ISO Presentation

Quantum Programming

Tristan Nemoz
Supervisor: Mario Berta

6 May 2020

Imperial College London

Quantum Computing applied to Machine Learning

- Goal: exponential speed-up

Quantum Computing applied to Machine Learning

- Goal: exponential speed-up
- HHL[1] (2009): $O(n) \to O\left(\log(n)\, \kappa^2\right)$

[1] Harrow, Hassidim, and Lloyd, "Quantum Algorithm for Linear Systems of Equations".

Quantum Computing applied to Machine Learning

- Goal: exponential speed-up
- HHL[1] (2009): $O(n) \to O\left(\log(n)\,\kappa^2\right)$
- Quantum Recommendation Systems[2] (2016):
  $O\left(n^2\right) \to O(\text{poly}(k)\ \text{polylog}(m\,n))$

[1] Harrow, Hassidim, and Lloyd, "Quantum Algorithm for Linear Systems of Equations".

[2] Kerenidis and Prakash, *Quantum Recommendation Systems*.

Imperial College London

Quantum Computing applied to Machine Learning

- Goal: exponential speed-up
- HHL[1] (2009): $O(n) \to O\left(\log(n)\,\kappa^2\right)$
- Quantum Recommendation Systems[2] (2016): $O\left(n^2\right) \to O(\text{poly}(k)\ \text{polylog}(m\,n))$

[1] Harrow, Hassidim, and Lloyd, "Quantum Algorithm for Linear Systems of Equations".

[2] Kerenidis and Prakash, *Quantum Recommendation Systems*.

Real-world problem overview

- m users can rate n products

Imperial College London

Real-world problem overview

- m users can rate n products (binary voting)

Real-world problem overview

- m users can rate n products (binary voting)
- Not all ratings are known

Real-world problem overview

- m users can rate n products (binary voting)
- Not all ratings are known
- Can we predict whether user j will like product k?

Real-world problem overview

- m users can rate n products (binary voting)
- Not all ratings are known
- Can we predict whether user j will like product k? Can we **recommend** a product to this user?

Mathematical description

Binary recommendation system

Let $\mathbf{P} \in \{0\,;\,1\}^{m\times n}$, whose coefficients are known with a given probability. A quantum recommendation system, given $\hat{\mathbf{P}}$ the incomplete respresentation of $\mathbf{P}$ and an user j, predicts k such that $\mathbf{P}_{j,k} = 1$ with high probability.

Mathematical description

Binary recommendation system

Let $\mathbf{P} \in \{0\,;\,1\}^{m \times n}$, whose coefficients are known with a given probability. A quantum recommendation system, given $\hat{\mathbf{P}}$ the incomplete respresentation of $\mathbf{P}$ and an user j, predicts k such that $\mathbf{P}_{j,k} = 1$ with high probability.

Solved by Kerenidis and Prakash.

Mathematical description

Binary recommendation system

Let $\mathbf{P} \in \{0\,;\,1\}^{m \times n}$, whose coefficients are known with a given probability. A quantum recommendation system, given $\hat{\mathbf{P}}$ the incomplete respresentation of $\mathbf{P}$ and an user j, predicts k such that $\mathbf{P}_{j,k} = 1$ with high probability.

Solved by Kerenidis and Prakash without practical implementation proposed.

Mathematical description

Binary recommendation system

Let $\mathbf{P} \in \{0\,;\,1\}^{m \times n}$, whose coefficients are known with a given probability. A quantum recommendation system, given $\hat{\mathbf{P}}$ the incomplete respresentation of $\mathbf{P}$ and an user j, predicts k such that $\mathbf{P}_{j,k} = 1$ with high probability.

Solved by Kerenidis and Prakash without practical implementation proposed.

Problem statement

How does the real-world implementation of the Quantum Recommendation System algorithm differs from its theoretic implementation?

Plan

Imperial College London

Classical computing

- Operates on bit-strings 0110110001

Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$

Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not

Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not
- Cloning or setting a bit to 0 can be done in $O(1)$

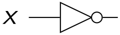
Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not
- Cloning or setting a bit to 0 can be done in $O(1)$
- Operates with a limited set of gates:

Imperial College London

Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not
- Cloning or setting a bit to 0 can be done in $O(1)$
- Operates with a limited set of gates:



(a) NOT gate

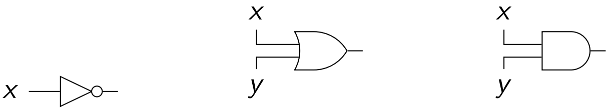
Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not
- Cloning or setting a bit to 0 can be done in $O(1)$
- Operates with a limited set of gates:

(a) NOT gate **(b)** OR gate

Classical computing

- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not
- Cloning or setting a bit to 0 can be done in $O(1)$
- Operates with a limited set of gates:

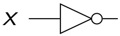


(a) NOT gate **(b)** OR gate **(c)** AND gate

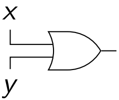
Figure: Classical gates

Classical computing

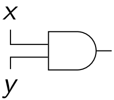
- Operates on bit-strings 0110110001
- Bits are elements of $\{0\,;\,1\}$
- Some operations are reversible, some are not
- Cloning or setting a bit to 0 can be done in $O(1)$
- Operates with a limited set of gates:



(a) NOT gate



(b) OR gate



(c) AND gate

Figure: Classical gates

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**

Imperial College London

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations are reversible

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible
- Cloning or setting a qubit to $|0\rangle$ is **impossible**

Quantum computing

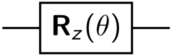
- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible
- Cloning or setting a qubit to $|0\rangle$ is **impossible** (in the general case)

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible
- Cloning or setting a qubit to $|0\rangle$ is **impossible** (in the general case)
- Operates with a limited set of matrices/gates:

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible
- Cloning or setting a qubit to $|0\rangle$ is **impossible** (in the general case)
- Operates with a limited set of matrices/gates:



(a) The $\mathbf{R}_z(\theta)$ gate

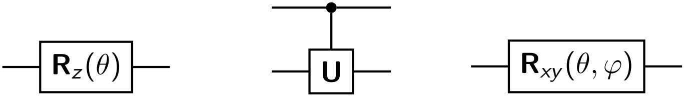
Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible
- Cloning or setting a qubit to $|0\rangle$ is **impossible** (in the general case)
- Operates with a limited set of matrices/gates:

(a) The $\mathbf{R}_z(\theta)$ gate **(b)** A controlled gate

Quantum computing

- Operates on **qubits**-strings $|q_1\rangle\ |q_2\rangle\ |q_3\rangle\ |q_4\rangle\ |q_5\rangle$
- Qubits are **normalized vectors of $\mathbf{C}^2$**
- **All** operations (aside from measuring) are reversible
- Cloning or setting a qubit to $|0\rangle$ is **impossible** (in the general case)
- Operates with a limited set of matrices/gates:



(a) The $\mathbf{R}_z(\theta)$ gate **(b)** A controlled gate **(c)** The $\mathbf{R}_{xy}(\theta, \varphi)$ gate

Figure: Quantum gates

Quantum Computing formalism

Qubits

A qubit $|q\rangle$ is a normalized vector of $\mathbf{C}^2$. We denote:

$$|q\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Quantum Computing formalism

Qubits

A qubit $|q\rangle$ is a normalized vector of $\mathbf{C}^2$. We denote:

$$|q\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Quantum Computing formalism

Qubits

A qubit $|q\rangle$ is a normalized vector of $\mathbf{C}^2$. We denote:

$$|q\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha \, |0\rangle + \beta \, |1\rangle \, .$$

Quantum Computing formalism

Qubits

A qubit $|q\rangle$ is a normalized vector of $\mathbf{C}^2$. We denote:

$$|q\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle\,.$$

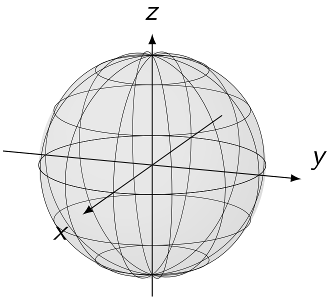
A qubits-string is called a **quantum register**.

The Bloch sphere

A qubit $|q\rangle$ can be represented on the so-called Bloch sphere:

The Bloch sphere

A qubit $|q\rangle$ can be represented on the so-called Bloch sphere:



The Bloch sphere

A qubit $|q\rangle$ can be represented on the so-called Bloch sphere:

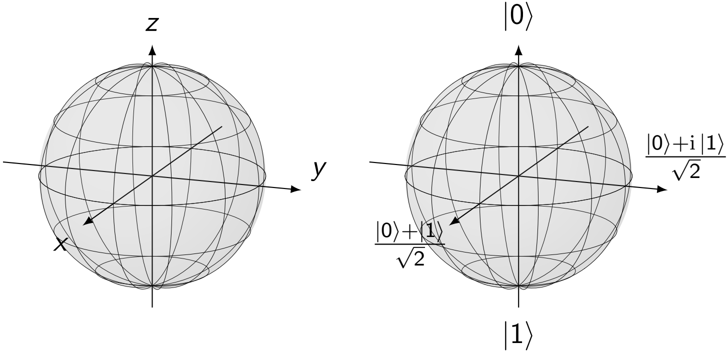


Figure: The Bloch Sphere

The Bloch sphere

A qubit $|q\rangle$ can be represented on the so-called Bloch sphere:

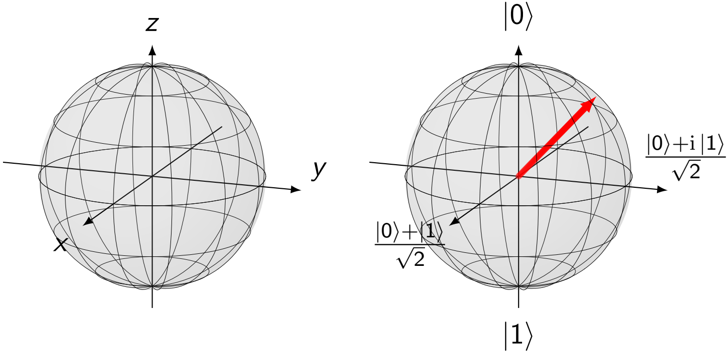


Figure: The Bloch Sphere

Quantum Computing principles

- Possibility to encode information with qubits-strings

Quantum Computing principles

- Possibility to encode information with qubits-strings
- Linearity allows to apply operations on several qubits-string at a time

Quantum Computing principles

- Possibility to encode information with qubits-strings
- Linearity allows to apply operations on several qubits-string at a time
- Possibility to get a probabilistic view of a qubit at the price of forcing it into a certain state.

Quantum Computing principles

- Possibility to encode information with qubits-strings
- Linearity allows to apply operations on several qubits-string at a time
- Possibility to get a probabilistic view of a qubit at the price of forcing it into a certain state. Measurement destroys information.

Plan

Imperial College London

Implementing a Quantum Recommendation system

- The real-world implementation of the algorithm is quite straight-forward, except for:

Implementing a Quantum Recommendation system

- The real-world implementation of the algorithm is quite straight-forward, except for:
  - Loading a vector stored in a classical structure as a quantum state

Implementing a Quantum Recommendation system

- The real-world implementation of the algorithm is quite straight-forward, except for:
  - Loading a vector stored in a classical structure as a quantum state
  - Applying the Quantum Phase Estimation subroutine

Implementing a Quantum Recommendation system

- The real-world implementation of the algorithm is quite straight-forward, except for:
  - Loading a vector stored in a classical structure as a quantum state
  - Applying the Quantum Phase Estimation subroutine
  - Comparing a qubits-string and a bits-string

Implementing a Quantum Recommendation system

- The real-world implementation of the algorithm is quite straight-forward, except for:
  - Loading a vector stored in a classical structure as a quantum state
  - Applying the Quantum Phase Estimation subroutine
  - Comparing a qubits-string and a bits-string

Loading a vector as a quantum state

Loading a vector as a quantum state

Let $\mathbf{x} \in \mathbf{R}^n$ be a normalized vector.

Loading a vector as a quantum state

Loading a vector as a quantum state

Let $\mathbf{x} \in \mathbf{R}^n$ be a normalized vector.Then, its associated quantum state is:

$$|x\rangle = \sum_{j \in \{0\,;\,1\}^{\lceil \log_2(n) \rceil}} \mathbf{x}_j \, |j\rangle \; .$$

Loading a vector as a quantum state

Loading a vector as a quantum state

Let $\mathbf{x} \in \mathbf{R}^n$ be a normalized vector. Then, its associated quantum state is:

$$|x\rangle = \sum_{j \in \{0\,;\,1\}^{\lceil \log_2(n) \rceil}} \mathbf{x}_j \, |j\rangle \, .$$

Loading $\mathbf{x}$ means creating $|x\rangle$ from $|0\rangle^{\otimes \lceil \log_2(n) \rceil}$ with a polylogarithmic number of gates[3].

Imperial College London

[3] Prakash, "Quantum Algorithms for Linear Algebra and Machine Learning."

QRAM[4]

A QRAM is a binary tree whose leaves stores the coefficients of a vector $\mathbf{x}$ and whose nodes stores the sum of its leaves values.

[4] Prakash, "Quantum Algorithms for Linear Algebra and Machine Learning."

QRAM[4]

A QRAM is a binary tree whose leaves stores the coefficients of a vector **x** and whose nodes stores the sum of its leaves values.

1

Figure: An example of a QRAM tree

[4] Prakash, "Quantum Algorithms for Linear Algebra and Machine Learning."

Quantum RAM

QRAM[4]

A QRAM is a binary tree whose leaves stores the coefficients of a vector $\mathbf{x}$ and whose nodes stores the sum of its leaves values.

$$1$$

$$\mathbf{x}_1^2 + \mathbf{x}_2^2 \qquad\qquad \mathbf{x}_3^2 + \mathbf{x}_4^2$$

Figure: An example of a QRAM tree

[4] Prakash, "Quantum Algorithms for Linear Algebra and Machine Learning."

QRAM[4]

A QRAM is a binary tree whose leaves stores the coefficients of a vector $\mathbf{x}$ and whose nodes stores the sum of its leaves values.

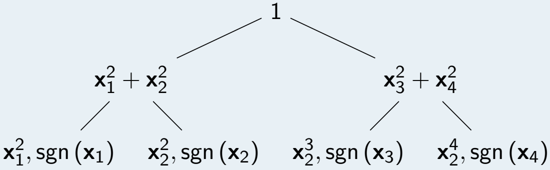


Figure: An example of a QRAM tree

[4] Prakash, "Quantum Algorithms for Linear Algebra and Machine Learning."

Loading from QRAM

- Method described in Dervovic et al., *Quantum linear systems algorithms: a primer*

Loading from QRAM

- Method described in Dervovic et al., *Quantum linear systems algorithms: a primer*
- Uses 2^k rotations gates around the y-axis at level k of the tree whose angle are determined using QRAM

Loading from QRAM

- Method described in Dervovic et al., *Quantum linear systems algorithms: a primer*
- Uses 2^k rotations gates around the y-axis at level k of the tree whose angle are determined using QRAM
- Goal: parallelize the execution of the rotations using superposition

Parallel execution

Parallel execution of controlled rotations

Let $|x\rangle$ be a n-qubits quantum register and $|0\rangle$ be a target qubit.

Parallel execution

Parallel execution of controlled rotations

Let $|x\rangle$ be a n-qubits quantum register and $|0\rangle$ be a target qubit. Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$

Parallel execution

Parallel execution of controlled rotations

Let $|x\rangle$ be a n-qubits quantum register and $|0\rangle$ be a target qubit. Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.

Parallel execution

Parallel execution of controlled rotations

Let $|x\rangle$ be a n-qubits quantum register and $|0\rangle$ be a target qubit. Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.

QRAM function assumption

It is possible, using QRAM, to design a gate $\mathbf{L}_k$ such that:

$$\mathbf{L}_k \; |k\rangle \; |0\rangle^{\otimes t} = |k\rangle \; \left|\overline{\theta_k}\right\rangle$$

where $\overline{\theta_k}$ is the best t-bits approximation of θ_k.

A simpler problem

Parallel execution of rotations

It is possible to get the state $|\theta\rangle \, \mathrm{e}^{\mathrm{i}\,\theta} \, |x\rangle$ from the state $|\theta\rangle \, (\alpha \, |0\rangle + \beta \, |1\rangle)$

A simpler problem

Parallel execution of rotations

It is possible to get the state $|\theta\rangle\ e^{i\theta}\ |x\rangle$ from the state $|\theta\rangle\ (\alpha\ |0\rangle + \beta\ |1\rangle)$ in time $O(1)$.

A simpler problem

Parallel execution of rotations

It is possible to get the state $|\theta\rangle\ \mathrm{e}^{\mathrm{i}\,\theta}\ |x\rangle$ from the state $|\theta\rangle\ (\alpha\ |0\rangle + \beta\ |1\rangle)$ in time $O(1)$.

The z-rotation

Given an angle θ, the rotation around the z-axis of the Bloch sphere is given by the gate:

$$\mathbf{R}_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\,\theta} \end{pmatrix}\ .$$

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

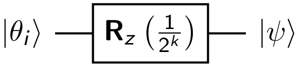


Figure: Effect of the z-rotation on a qubit

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

$$|\theta_i\rangle \;-\!\!\!-\; \boxed{\mathbf{R}_z\left(\frac{1}{2^k}\right)} \;-\!\!\!-\; |\psi\rangle$$

Figure: Effect of the z-rotation on a qubit

$$|\psi\rangle =$$

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

$$|\theta_i\rangle \;—\boxed{\mathbf{R}_z\left(\frac{1}{2^k}\right)}—\; |\psi\rangle$$

Figure: Effect of the z-rotation on a qubit

$$|\psi\rangle = \begin{cases} |0\rangle & \text{if } \theta_i = 0 \\ & \end{cases}$$

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

$$|\theta_i\rangle \; — \boxed{\mathbf{R}_z\left(\tfrac{1}{2^k}\right)} — \; |\psi\rangle$$

Figure: Effect of the z-rotation on a qubit

$$|\psi\rangle = \begin{cases} |0\rangle & \text{if } \theta_i = 0 \\ \mathrm{e}^{\mathrm{i}\,2^{-k}}\;|1\rangle & \text{if } \theta_i = 1 \end{cases}$$

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

$$|\theta_i\rangle \;—\; \boxed{\mathbf{R}_z\left(\tfrac{1}{2^k}\right)} \;—\; |\psi\rangle$$

Figure: Effect of the z-rotation on a qubit

$$|\psi\rangle = \begin{cases} |0\rangle & \text{if } \theta_i = 0 \\ \mathrm{e}^{\mathrm{i}\,2^{-k}}\;|1\rangle & \text{if } \theta_i = 1 \end{cases} = \mathrm{e}^{\mathrm{i}\,\theta_i\,2^{-k}}\;|\theta_i\rangle$$

Parallel execution of rotations

Let $\theta_i \in \{0\,;\,1\}$.

$$|\theta_i\rangle \;-\!\!\boxed{\mathbf{R}_z\left(\tfrac{1}{2^k}\right)}\!\!-\; |\psi\rangle$$

Figure: Effect of the z-rotation on a qubit

$$|\psi\rangle = \begin{cases} |0\rangle & \text{if } \theta_i = 0 \\ \mathrm{e}^{\mathrm{i}\,2^{-k}}\;|1\rangle & \text{if } \theta_i = 1 \end{cases} = \mathrm{e}^{\mathrm{i}\,\color{red}{\theta_i}\,2^{-k}}\;|\theta_i\rangle$$

Parallel execution of rotations

Let $\theta = \sum_{k=0}^{t-1} \theta_k \, 2^{-k}$.

Parallel execution of rotations

Let $\theta = \sum_{k=0}^{t-1} \theta_k \, 2^{-k}$.

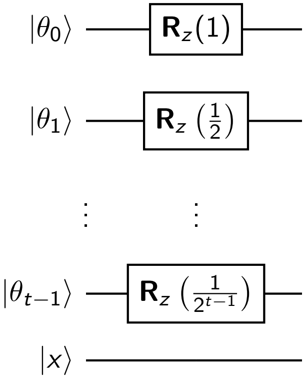
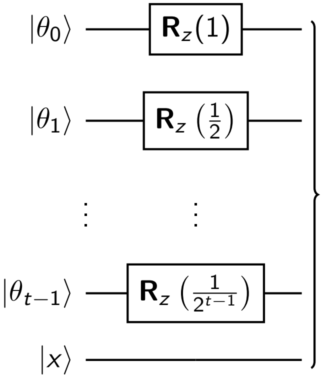


Figure: Parallel execution of z-rotations

Parallel execution of rotations

Let $\theta = \sum_{k=0}^{t-1} \theta_k \, 2^{-k}$.



$$\mathrm{e}^{\mathrm{i} \sum_{k=0}^{t-1} \theta_k \, 2^{-k}} \; |\theta\rangle \; |x\rangle = \mathrm{e}^{\mathrm{i}\,\theta} \; |\theta\rangle \; |x\rangle$$

Figure: Parallel execution of z-rotations

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Applying the z-rotation gates in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k in time $O(1)$.

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Applying the z-rotation gates in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k in time $O(1)$.

Imperial College London

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Applying the z-rotation gates in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k in time $O(1)$.

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Controlling the z-rotation gates on the target qubit in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time $O(t)$.

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Controlling the z-rotation gates on the target qubit in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time $O(t)$.

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Controlling the z-rotation gates on the target qubit in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time $O(t)$.

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Controlling the z-rotation gates on the target qubit in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time $O(t)$.

Parallel execution of controlled z-rotations

As a recall:

> *Parallel execution of controlled rotations consists in applying on the target qubit a rotation around the y-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time constant with respect to n.*

Controlling the z-rotation gates on the target qubit in the previous circuit applies a rotation on the target qubit around the z-axis of angle θ_k only if the first quantum register is in state $|k\rangle$ in time $O(t)$.

Is it possible to transform a y-rotation into a z-rotation of the same angle?

Converting a z-rotation to a y-rotation

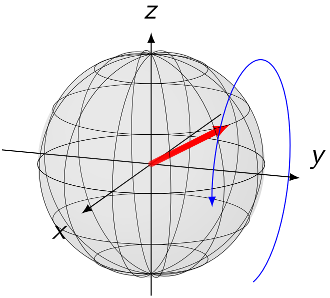


Figure: An exemple of a transformation of a y-rotation into a z-rotation

Converting a z-rotation to a y-rotation

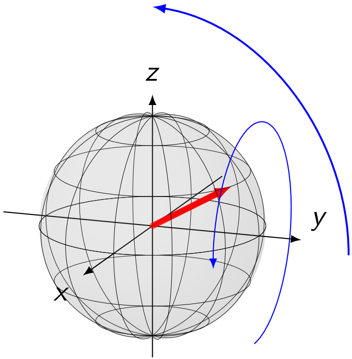


Figure: An exemple of a transformation of a y-rotation into a z-rotation

Converting a z-rotation to a y-rotation

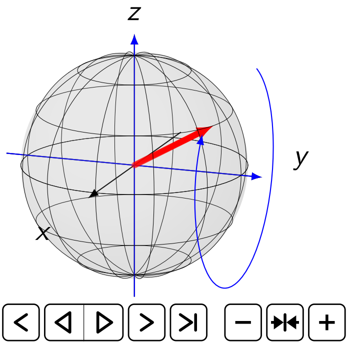


Figure: An exemple of a transformation of a y-rotation into a z-rotation

Converting a z-rotation to a y-rotation

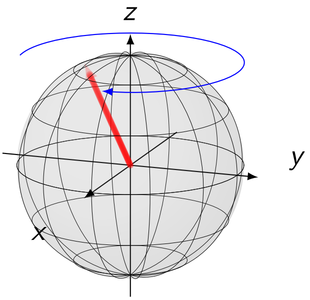


Figure: An exemple of a transformation of a y-rotation into a z-rotation

Converting a z-rotation to a y-rotation

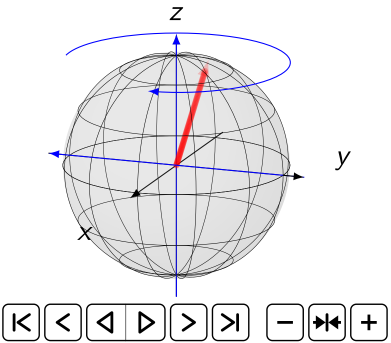


Figure: An exemple of a transformation of a y-rotation into a z-rotation

Plan

Errata

- It **is** possible to apply QPE without measuring

Errata

- It **is** possible to apply QPE without measuring (the error analysis is complex though)

Plan

Imperial College London

Conclusion

- QPE, strings comparison

Conclusion

- QPE, strings comparison
- Difference between real-world implementation and theoretic implementation

Conclusion

- QPE, strings comparison
- Difference between real-world implementation and theoretic implementation (dequantized algorithms)

References I

- Dervovic, Danial et al. *Quantum linear systems algorithms: a primer*. 2018. arXiv: 1802.08227 [quant-ph].
- Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103.15 (2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.103.150502. URL: http://dx.doi.org/10.1103/PhysRevLett.103.150502.
- Kerenidis, Iordanis and Anupam Prakash. *Quantum Recommendation Systems*. 2016. arXiv: 1603.08675 [quant-ph].
- Prakash, Anupam. "Quantum Algorithms for Linear Algebra and Machine Learning.". PhD thesis. EECS Department, University of California, Berkeley, 2014. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-211.html.

Imperial College London