L(\theta; x) = \prod_{i=1}^{N} \mathbb{P}(X_i = x_i|\theta).$$

Definition 8 *The maximum likelihood estimator (MLE) is $\hat{\theta}_{MLE} = argmax_\theta L(\theta; x)$.*

In order to efficiently compute this maximum the derivative of the likelihood has to be known, which is cumbersome since it consists of a big product of factors dependent on θ. This can be solved by taking the logarithm of the likelihood. Since the logarithmic function is increasing, $\hat{\theta}_{MLE} = \text{argmax}_\theta \log \mathcal{L}(\theta; x)$ is an equivalent definition for the MLE, and the product breaks down to a sum when taking the logarithm of $L(\theta; x)$.

In the problem of quantum state estimation, the log-likelihood is given by

$$\ell(p; o) = \sum_{i=1}^{4^n-1} \sum_{j=1}^{N_i} \left[o_{i,j} \log(p_i) + (1 - o_{i,j}) \log(1 - p_i)\right]$$

where $o_{i,j}$ is the outcome of the $j-$th measurement of B_i, and $p_i = \frac{1}{2} + \frac{1}{2}c_i$. When doing these simulations in Python, it is efficient to store data in the format `(outcome, label)`, where the outcome is 0 or 1 and the label $i \in \{1, \ldots 4^n - 1\}$, is the measurement done to get the outcome. The data should then be seen as pairs $(o_1, l_1), (o_2, l_2), \ldots (o_N, l_N)$ where each $o_j \sim \text{Ber}(p_{l_j})$. The log-likelihood can then be written as

$$\ell(p; (o, l)) = \sum_{j=1}^{N} \left[o_j \log(p_{l_j}) + (1 - o_j) \log(1 - p_{l_j})\right],$$

where N is now the total amount of data points, so the sum is over all `(outcome, label)` pairs, and p_{l_j} is the Bernoulli parameter corresponding to the label l_j. This allows an efficient and very intuitive way to evaluate the likelihood in Python due to how array indexing works. If `outcomes` is an array with N outcomes, `labels` an array with N corresponding labels, and `p` an array storing the $4^n - 1$ Bernoulli parameters, then the single line

```
sum(outcomes * log(p[labels]) + (1-outcomes) * log(1-p[labels]))
```

computes the likelihood.

The following two lemmas argue that the maximum likelihood estimator is a promising estimator for this problem.

Lemma 4 *The negative log-likelihood function is convex.*

Proof 4 *Differentiating ℓ towards any of the probabilities p_i twice results in*

$$-\frac{\partial^2 \ell}{\partial p_i^2} = \sum_{j:l_j=i} \frac{o_j}{p_i^2} + \frac{1-o_j}{(1-p_i)^2} > 0,$$

so the negative log-likelihood as a function of $p = (p_1, \ldots, p_{4^n-1})$ is convex.

Lemma 5 *The set of density matrices is convex.*

Proof 5 *Recall that complex $d \times d$ matrix ρ is positive semidefinite if and only if $x^* \rho x \geq 0$ for all $x \in \mathbb{C}^d$.*
Let ρ, ρ' be $d \times d$ density matrices, $x \in C^d$ and $\lambda \in [0, 1]$. Then

$$x^* \left(\lambda\rho + (1-\lambda)\rho'\right) x = x^* \lambda \rho x + x^*(1-\lambda)\rho' x \geqslant 0$$

since ρ and ρ' are positive semidefinite, and both λ and $1-\lambda$ are $\geqslant 0$.
Furthermore, it holds that Tr $(\lambda\rho + (1-\lambda)\rho') = \lambda + (1-\lambda) = 1$. Thus $\lambda\rho + (1-\lambda)\rho'$ is again a density matrix, proving that the space of density matrices is convex.

Convex functions on convex sets have a unique maximum, so maximum likelihood estimation should work. It is however difficult to do numerically because the boundary of the space of positive semidefinite matrices is hard to compute. Naively implementing convex optimization, e.g., via gradient descent [7], would require to check positive semidefiniteness at each iteration which is roughly of complexity $\mathcal{O}(2^{3n})$ for a $2^n \times 2^n$ matrix.

This problem may be averted by parametrizing the space in a different way, but this has not been implemented for this thesis. An idea to explore further in future work, would be to use the Cholesky decomposition of positive semidefinite matrices. The Cholesky decomposition of a positive semidefinite matrix P is is a unique upper triangular matrix U such that

$P = U^*U$. Since it is necessary that a density matrix ρ has trace 1, a normalizing factor is needed so that $\rho = U^*U/\mathrm{Tr}\,(U^*U)$. The idea would then be to calculate the partial derivatives of the likelihood function towards the entries of U, since these have no restriction. It turns out however that working this out on a piece of paper is quite involved, and the end result may not be computationally tractable nor is it without any additional theoretical inspection clear that this would work correctly. A few problems could be that in "U-space" the maximum is no longer unique since the normalizing factor undermines the uniqueness that the Cholesky decomposition provides, and the space or the likelihood as a function of U may no longer be convex. I think this is a rather interesting problem for future work.

On a positive note, for small systems the eigenvalues can still be calculated, so in these cases maximum likelihood can be implemented in the naive way. The way it is implemented in this thesis uses a form of gradient descent, in which iteratively steps are made in direction of the gradient to get to the minimum. In order to stay in the convex space of density matrices, at each iteration it is calculated whether the next step will break the positivity constraint, and if so, the size of the step is decreased. This makes it inefficient for large matrices, and can cause it to get stuck close to the boundary even though the maximum is not yet reached. This script is not a very good implementation of MLE, but serves its purpose as an example to show that MLE is not simple to do in this scenario.

Chapter 4

Bayesian methods for Quantum State Estimation

This chapter will introduce the Bayesian mean estimator, and provide details on how it can numerically be computed using MCMC methods. In particular, the Metropolis-Hastings algorithm is discussed, and it is shown how this algorithm can be applied to the specific problem of quantum state estimation where the parameter space is the set of positive semidefinite matrices. This chapter will not go into good choices for priors, these are left as a subject for further research.

4.1 Bayesian Mean Estimation

In Bayesian mean estimation, the parameter is equipped with a probability distribution called the prior, which we assume admits a density denoted by π.

Definition 9 *The posterior distribution of a parameter $\theta \in \mathbb{R}$ is given by*

$$p(\theta|x) = \frac{p(x|\theta)\pi(\theta)}{\int p_\theta(x)\pi(\theta)d\theta}.$$

The mean of the posterior can be used as an estimator: $\hat{\theta} = \int \theta p(\theta|x)d\theta$. Furthermore, the posterior distribution can be used to construct credible sets, providing justified error bars on the estimate [4].

In general, computing the required integrals can be intractable, but the posterior can often be sufficiently well approximated by taking samples from it using Monte Carlo methods. The particular method used in this thesis will be the Metropolis-Hastings algorithm.[8]

4.2 Metropolis-Hastings

In Metropolis-Hastings, a Markov chain is constructed which has a stationary distribution equal to the desired distribution. For Bayesian mean estimation, it allows sampling the unnormalized posterior distribution to create an estimate of its mean. Below is the Metropolis-Hastings algorithm, where $f(\theta)$ is the unnormalized posterior distribution.

Algorithm 1: Metropolis-Hastings

```
Initialize $S = \{\theta_0\}$
for $i = 1, 2, ...$ do
    $\theta_p \sim q(\theta|\theta_{i-1})$
    $p_{accept} \leftarrow \frac{f(\theta_p)q(\theta_{i-1}|\theta_p)}{f(\theta_{i-1})q(\theta_p|\theta_{i-1})}$
    $u \sim Uniform(0, 1)$
    if $u < p_{accept}$ then
        $\theta_i \leftarrow \theta_p$
    else
        $\theta_i \leftarrow \theta_{i-1}$
    end
    $S \leftarrow S \cup \{\theta_i\}$
end
```

The algorithm starts off by initializing an array to hold the samples it collects. Afterwards, it starts in state[*] θ_0, where it produces a proposal state according to the distribution $q(\theta|\theta_0)$. The distribution q is known as the proposal distribution. The algorithm proceeds by calculating the probability p_{accept} by which it should accept this proposal state as the new state. The next state is either the proposed state with the probability p_{accept}, or it stays in the the current state with probability $1 - p_{accept}$. The state is saved as a sample and the loop is repeated until enough samples are collected. It may take a few iterations before reaching the target distribution, so the first few samples can be thrown away, so called "burn-in" samples.

Before going into details, notice that when defining $f(\theta) = p(x|\theta)\pi(\theta)$,

[*] Here "state" is a state of a Markov chain, not a quantum state

while calculating the acceptance probability $\frac{f(\theta_p)q(\theta_{i-1}|\theta_p)}{f(\theta_{i-1})q(\theta_p|\theta_{i-1})}$ any normalization factor in f will be canceled, meaning that the posterior distribution can be sampled without being normalized. Inspecting Definition 9 also shows that the prior does not have to be normalized anyway, because any normalizing factor in the numerator would just cancel out the one in the denominator. This is useful when defining a prior over positive semidefinite matrices, since this removes any worry about properly defining probability measures over this complicated space.

Also a minor optimization can be made by calculating $\log p_{accept}$ using the log-likelihood. This is numerically easier to compute since for a large data set of many independent observations the likelihood becomes a huge product which is often hard to evaluate. The acceptance step then looks like

```
$p_{accept} \leftarrow \log f(\theta_p) + \log q(\theta_{i-1}|\theta_p) - \log f(\theta_{i-1}) - \log q(\theta_p|\theta_{i-1})$
$u \sim Uniform(0,1)$
if $u < e^{p_{accept}}$ then
    $\theta_i \leftarrow \theta_p$
else
    $\theta_i \leftarrow \theta_{i-1}$
end
```

While the algorithm promises to work under certain conditions, it does not always work well under these conditions. Some thought is required to define a good proposal distribution so that Metropolis-Hastings works efficiently. Making a graph of the state of the Markov chain at each iteration, a so called "trace plot", can help with qualitatively assessing if parameters were chosen correctly. As a simple example, consider the situation where there is only one Bernoulli parameter parameter $p \in (0,1)$ to be estimated. The Metropolis-Hastings algorithm will be used to estimate the posterior distribution of p when given data $x_1, \ldots, x_n \sim \text{Ber}(p)$. The log-likelihood is given by $\ell(p;x) = \sum_{i=1}^{n} [x_i \log(p) + (1-x_i)\log(1-p)]$, and in this example an unnormalized uniform prior $\log \pi(p) = 0$ is used.

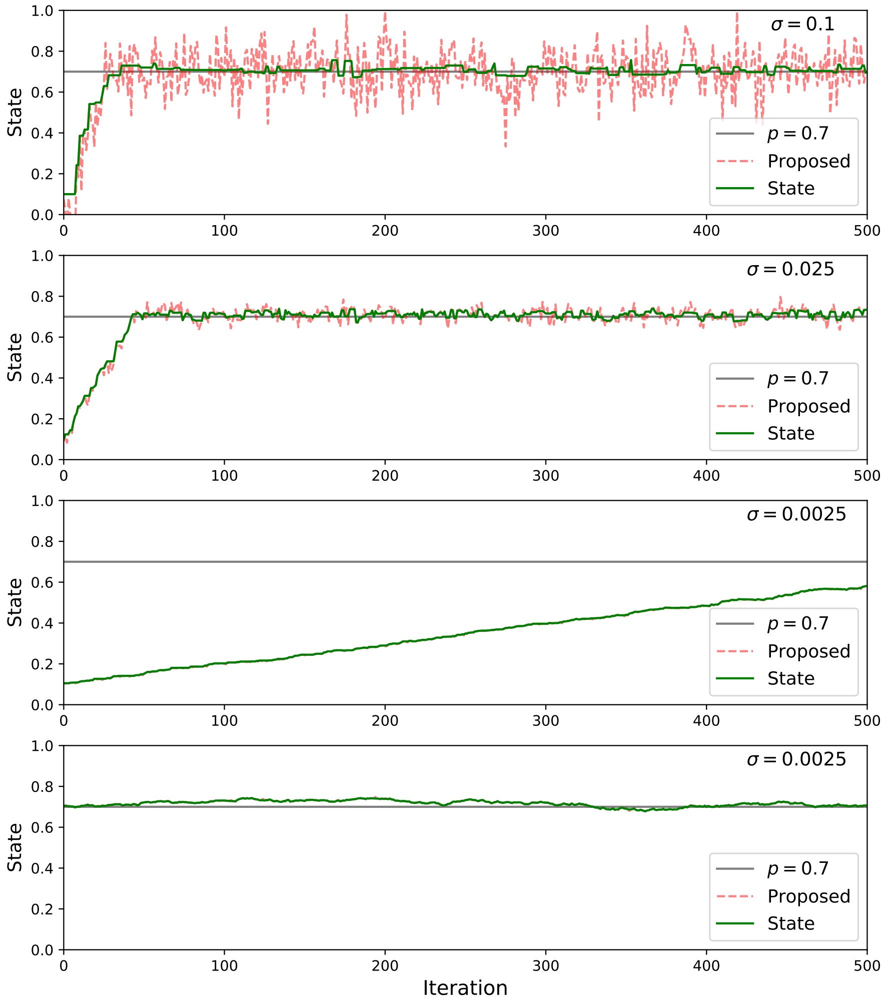


***Figure 4.1:** Trace plots for four Metropolis-Hastings runs with different standard deviations for the proposal distribution $q(p_i|p_{i-1}) = \mathcal{N}(p_{i-1}, \sigma)$ or starting states. The red line represents the states that are proposed, while the green line represents the states that are accepted. The gray horizontal line is the real Bernoulli parameter p. Trace plots show in which state the Markov chain was at each iteration during the run. They provide visual means to judge whether good parameters where chosen.*

In Figure 4.1 the importance of choosing a good proposal distribution can be seen. When using the normal distribution to generate proposals, the standard deviation can be thought of as a step size. The first plot, where $\sigma = 0.1$, demonstrates how an overlarge step size can cause the Metropolis-Hastings algorithm to sample a state too often. The proposal distribution usually proposes states far away from the current state, causing a lot of proposals to be rejected if the target distribution has relatively high density in the current state, resulting in the the Markov chain remaining in the same state for long periods of time. This will cause that state to be sampled too often. The third plot, where $\sigma = 0.0025$ demonstrates how a small step size causes the Markov chain to converge to the target distribution very slowly. Even after 500 iterations a state near $p = 0.7$ is still not reached. The last plot, where again $\sigma = 0.0025$, demonstrates how a small step size can also cause the effect known as bad mixing. Even if the target distribution is reached with a small step size, the run is still not very good. Almost every proposed state is accepted and very small steps are made, causing the chain to be near the same state for many iterations. In order to still get a good estimation a lot of samples would have to be made to explore the entire posterior. Therefore, the step size has to be not too large or too small, and in this case in the second run it can be seen that $\sigma = 0.025$ works quite well. It converges quickly, is not stuck in the same state for a long time and mixes well. The importance of burn-in is also seen in this example, in the first two runs the Markov chain goes through about 50 iterations before reaching a state around $p = 0.7$. When calculating the estimate

$$\hat{p}_{BME} = \texttt{mean(samples)}$$

it would be wise to drop the first 60 samples or so, since the Markov chain is still converging to the posterior distribution. The burn-in varies per problem, but can usually easily be chosen when looking at trace plots.

The largest inconvenience of Metropolis-Hastings is that the proposal distribution can be difficult to tune. In this simple model it does not take too long to tune the proposal distribution, but for higher dimensional problems it gets increasingly more difficult to get good results. There are theoretical results [9] in literature that give information on what efficient proposal distributions are for specific posterior distributions, but the problem in this thesis is a rather special case. It would definitely be an interesting problem for future work to find theoretical results on how this particular problem should be tuned.

4.3 Positive Semidefinite MCMC

While the process of quantum state estimation requires the estimation of Bernoulli parameters, there is a major difference between the previous example and the estimation of a density matrix ρ, namely that it is desirable that a positive semidefinite estimate is found.

A solution could be to propose states by adding a normally distributed vector to all Bernoulli parameters of the previous state, which is essentially the same as the example but p would be a $4^n - 1$ dimensional vector instead. After this, to enforce positive semidefiniteness, the eigenvalues of the corresponding density could be computed, and if any of them are negative the proposal can immediately be rejected. This however has two major problems. One of which is the same as in the case of maximum likelihood estimation, which is that computing these eigenvalues is computationally intensive. Even worse, is that for larger systems the proposals have increasingly lower probability of being positive semidefinite. This causes a lot of rejections, which are even worse now since it is so computationally intensive to check these proposals in the first place.

A way of avoiding both problems is to enforce that only positive semidefinite matrices are proposed. The following simple lemma will help.

Lemma 6 *For any matrix $A \in Mat(\mathbb{C}^n, \mathbb{C}^n)$, the matrix $A^*A/Tr(A^*A)$ is hermitian, positive semidefinite and has trace 1.*

Proof 6 *By the identity $(A^*A)^* = (A^*)^*A^* = A^*A$ it is clear that A^*A is hermitian. Furthermore, the fact that $\langle x|A^*Ax\rangle = \langle Ax|Ax\rangle \geq 0$ proves that A^*A is positive semidefinite, and by dividing by its trace it is enforced that $A^*A/Tr(A^*A)$ has trace 1.*

This lemma creates a recipe for constructing density matrices, which can be used in the Metropolis-Hastings algorithm. The key is to sample matrices $A \in \mathrm{Mat}(\mathbb{C}^n, \mathbb{C}^n)$, treating the likelihood and prior to be functions of A. Proposals for A can be created by intuitive methods, for example by adding normally distributed variables to all components, similar to the one dimensional example from earlier.

Proposing new density matrices this way, which will be called the "root method", comes with a few new issues. First of all, every density matrix ρ has multiple roots A such that $A^*A/\mathrm{Tr}(A^*A) = \rho$, which intuitively is not good. If the original posterior had a nice peak, the posterior as a function of A will have a lot of different peaks which is generally harder to sample as the Metropolis-Hastings algorithm will have a more complex distribution to explore. Furthermore, the notion of “step size” is almost gone due to renormalizing, which makes it difficult to tune the proposal distribution well. Also, it is not simple to find out how the prior should be defined on A in order to get the desired prior on ρ.

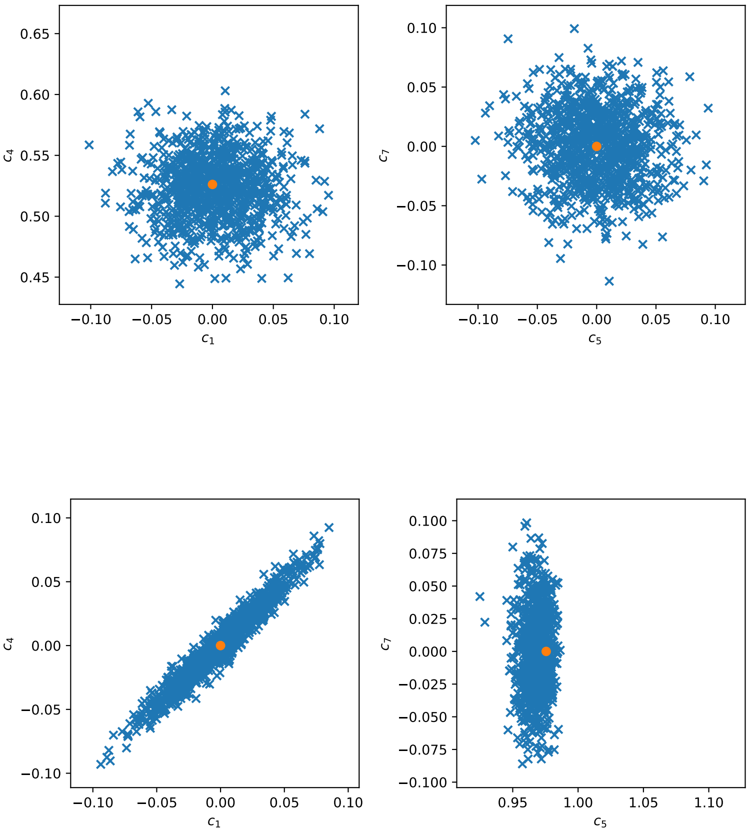


Figure 4.2: *Proposals for 4×4 density matrices drawn with the root method. The orange dot is the current state. The scatter plot is a projection of the space of density matrices onto a plane spanned by two of the fifteen coefficients. On the left these coefficients are c_1 and c_4, on the right they are c_5 and c_7. On the top two figures the current state is chosen to be not to close to the boundary, while on the bottom the current state is close to the boundary.*

A different way of proposing density matrices similar to a density ρ is by using *Lemma* 5, and a particular distribution resembling the complex Wishart distribution. The idea is to construct another density matrix ρ', and then for a fixed $\lambda \in (0, 1)$ propose the matrix $\rho^* = (1-\lambda)\rho + \lambda\rho'$. Where the convexity of the space of density matrices ensures that this is again a density matrix. A good way of constructing ρ' is to generate a random complex vector $|\phi\rangle = x + iy$ where x and y are multivariate normally distributed with covariance $\text{Re}[\rho]$, and then compute $\rho' = |\phi\rangle\langle\phi| / \text{Tr}\left(|\phi\rangle\langle\phi|\right)$. A visual simulation of this proposal distribution can ben seen in Figure 4.3. Comparing this figure to Figure 4.2 shows that this proposal distribution seems to follow the geometry of the space of density matrices more closely.

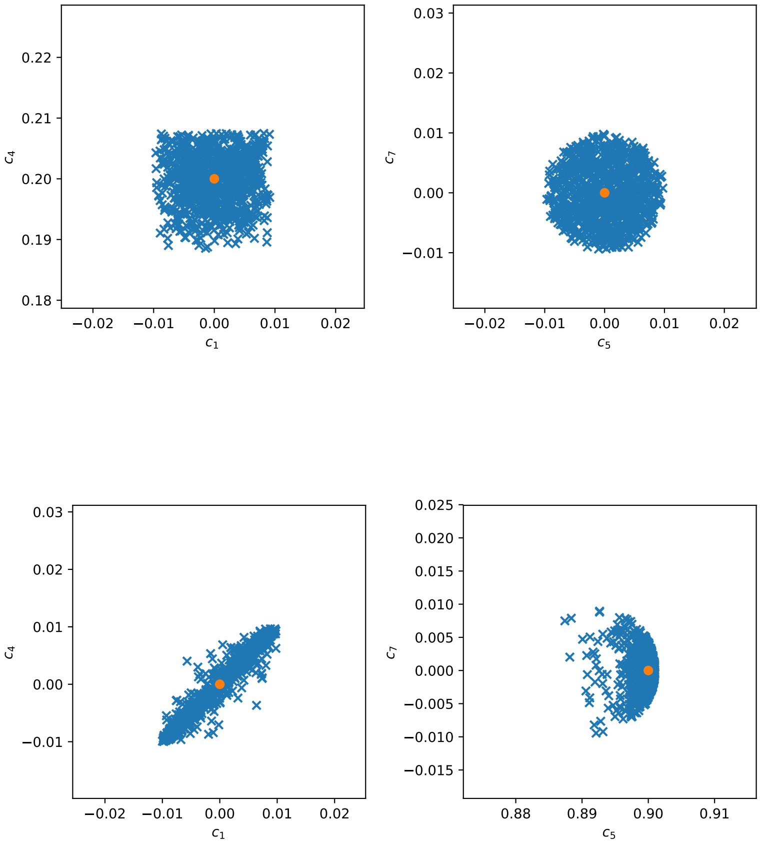


Figure 4.3: *Proposals for* 4×4 *density matrices constructed by using convexity, the orange dot is the current state. On the top two figures the current state is chosen to be not to close to the boundary, while on the bottom the current state is close to the boundary.*

This proposal method has the desirable property that it is easier to tune. The parameter λ intuitively serves as a "step size", and it can be made smaller or larger to more easily set the desired acceptance rate during the Metropolis-Hastings algorithm. The implementation of the algorithm in this thesis automatically tunes the the parameter λ during the burn-in to try to get an acceptance rate of 0.25. This is the ideal acceptance rate for a normal distribution posterior [9], and it also has decent results in this different problem. The parameter is tuned by increasing λ when the acceptance ratio is too high, and decreasing it when it is too low.

This proposal method has the undesirable property that it is difficult to calculate the density associated with this distribution, meaning that the proposal densities $q(\rho^*|\rho)$ and $q(\rho|\rho^*)$ needed for Metropolis-Hastings are hard to find. In this thesis this problem is ignored, and no proposal densities are used. This can be somewhat justified by the fact that the algorithm still performs similar to using the theoretically correct but difficult to tune root method. An interesting point of research could be to further explore this proposal distribution and theoretically derive an expression for the density.

Chapter 5

Quantum State Estimation with Experimental Errors

This chapter will introduce the concept of experimental errors and the consequences these have on the methods in Chapter 3 and Chapter 4. With some probability an experiment fails, and the outcome of a measurement does not depend on the to be estimated state. Two scenarios will be discussed, one in which measurements with a high probability of being faulty are ignored and not used in estimation, called thresholding, and one in which the actual probability of failure is incorporated into the estimator, which will be called the "error adapted" estimator.

5.1 Experimental Errors

So far it was assumed that a device could generate a state ρ consistently without error. In reality, this is not the case. With some probability the device fails to produce this state, and instead produces the maximally mixed state: $\frac{1}{2^n}\mathbb{I}$. This state has $c_i = 0$ for all i, meaning that measuring B_i for any i will return 0 or 1 with 0.5 probability. Doing a measurement on this provides no information on ρ, thus ideally the outcome of these measurements would be ignored when estimating ρ. It is sadly not possible to be certain whether the device correctly produced ρ or the maximally mixed state. It is however possible to individually create an estimate for each measurement whether the experiment succeeded or failed, and this probability can be used to improve the quantum state estimate in a few different ways. It is assumed that in this case the error estimate given is actually the real probability that the device failed to create ρ. Each measurement outcome o_j will have its own probability of being the result of a failed measurement, so data will be in the form of pairs (o_j, l_j, ϵ_j) where ϵ_j denotes the probability that this particular data point is wrong, i.e., *not* received by measuring ρ.

5.2 Thresholding Estimators

When thresholding, a threshold ϵ is set, and each data point (o_j, l_j, ϵ_j) such that $\epsilon_j > \epsilon$ is ignored, while all other points are accepted and assumed to be produced by the true state ρ so that the original methods can be used as if all data is now correct. These will obviously not work as well as they are supposed to, since there are still data points not produced by ρ used in the estimation of ρ. There is a bias-variance tradeoff when setting the threshold ϵ, as setting it low will mean that less wrong measurements that cause bias are used, but using less measurements will cause the variance of the estimator to be higher. Setting the threshold high will cause the opposite to happen. Since it is not known what the best threshold is beforehand, avoiding thresholding may be advantageous.

5.3 Error Adapted Estimators

The estimation methods in this thesis can be adapted to consider the probability of a measurement failing, so that all measurement results can be used and no threshold has to be set. When considering a probability ϵ_j of failure of the experiment, conditional probabilities can be used to derive the probability of finding outcome 0 or outcome 1 on a measurement. The probability of outcome 1 is

$$\begin{aligned}\mathbb{P}(o_j = 1) = \mathbb{P}(o_j = 1 | \rho \text{ was produced}) \mathbb{P}(\rho \text{ was produced}) + \\ \mathbb{P}(o_j = 1 | \frac{1}{2^n}\mathbb{I} \text{ was produced}) \mathbb{P}(\frac{1}{2^n}\mathbb{I} \text{ was produced})\end{aligned}$$

Originally for each measurement B_i there was a probability $p_i := \frac{1}{2} + \frac{1}{2}c_i$ of outcome 1, and measuring the mixed state results in outcome 1 with probability 0.5, so by filling in the probabilities ϵ_j of producing state $\frac{1}{2^n}\mathbb{I}$ and $(1 - \epsilon_j)$ of producing ρ it is found that

$$\mathbb{P}(o_j = 1) = (1 - \epsilon_j) p_{l_j} + \frac{1}{2}\epsilon_j.$$

The moment estimator can be adapted by taking the weighed mean over over all data points according to their probability of having been produced by the state ρ. This can be written as

$$\hat{p}_i = \frac{\sum_{j\text{s.t.}l_j=i} \left[o_j(1-\epsilon_j) - \frac{1}{2}\epsilon_j(1-\epsilon_j) \right]}{\sum_{j\text{s.t.}l_j=i}(1-\epsilon_j)},$$

where the second part in the numerator is deterministic and just serves to make the estimator unbiased.

Maximum likelihood and the Bayesian methods can be adapted by simply incorporating the correct probabilities $\mathbb{P}(o_j = 1) = (1-\epsilon_j)p_{l_j} + \frac{1}{2}\epsilon_j$ into the likelihood. This results in

$$\ell(p,(o,l,\epsilon)) = \sum_{j=1}^{N} \left[o_j \log\left((1-\epsilon_j)p_{l_j} + \frac{\epsilon_j}{2} \right) + (1-o_j)\log\left((1-\epsilon_j)(1-p_{l_j}) + \frac{\epsilon_j}{2} \right) \right].$$

After this change, the methods stay the same. Using the chain rule it can be proven that the negative log-likelihood function is still convex, so maximum likelihood works as expected.

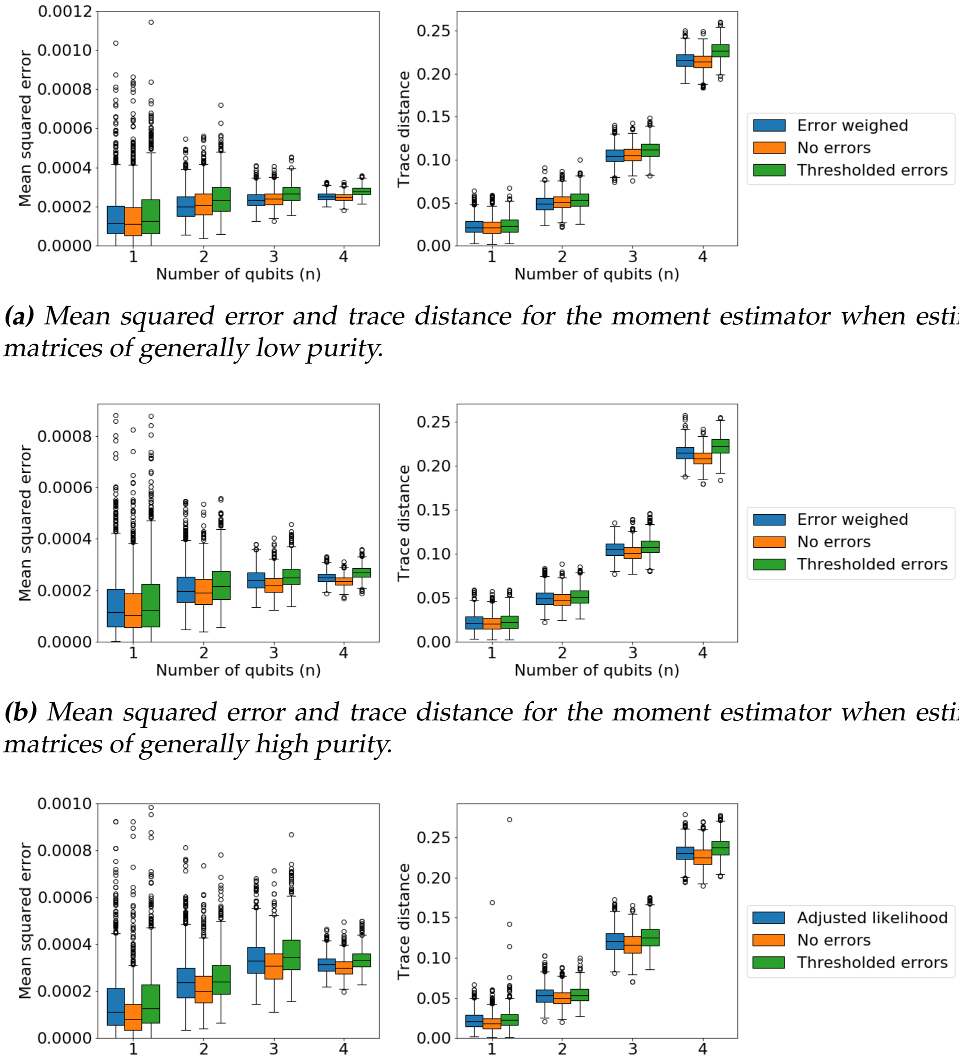
Chapter 6

Simulation Results

In this chapter each method is evaluated by how it performs without errors, when thresholding errors and when incorporating the probability of errors in the method. The methods are evaluated by the trace distance of $\hat{\rho}$ to the real state ρ, and the mean squared error of the Bernoulli parameters of $\hat{\rho}$ and ρ.

In each scenario for each $n \in \{1, 2, 3, 4\}$, 2000 density matrices of size $2^n \times 2^n$ are made, half of which are generated so that they are of high purity, and the other half is of low purity. Then for each density matrix $(4^n - 1) \cdot 1000$ random Pauli measurements are done, and the estimators are calculated. When simulating measurements, the probabilities of measurement errors occurring were sampled from a fixed distribution with most of the mass close to 0. The threshold ϵ is set at 0.05, meaning that about 10% of the measurements are not used. For the Bayesian estimator an unnormalized uniform prior is used, $\log \pi(\rho) = 0$.

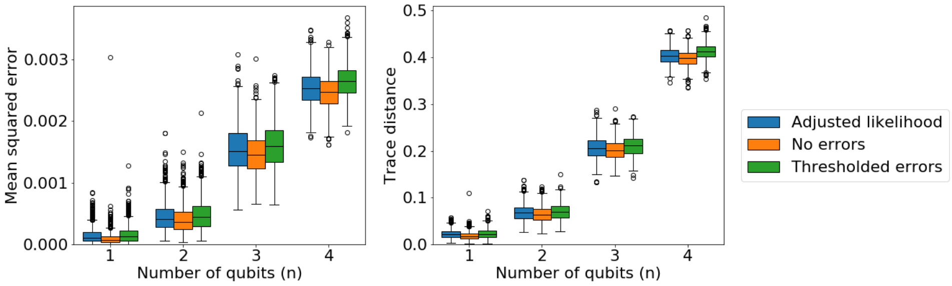
Figure 6.1: *Comparisons between the cases where there are no measurement errors, where the measurement errors are thresholded, and where the probability of errors is incorporated into the method. For each* $2^n \times 2^n$ *estimate,* $1000 \cdot (4^n - 1)$ *random Pauli measurements are done.*



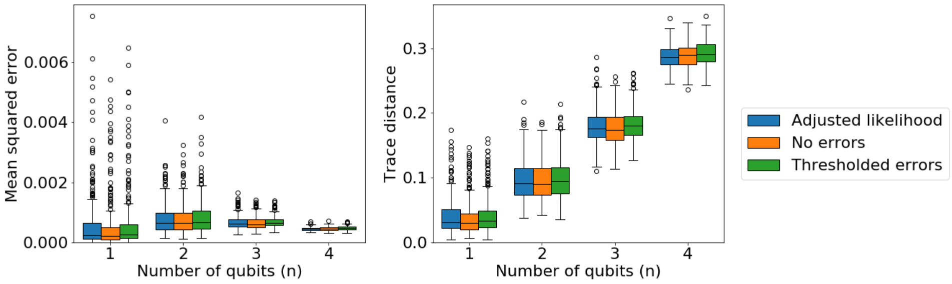
(a) *Mean squared error and trace distance for the moment estimator when estimating matrices of generally low purity.*

(b) *Mean squared error and trace distance for the moment estimator when estimating matrices of generally high purity.*

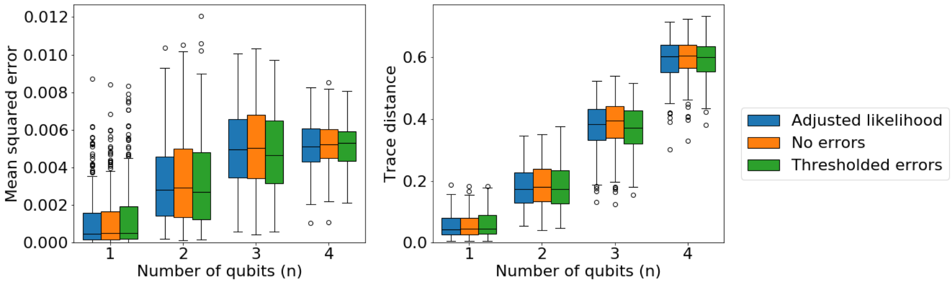
(c) *Mean squared error and trace distance for the maximum likelihood estimator when estimating matrices of generally low purity.*



(d) *Mean squared error and trace distance for the maximum likelihood estimator when estimating matrices of generally high purity.*



(e) *Mean squared error and trace distance for the Bayesian mean estimator when estimating matrices of generally low purity.*

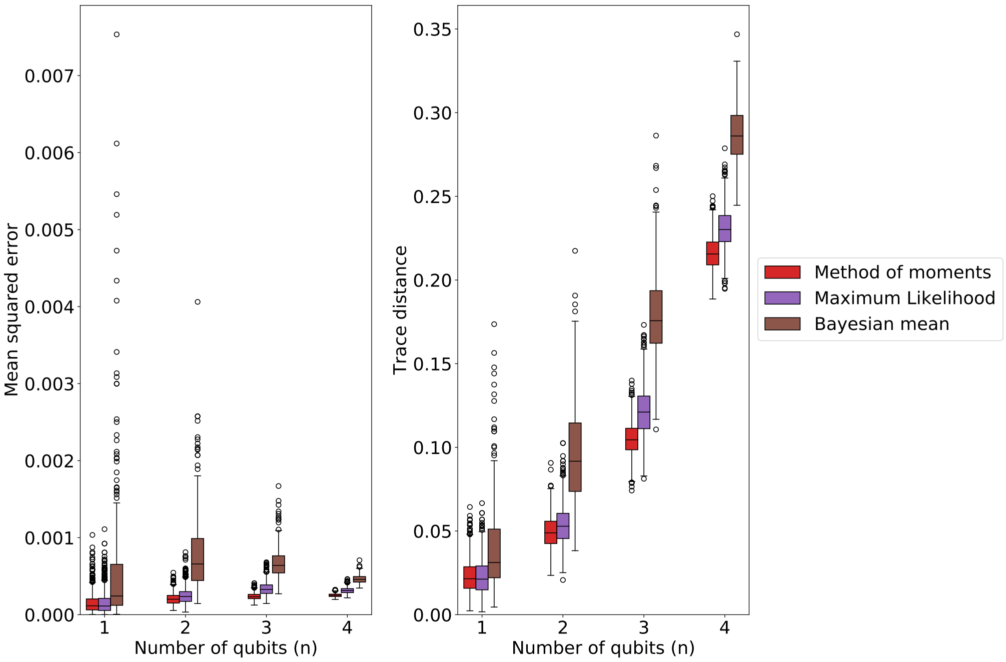


(f) *Mean squared error and trace distance for the Bayesian mean estimator when estimating matrices of generally high purity.*

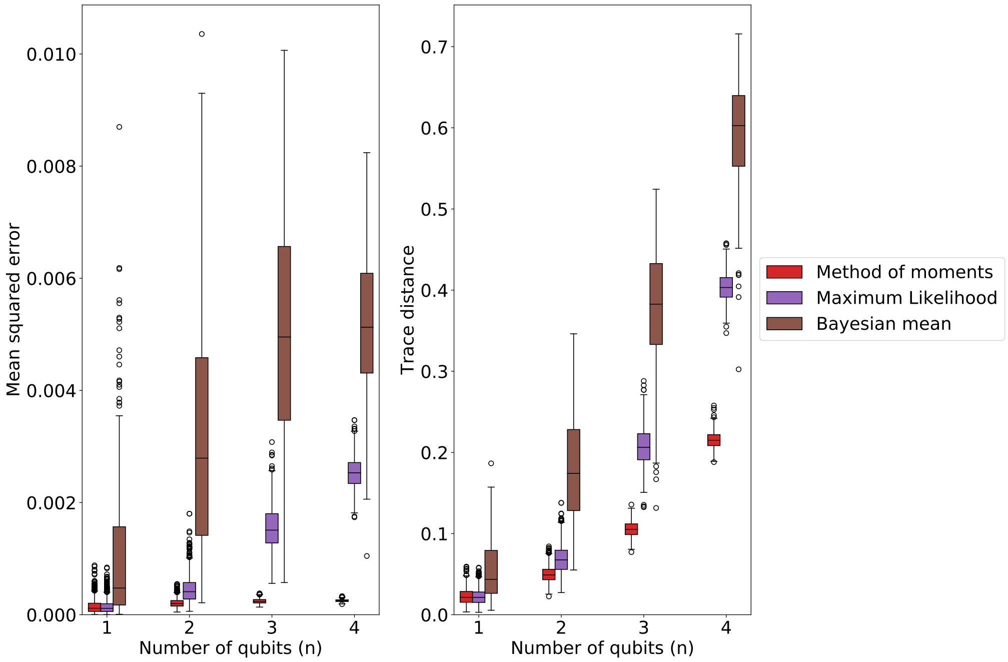
In Figure 6.1 it can be seen that for each method the error adapted estimator performs at least as well as thresholding, and sometimes slightly outperforms thresholding.

In Figure 6.2 the performance of the individual methods are compared, and the major result is that the method of moments estimator beats the other methods in both mean squared error and trace distance. This outperformance is larger when estimating states of high purity.

Figure 6.2: *A comparison of the performance of the method of moments estimator, maximum likelihood estimation, and Bayesian mean estimation when incorporating the measurement error likelihoods into the methods. They are compared in two settings: when estimating states of generally low purity, and when estimating states of generally high purity.*



(a) *A comparison of the performance of the method of moments estimator, maximum likelihood estimation, and Bayesian mean estimation when estimating states of generally low purity without measurement errors.*



(b) *A comparison of the performance of the method of moments estimator, maximum likelihood estimation, and Bayesian mean estimation when estimating states of generally high purity without measurement errors.*

Chapter 7

Discussion

The results from Figure 6.1 show that adapting the methods to use the likelihood of failed measurements should be preferred over thresholding, since it offers a slight performance boost and avoids the problem of having to set a good threshold. An important point however, is that the assumption that these error likelihoods are exactly known is false in practice, since these have to be estimated. In further research simulations could be done in which these error likelihoods are slightly distorted. It can then be tested if adapting the method to use the slightly wrong error likelihoods will still outperform thresholding, as one can expect that the performance of the thresholding estimator will not suffer as much from slightly wrong error likelihoods, meaning that thresholding might have an advantage here.

The results in Figure 6.2 show that the method of moments estimator produces the best estimates in terms of mean squared error and trace distance, even though it ignores positive semidefiniteness. If one wants to estimate the coefficients of the state as precise as possible and positive semidefiniteness is not necessary, the methods of moments estimator should be used. If the goal is to find out which quantum state a device is actually in, one would want that the estimate can physically exist. In this case the maximum likelihood estimator should be used. There is also still a reason to use the Bayesian Mean estimation, since it provides error bars on the estimate. In further research it would be interesting to expand on Bayesian methods for this specific problem. Exploring how reliable the provided error bars can be and computing the correct density of the proposal distribution used here are both interesting directions. I personally think there is a lot of room for improvement here, since knowing that the true state is positive semidefinite should intuitively allow for better estimates if the information is used correctly. Therefore I think that constructing theoretically sound implementations of Bayesian mean estimation and maximum likelihood can outperform the methods of moment estimator.

Chapter 8

Conclusions

When doing quantum state estimation, the simple method of moments estimator severely outperforms the maximum likelihood estimator and Bayesian mean estimator in the way that they are implemented here, but only if positive semidefinite estimates are not necessary. For all of these estimators, adapting them to potential measurement errors in the way that is proposed in this thesis does offer a slight performance increase over thresholding.

Bibliography

[1] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:4213, Jul 2014.

[2] Madalin Guta, Jonas Kahn, Richard Kueng, and Joel A. Tropp. Fast state tomography with optimal error bounds. *arXiv e-prints*, page arXiv:1809.11162, Sep 2018.

[3] Z. Hradil. Quantum-state estimation. *Phys. Rev. A*, 55:R1561–R1564, Mar 1997.

[4] Robin Blume-Kohout. Optimal, reliable estimation of quantum states. *New Journal of Physics*, 12(4):043034, Apr 2010.

[5] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[6] Yi-Kai Liu, Matthias Christandl, and F. Verstraete. Quantum Computational Complexity of the N-Representability Problem: QMA Complete. *Physical Review Letters*, 98(11):110503, Mar 2007.

[7] Lieven Boyd, Stephen; Vandenberghe. *Unconstrained Minimization*. Cambridge University Press, 2004.

[8] W.K. Hastings. Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika, Vol. 57, No. 1, p. 97-109, 1970*, 57:97–109, April 1970.

[9] Gelman, A., Roberts, G. O. and Gilks, W. R. Efficient metropolis jumping rules. *Bayesian Statistics 5*, pages 599–608, 1996.