Master Thesis in Quantum-Assisted Optimization for the Automation of Accounting

Simon Botond

Technical University of Munich – Datev EG

08/01/2025

Official thesis description

My tasks during the writing of the thesis:

- Work on the automated generation of accounting statements and use available graphical models, which have been trained on data.
- Apply the known representation of Bayesian Networks by Quantum Circuits and investigate the inference via Quantum Rejection Sampling.
- Investigate the feasibility of quantum algorithms to train models based on data represented in a Knowledge Graph.

The complete description: official webpage.

Advancements since then – 2025 May first half

Current state of the AA program

- Confident run at around 25 qubits on my personal computer (around $+5$ qubits even)
- Universal amplitude amplification, meaning it can be tailored to any state preparation procedure
  - uniform Oracle as in we want to work around binary networks in general so generally $\hat{O} = \hat{Z}$ or $D_O = \hat{X}\hat{Z}\hat{X}$
  - Uniform Diffusion operator as I have utilized the mathematical description of the AA procedure, where $\hat{D} = \mathcal{A}\mathcal{A}^{+}$, thus it inherently contains the state preparation unitary $\mathcal{A}$, and 'rotates' around the mean amplitude of the uniform superposition state. With this any probability sensing procedures are skipped and only the state preparation and simple diffusion are used, which is faster and scalable, as per original paper.

Next steps for the thesis – Model mapping

Extend the mapping from the hard logical use case example with widely used graphical models/distributions to exponential families and then work out the slice tensor decomposition technique (relying on tensor mapping to quantum circuits, and exponential families as Tensor Networks).

As a thought:

- Sufficient statistics $\longleftrightarrow$ groundings of formulas
- Canonical parameters $\longleftrightarrow$ weights of groundings

Which can be easily seen with the log-linear form of models/distributions:

$$\frac{1}{Z} exp\{\sum_i w_i f_i(x)\} \longleftrightarrow exp\{< \theta, \phi(x) > -A(\theta)\}$$

Next steps for the thesis – Quantum platform

As a first it would be nice to have an idea of how it could be run in the near future, and what real life 'background' do I need to have to run this procedure. So we should select a suitable quantum platform:

	Trapped ion	Superconducting
Gate fidelity	>99.9 % (1-2 qubit)	99.5-99.9 %
Coherence time	second scale	100 s - ms range
Connectivity	all-to-all	Nearest-neighbour
Speed	10-100s / gate	ns - s range

Table. 1: Basic comparison of two possible candidates for my procedure

Next steps for the thesis – Quantum platform

- Although first the Trapped ion platform would look much more promising, in my opinion the superconducting should be the one.
  - The process utilizes a lot of Hadamard, and CNot gates, which will be very important on later uses when we turn our attention to decomposing and representing various binary models/distributions, as with the slice tensor decomposition we would need to obey mod-2 calculus. Which these gates inherently do, but they are not natively encoded to trapped-ion platforms
  - Toffoli gates are also represented in both the logical connectives and the diffusion operator, which are not native to any platform, but with CNot gates being native to superconducting hardware we can efficiently use them (maybe with ancillas)

As a final thought then I would need to add some resource estimation analysis. This means:

- Selecting a suitable quantum platform
- estimating circuit depth and gate times
- Asymptotic speedup is 'given' (quadratic for Grover algorithm), but what qubit numbers can we efficiently map/compile, and what would it take to have a procedure run in real life

Advancements since then – 2025 May second half

Exponential family through a connective

I have defined the family through a connective as such:

$$p(x) = \frac{1}{Z} exp\{\theta \cdot \hat{1}[A \wedge B]\} = \frac{1}{Z} exp\{\theta \cdot \hat{1}[x_A = 1, x_B = 1]\}$$

where we see that of course we need $x_A = 1, x_B = 1$. The probabilities then are parametrized as:

- $p(x = 1) = \frac{e^\theta}{3+e^\theta} = 0.25$
- $p(x = 0) = \frac{3}{3+e^\theta} = 0.75$

from which we get the θ value, which **always** results in $\theta = 0$ for our starting distribution, as seen in the next slide

Starting canonical parameter – θ

We start with an uniform distribution as such:

- N states, $N = 2^n$, for n qubits
- M being the number of good states (N-M the 'bad' ones)
- thus we have: $p(x_{good}) = \frac{Me^{\theta}}{(N-M)+Me^{\theta}} = \frac{M}{N}$ and $p(x_{bad}) = \frac{N-M}{(N-M)+Me^{\theta}} = \frac{N-M}{N}$

where if we use $p(x_{good}) = \frac{M}{N}$, we will have:

$$
\begin{aligned}
NMe^{\theta} &= MN - M^2 + M^2 e^{\theta} \\
Ne^{\theta} &= N - M + Me^{\theta} \\
(N-M)e^{\theta} &= N - M \\
e^{\theta} &= 1 \implies \theta = 0
\end{aligned}
$$

so with each and every N and M value we will have $\theta = 0$.

Change in canonical parameter

Amplitude amplification increases the probability of measuring a "good" state. At iteration k, the amplitude (angle) is:

- $\alpha_k = (2k+1)\alpha_0$
- $P_k = \sin^2(\alpha_k)$ = probability of measuring a good state

We now interpret this probability using a canonical parameter θ, e.g., assuming a logistic form:

$$p_k(x_{\text{good}}) = \frac{Me^\theta}{(N-M) + Me^\theta}$$

Solving for θ in terms of P_k, we obtain:

$$P_k = \frac{Me^\theta}{(N-M) + Me^\theta}$$

$$e^\theta = \frac{(N-M)P_k}{M(1-P_k)}$$

Change in canonical parameter

Of course then θ can be deducted as a logarithm of the division:

$$\theta = \ln\left(\frac{(N-M)P_k}{M(1-P_k)}\right)$$

where we know that $M = NP_0$

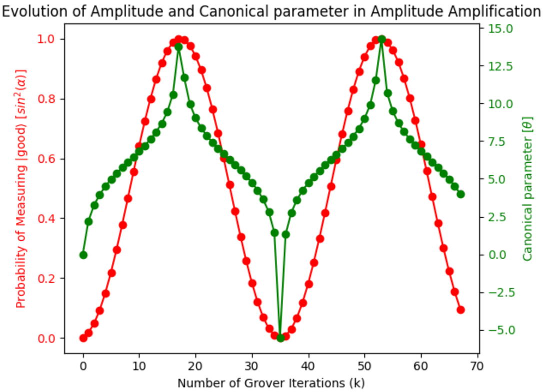
$$\theta = \ln\left(\frac{(N-NP_0)P_k}{NP_0(1-P_k)}\right)$$

So we can simplify by cancelling N:

$$\theta = \ln\left(\frac{(1-P_0)P_k}{P_0(1-P_k)}\right)$$

Visual representation

As I have shown before we start with $\theta_0 = 0$ and carry on with $\theta = \ln\left(\frac{(1-P_0)P_k}{P_0(1-P_k)}\right)$, thus θ will only depend on the starting probability P_0 and the 'current' P_k:



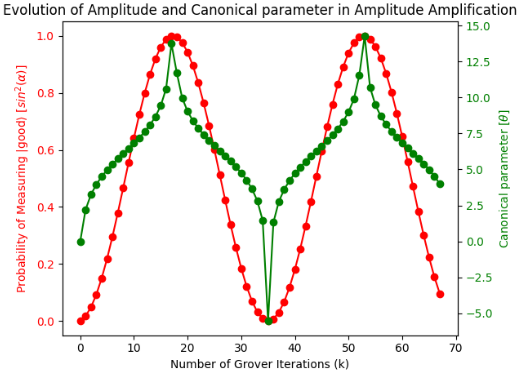
Change in the canonical parameter (shown in green curve), compared to the change in probability (red curve).

If we simplify the formula:

$$\theta = \ln\left(\frac{(1-P_0)P_k}{P_0(1-P_k)}\right)$$
$$\theta = \ln\left(\frac{P_k}{1-P_k}\right) - \ln\left(\frac{P_0}{1-P_0}\right)$$

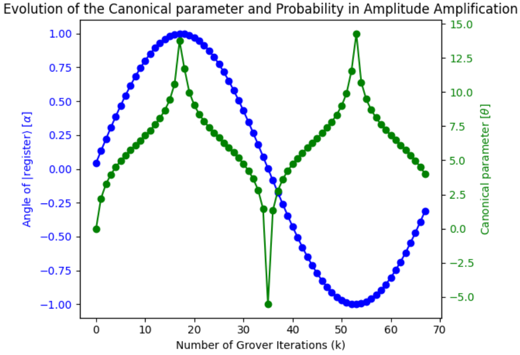
So θ_k is just the difference of log-odds (i.e. *logit*, the inverse of the *sigmoid*) between the current probability $P_k$and the initial probability P_0. The log-odds function has vertical asymptotes at $P = 0$ and $P = 1$, with the added change in behaviour coming from the non-monotonic, but sinusoidal change of P_k.

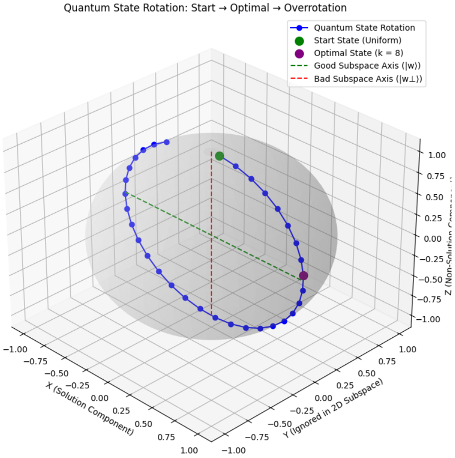
Understanding the Visual representation



- Near the center of each rise or fall, the curve looks logarithmic, but near the asymptotes the log-odds explodes, giving the appearance of exponential behavior
- Over one full amplification $sin^2(\alpha_k)$ completes $\frac{1}{4}$ oscillation, so each quarter period marks Min (0), Max (1), or mid (0.5). The log-odds transformation reacts extremely to values close to 0 or 1, causing those explosive "jumps".

Other Useful properties for visualization





- In my opinion the superconducting quantum platform would be more beneficial for the thesis:
  - The process utilizes a lot of Hadamard, and CNot gates, which are natively supported
  - Toffoli can be implemented with CNot gates being native to superconducting hardware we can efficiently use them (maybe with ancillas)
  - Superconducting platforms have undergone some advancements, just like for transmon architectures we see higher life times and dephasing times (500 μs, and 300 μs respectively), gate fidelities of 99.90%

Advancements since then – 2025 June

Writing the thesis – sections

- **Chapter 2** reviews foundational concepts in logical connectives, graphical models, and exponential familiy distributions, with their tensor network representations.
- **Chapter 3** introduces quantum computing, defining quantum gates and measurements, tensor network mapping, then turning to quantum amplitude amplification.
- **Chapter 4** destribes my implementation of mapping a one dimensional logical formula to a quantum circuit, and performing the previously defined quantum measurement and amplitude amplification.
- **Chapter 5** benchmarks quantum sampling against classical methods, with analyzing scalability, and comparing theoretical speedups, and discusses practical deployment constraints.
- **Chapter 6** Discuss broader implications, limitations, and future research directions based on results, and finally concludes with a synthesis of contributions and an outlook on quantum sampling in AI.

I figured *Chapter 2 & 3* should be the foundations, collecting all the concepts that have been **collected** throughout the research.

- **Chapter 2:** collects the classical foundations,
- **Chapter 3:** collects the quantum foundations.

So starts from various representations of data, mostly focusing on the exponential family picture; then turning to the Tensor Network representation. This latter is useful as it can be mapped to quantum circuits via slice tensor decomposition (obeying mod 2 calc.), so the knowledge base is then represented in a quantum state, from which – after a brief measurement and amplitude amplification how-to – we can sample.

Chapter 4 presents the **results** of the procedure, relying on the previously defined concepts of data representation and the utilization of quantum measurement and amplitude amplification as the sampling itself. It consists/will consist of:

- Mapping of a logical formula to a quantum circuit through exp. fam. repr. and Slice Tensor decomp.
- Evaluating the mapping/probabilities and then based on the results perform the amplitude amplification
- Show the changes in probability after the AA procedure, and how they stack up against the change in exp. fam. variables
- Include scalability, i.e. turn to higher dimensions of logical formulas, MLNs, etc.

Chapter 5 is the part where I will turn to comparative analysis:

- first analyzing the depth of the circuit the procedure requires, and comparing it to classical sampling algorithms, like Brute force sampling, rejection sampling or Gibbs sampling, and quantifying the asymptotic speedup
- after realizing the circuit depth, we can turn to realistic use cases, with NISQ devices (superconducting platform), and defining/citing coherence times, gate times, fidelities. At the end a possible turn over point from classical to quantum is expected.