

# Quantum Gibbs Sampling

Fernando G.S.L. Brandão  
Caltech

Based on joint work with  
Krysta Svore (MSR) and Michael Kastoryano (U. Copenhagen)

SQUINT 2017

# Gibbs States

Thermal state of Hamiltonian  $H$  at temperature  $T$ :

$$\rho_T = \frac{e^{-H/T}}{\text{tr}(e^{-H/T})}$$

Crucial to understand equilibrium properties of system at finite temperature

Questions:

1. When can we prepare  $\rho_T$  in reasonable time?
2. What can we do with  $\rho_T$ ?

# Classical Gibbs States

Questions:

1. When can we sample from  $\rho_T$  in reasonable time?

Well developed theory:

**Algorithm:** Glauber dynamics (Metropolis)

Consider e.g. Ising model:  $H = \sum_{\langle i,j \rangle} J_{ij} s_i s_j + \sum_i g_i s_i$

Coupling to bath modeled by stochastic map  $Q$

$$|\uparrow \dots \uparrow \downarrow \uparrow \downarrow \dots\rangle \rightarrow |\uparrow \dots \uparrow \uparrow \uparrow \downarrow \dots\rangle$$

**Metropolis Update:**  $Q_{i \rightarrow j} = \min(1, \exp(-\beta(E_j - E_i)))/n$

The stationary state is the thermal (Gibbs) state:  $p_i = e^{-\beta E_i} / Z$

# Classical Gibbs States

Questions:

1. When can we sample from  $\rho_T$  in reasonable time?

Well developed theory:

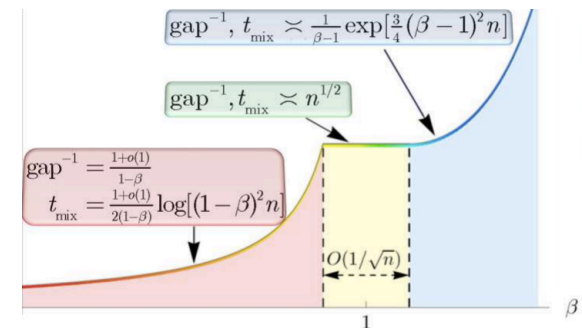
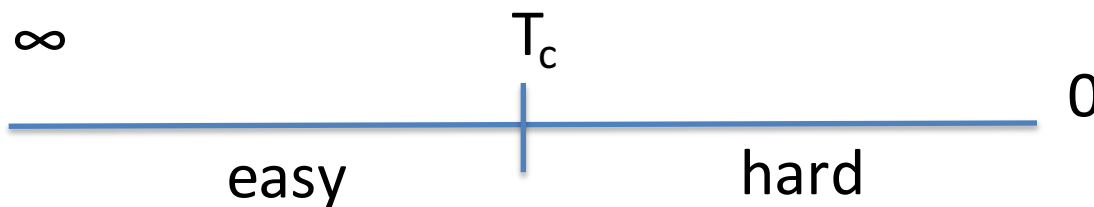
**Algorithm:** Glauber dynamics (Metropolis)

**Performance:**

(Stroock, Zergalinski '95, ...)

efficient above thermal phase transition

inefficient below (in fact NP-hard for some models (Sly '10, ...))



# Classical Gibbs States

Questions:

2. What can we do with  $\rho_T$ ?

- **Simulate thermal properties** classical models
- *And much more:*

Gibbs distributions (a.k.a. Markov random fields, Boltzmann distribution, ...) have central role in

- **Markov Chain Monte Carlo methods**  
(e.g. estimating volume convex body)
- **Machine learning**  
(e.g. Boltzmann machines)
- **Optimization**  
(e.g. matrix multiplicative update )

# Quantum Gibbs States

Questions:

1. When can we prepare  $\rho_T$  in reasonable time?

(Temme et al '07, ...) Quantum analogue of Metropolis

But **not as useful as Metropolis**

(Non-local. Very hard to analyse; no q. computer to try it out)

2. What can we do with  $\rho_T$ ?

**We don't know....**

**This talk:** progress on both questions

# This Talk

Part 1: Using Quantum Gibbs States for Quantum Speed-ups

Part 2: Conditions for Efficient Preparation of Gibbs States

# Quantum Algorithms

## Exponential speed-ups:

Simulate quantum physics, factor big numbers (Shor's algorithm), ...,

## Polynomial Speed-ups:

Searching (Grover's algorithm), ...

## Heuristics:

Quantum annealing, adiabatic optimization, ...

# Quantum Algorithms

## Exponential speed-ups:

Simulate quantum physics, factor big numbers (Shor's algorithm), ...,

## Polynomial Speed-ups:

Searching (Grover's algorithm), ...

## Heuristics:

Quantum annealing, adiabatic optimization, ...



Solving **Semidefinite Programming** belongs here

# Quantum Algorithms

## Exponential speed-ups:

Simulate quantum physics, factor big numbers (Shor's algorithm), ...

## Polynomial Speed-ups:

Searching (Grover's algorithm), ...

## Heuristics:

Quantum annealing, adiabatic optimization, ...



Solving **Semidefinite Programming** belongs here  
+ *perhaps* even exponential speed-ups too

# Semidefinite Programming

... is an important class of convex optimization problems

$$\begin{aligned} & \max \operatorname{tr}(CX) \\ \forall j \in [m], & \quad \operatorname{tr}(A_j X) \leq b_j \\ & X \succeq 0. \end{aligned}$$

Input:  $n \times n$ ,  $s$ -sparse matrices  $C, A_1, \dots, A_m$  and numbers  $b_1, \dots, b_m$

Output:  $X$

# Semidefinite Programming

... is an important class of convex optimization problems

$$\begin{aligned} & \max \operatorname{tr}(CX) \\ \forall j \in [m], & \operatorname{tr}(A_j X) \leq b_j \\ & X \succeq 0. \end{aligned}$$

Input:  $n \times n$ ,  $s$ -sparse matrices  $C, A_1, \dots, A_m$  and numbers  $b_1, \dots, b_m$

Output:  $X$

Linear Programming: special case

Many applications (combinatorial optimization, operational research, ...)

Natural in quantum (density matrices, ...)

# Semidefinite Programming

... is an important class of convex optimization problems

$$\begin{aligned} & \max \operatorname{tr}(CX) \\ \forall j \in [m], & \operatorname{tr}(A_j X) \leq b_j \\ & X \succeq 0. \end{aligned}$$

Input:  $n \times n$ ,  $s$ -sparse matrices  $C, A_1, \dots, A_m$  and numbers  $b_1, \dots, b_m$

Output:  $X$

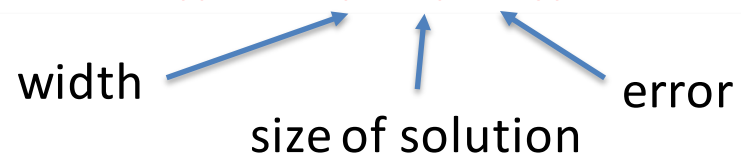
Linear Programming: special case

Many applications (combinatorial optimization, operational research, ...)

Natural in quantum (density matrices, ...)

Algorithms Interior points:  $O((m^2ns + mn^2)\log(1/\delta))$

Multiplicative Weights:  $O((mns (\omega R)/\delta^2))$



# Semidefinite Programming

... is an important class of convex optimization problems

$$\begin{aligned} & \max \operatorname{tr}(CX) \\ \forall j \in [m], & \operatorname{tr}(A_j X) \leq b_j \end{aligned}$$

Input:  $n \times n$

Output:  $n \times n$

Linear Pr

Many ap

Natural in quantum (Hermitian matrices)

Are there quantum speed-ups for SDPs/LPs?  
 Natural question. But unexplored so far

$b_m$

....)

Algorithms

Interior points:

$$O((m^2ns + mn^2)\log(1/\delta))$$

Multiplicative Weights:

$$O(mns (\omega R)/\delta^2)$$

width

size of solution

error

# SDPs and Gibbs States

What Gibbs state have to do with this?

(Jaynes 57) Let  $\rho$  be a quantum state s.t.  $\text{tr}(\rho M_i) = c_i$

Then there is a Gibbs state of the form  $\exp\left(\sum_i \lambda_i M_i\right) / \text{tr}(\dots)$

with same expectation values.

Moreover Gibbs state is the state of maximum entropy with the right expectation values

# SDPs and Gibbs States

$$\begin{array}{ll} \text{SDP:} & \begin{array}{l} \max \text{tr}(CX) \\ \forall j \in [m], \quad \text{tr}(A_j X) \leq b_j \\ X \geq 0. \end{array} \end{array}$$

By Jaynes' principle there is a solution of the form

$$X = \exp \left( \sum_j \lambda_j A_j + \lambda_0 C \right)$$

Why? Apply it to  $X_{\text{opt}}/\text{tr}(X_{\text{opt}})$  to get Gibbs state with same expectation values for  $A_j, C$

How can we find  $\lambda_j$ 's?

# SDP Duality

**Primal:**  $\forall j \in [m],$

$$\begin{aligned} & \max \operatorname{tr}(CX) \\ & \operatorname{tr}(A_j X) \leq b_j \\ & X \geq 0. \end{aligned}$$

**Dual:**

$$\begin{aligned} & \min b \cdot y \\ & \sum_{j=1}^m y_j A_j \geq C \\ & y \geq 0. \end{aligned}$$

$y$ :  $m$ -dimensional vector

Under mild conditions:  $\text{Opt}_{\text{primal}} = \text{Opt}_{\text{dual}}$

# Size of Solutions

$$\begin{aligned} \text{Primal: } & \max \operatorname{tr}(CX) \\ & \forall j \in [m], \quad \operatorname{tr}(A_j X) \leq b_j \\ & X \geq 0. \end{aligned}$$

R parameter:  $\operatorname{Tr}(X_{\text{opt}}) \leq R$

$$\begin{aligned} \text{Dual: } & \min b \cdot y \\ & \sum_{j=1}^m y_j A_j \geq C \\ & y \geq 0. \end{aligned}$$

r parameter:  $\sum_i (y_{\text{opt}})_i \leq r$

# SDP Lower Bounds

Even to write down optimal solutions take time:

**Primal** ( $n \times n$  PSD matrix  $X$ ):  $\Omega(n^2)$

**Dual** ( $m$  dim vector  $y$ ):  $\Omega(m)$

---

# SDP Lower Bounds

Even to write down optimal solutions take time:

**Primal** ( $n \times n$  PSD matrix  $X$ ):  $\Omega(n^2)$

**Dual** ( $m$  dim vector  $y$ ):  $\Omega(m)$

Even just to compute optimal value requires:

**Classical:**  $\Omega(n+m)$  (for constant  $r, R, s, \delta$ )

**Quantum:**  $\Omega(n^{1/2} + m^{1/2})$  (for constant  $r, R, s, \delta$ )

Easy reduction to search problem

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

Input:  $n \times n$ ,  $s$ -sparse matrices  $C, A_1, \dots, A_m$  and numbers  $b_1, \dots, b_m$

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

Input:  $n \times n$ ,  $s$ -sparse matrices  $C, A_1, \dots, A_m$  and numbers  $b_1, \dots, b_m$

Normalization:  $\|A_i\|, \|C\| \leq 1$

Output: Samples from  $y/\|y\|_1$  and value  $\|y\|_1$  and/or  
Quantum Samples from  $X/\text{tr}(X)$  and value  $\text{tr}(X)$

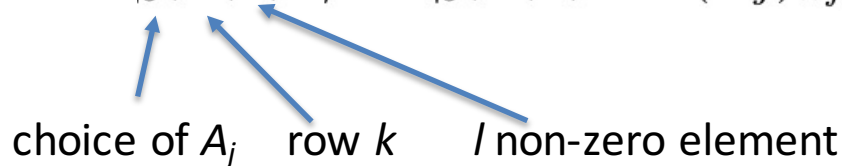
Value  $\text{opt} \pm \delta$

(output form similar to HHL Q. Algorithm for linear equations)

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

**Oracle Model:** We assume there's an oracle that outputs a chosen non-zero entry of  $C, A_1, \dots, A_m$  at unit cost:

$$|j, k, l, z\rangle \rightarrow |j, k, l, z \oplus (A_j)_{kf_{jk}(l)}\rangle \quad f_{jk} : [r] \rightarrow [n]$$


choice of  $A_j$     row  $k$      $l$  non-zero element

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

## The good:

Unconditional Quadratic speed-ups in terms of  $n$  and  $m$

Close to optimal:  $\Omega(n^{1/2} + m^{1/2})$  q. lower bound

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

## The good:

Unconditional Quadratic speed-ups in terms of  $n$  and  $m$

Close to optimal:  $\Omega(n^{1/2} + m^{1/2})$  q. lower bound

## The bad:

Close to optimal: no general super-polynomial speed-ups

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

## The good:

Unconditional Quadratic speed-ups in terms of  $n$  and  $m$

Close to optimal:  $\Omega(n^{1/2} + m^{1/2})$  q. lower bound

## The bad:

Close to optimal: no general super-polynomial speed-ups

Terrible dependence on other parameters:  $(Rr)^{32} \delta^{-18}$

# Quantum Algorithm for SDP

**Result 1:** (B., Svore '16) There is a quantum algorithm for solving SDPs running in time  $n^{1/2} m^{1/2} s^2 \text{poly}(\log(n, m), R, r, \delta)$

## The good:

Unconditional Quadratic speed-ups in terms of  $n$  and  $m$

Close to optimal:  $\Omega(n^{1/2} + m^{1/2})$  q. lower bound

## The bad:

Close to optimal: no general super-polynomial speed-ups

Terrible dependence on other parameters:  $(Rr)^{32} \delta^{-18}$

**But:** Since 2 days ago, improvement by Ambainis:  $(Rr)^{13} \delta^{-10}$

# Larger Speed-ups?

**Result 2:** There is a quantum algorithm for solving SDPs running in time  $T_{\text{Gibbs}} m^{1/2} \text{poly}(\log(n, m), s, R, r, \delta)$

# Larger Speed-ups?

**Result 2:** There is a quantum algorithm for solving SDPs running in time  $T_{\text{Gibbs}} m^{1/2} \text{poly}(\log(n, m), s, R, r, \delta)$

$T_{\text{Gibbs}}$  := Time to prepare on quantum computer Gibbs states of the form

$$\exp \left( \sum_{i=1}^m \nu_i A_i + \nu_0 C \right) / \text{tr}(\dots)$$

for real numbers  $|\nu_i| \leq O(\log(n), \text{poly}(1/\delta))$

# Larger Speed-ups?

**Result 2:** There is a quantum algorithm for solving SDPs running in time  $T_{\text{Gibbs}} m^{1/2} \text{poly}(\log(n, m), s, R, r, \delta)$

$T_{\text{Gibbs}}$  := Time to prepare on quantum computer Gibbs states of the form

$$\exp \left( \sum_{i=1}^m \nu_i A_i + \nu_0 C \right) / \text{tr}(\dots)$$

for real numbers  $|\nu_i| \leq O(\log(n), \text{poly}(1/\delta))$

Can use **Quantum Gibbs Sampling** (e.g. Quantum Metropolis) as heuristic. Exponential Speed-up if thermalization is quick (poly #qubits = polylog(n))

**Gives application of quantum Gibbs sampling outside simulating physical systems**

# Larger Speed-ups with “quantum input”

**Result 3:** There is a quantum algorithm for solving SDPs running in time  $m^{1/2} \text{poly}(\log(n, m), s, R, r, \delta, \text{rank})$  with data in quantum form

# Larger Speed-ups with “quantum input”

**Result 3:** There is a quantum algorithm for solving SDPs running in time  $m^{1/2} \text{poly}(\log(n, m), s, R, r, \delta, \text{rank})$  with data in quantum form

**Quantum Oracle Model:** There is an oracle that given  $i$ , outputs the eigenvalues of  $A_i$  and its eigenvectors as quantum states

$\text{rank} := \max(\max_i \text{rank}(A_i), \text{rank}(C))$

# Larger Speed-ups with “quantum input”

**Result 3:** There is a quantum algorithm for solving SDPs running in time  $m^{1/2} \text{poly}(\log(n, m), s, R, r, \delta, \text{rank})$  with data in quantum form

**Quantum Oracle Model:** There is an oracle that given  $i$ , outputs the eigenvalues of  $A_i$  and its eigenvectors as quantum states

$\text{rank} := \max(\max_i \text{rank}(A_i), \text{rank}(C))$

**Idea:** in this case one can easily perform the Gibbs sampling in  $\text{poly}(\log(n), \text{rank})$  time

**Question:** How relevant is this model?

# Partial Tomography

**Problem:** Given a set of measurements  $\{A_i\}$  ( $A_i^2=A_i$ ), and access to copies of an unknown quantum state  $\rho$ , want to find a description of state  $\sigma$  s.t.

$$|\text{tr}(\rho A_i) - \text{tr}(\sigma A_i)| \leq \varepsilon$$

# Partial Tomography

**Problem:** Given a set of measurements  $\{A_i\}$  ( $A_i^2=A_i$ ), and access to copies of an unknown quantum state  $\rho$ , want to find a description of state  $\sigma$  s.t.

$$|\text{tr}(\rho A_i) - \text{tr}(\sigma A_i)| \leq \varepsilon$$

This is a SDP, with  $b_j = \pm \text{tr}(A_i \rho) \pm \varepsilon$

We can implement an oracle to  $b_i$  by measuring  $A_i$  on  $\rho$

Assume the measurements are efficient ( $\text{poly}(\log(n))$  time)

# Partial Tomography

**Problem:** Given a set of measurements  $\{A_i\}$  ( $A_i^2=A_i$ ), and access to copies of an unknown quantum state  $\rho$ , want to find a description of state  $\sigma$  s.t.

$$|\text{tr}(\rho A_i) - \text{tr}(\sigma A_i)| \leq \varepsilon$$

Quantum algorithm gives  $\sigma = \exp\left(\sum_i \lambda_i A_i\right)$  and  $\lambda_i$ 's  
in time  **$m^{1/2} \text{poly}(\log(n, m), s, R, r, 1/\varepsilon, \text{rank})$**

For “low rank measurements” ( $\text{rank}(A_i) = \text{polylog}(n)$ )  
gives *exponential* speed-up

# Partial Tomography

**Problem:** Given a set of measurements  $\{A_i\}$  ( $A_i^2=A_i$ ), and access to copies of an unknown quantum state  $\rho$ , want to find a description of state  $\sigma$  s.t.

$$|\text{tr}(\rho A_i) - \text{tr}(\sigma A_i)| \leq \varepsilon$$

Quantum algorithm gives  $\sigma = \exp\left(\sum_i \lambda_i A_i\right)$  and  $\lambda_i$ 's  
in time  **$m^{1/2} \text{poly}(\log(n, m), s, R, r, 1/\varepsilon, \text{rank})$**

For “low rank measurements” ( $\text{rank}(A_i) = \text{polylog}(n)$ )  
gives *exponential* speed-up

**(Aaronson '17)** Can find  $\sigma$  with  $\text{poly}(\log(n), \log(m),)$  samples of  $\rho$

Can we improve sample complexity to match this result?

# Special Case: Max Eigenvalue

Computing the max eigenvalue of  $C$  is a SDP

$$\max \operatorname{tr}(CX): \quad \operatorname{tr}(X) = 1, \quad X \succeq 0$$

# Special Case: Max Eigenvalue

Computing the max eigenvalue of  $C$  is a SDP

$$\max \text{tr}(CX): \quad \text{tr}(X) = 1, \quad X \geq 0$$

This is a well studied problem:

**Quantum Annealing** (cool down  $-C$ ):

If we can prepare  $e^{\beta C} / \text{tr}(e^{\beta C})$  for  $\beta = O(\log(n)/\delta)$  can compute max eigenvalue to error  $\delta$

# Special Case: Max Eigenvalue

(Poulin, Wocjan '09) Can prepare  $e^{\beta C} / \text{tr}(e^{\beta C})$  for  $s$ -sparse  $C$  in time  $\tilde{O}(s n^{1/2})$  on quantum computer

**Idea:** Phase estimation + Amplitude amplification

$$C|\psi_i\rangle = E_i|\psi_i\rangle$$

$$\sum_i |\psi_i\rangle|\psi_i^*\rangle \xrightarrow{\text{phase estimation}} \sum_i |\psi_i\rangle|\psi_i^*\rangle|E_i\rangle \rightarrow \sum_i e^{-E_i/2} |\psi_i\rangle|\psi_i^*\rangle|E_i\rangle|0\rangle + \dots$$

phase estimation

Post-selecting on “0” gives a purification of Gibbs state with  $\text{Pr} > O(1/n)$

Using amplitude amplification can boost  $\text{Pr} > 1-o(1)$  with  $O(n^{1/2})$  iterations

# General Case:

# Quantizing Arora-Kale Algorithm

The quantum algorithm is based on a classical algorithm for SDP due to Arora and Kale (2007) based on the multiplicative weight method.

# General Case:

## Quantizing Arora-Kale Algorithm

The quantum algorithm is based on a classical algorithm for SDP due to Arora and Kale (2007) based on the multiplicative weight method. Let's review their method

### Assumptions:

We assume  $b_i \geq 1$ .

Can reduce general case to this with blow up of  $\text{poly}(r)$  in complexity

We also assume w.l.o.g.  $A_1 = I, b_1 = R$

Solve dual decision problem, of finding feasible  $y$  s.t.  $b \cdot y \leq \alpha$   
Optimization follows from binary search

# The Oracle

The Arora-Kale algorithm has an auxiliary algorithm (the ORACLE) which solves a simple linear programming:

ORACLE( $\rho$ )

Searches for a vector  $y$  s.t.

i)  $y \in D_\alpha := \{y : y \geq 0, b \cdot y \leq \alpha\}$

ii)  $\sum_{j=1}^m \text{tr}(A_j \rho) y_j - \text{tr}(C \rho) \geq 0$

# Arora-Kale Algorithm

$$\rho^1 = I/n, \quad \varepsilon = \frac{\delta}{2R}, \quad \varepsilon' = -\ln(1 - \varepsilon), \quad T = \frac{8R^2 \ln(n)}{\delta^2}$$

For  $t = 1, \dots, T$

1.  $y^t \leftarrow \text{ORACLE}(\rho^t)$

2.  $M^t = \left( \sum_{j=1}^m y_j^t A_j - C + RI \right) / 2R$

3.  $W^{t+1} = \exp \left( -\varepsilon' \left( \sum_{\tau=1}^t M^\tau \right) \right)$

4.  $\rho^{t+1} = W^{t+1} / \text{tr}(W^{t+1})$

Output:  $\bar{y} = \frac{\delta \alpha}{R} e_1 + \frac{1}{T} \sum_{t=1}^T y^t \quad e_1 = (1, 0, \dots, 0)$

# Arora-Kale Algorithm

$$\rho^1 = I/n, \quad \varepsilon = \frac{\delta}{2R}, \quad \varepsilon' = -\ln(1 - \varepsilon), \quad \alpha = \frac{8R^2 \ln(n)}{\delta^2}$$

For  $t = 1, \dots, T$

1.  $y^t \leftarrow \text{ORACLE}(\rho^t)$

2.

**Thm** (Arora-Kale '07)  $\bar{y} \cdot b \leq (1+\delta) \alpha$

Can find optimal value by binary search

3.

4.  $\rho^{t+1} = W^{t+1} / \text{tr}(W^{t+1})$

Output:  $\bar{y} = \frac{\delta \alpha}{R} e_1 + \frac{1}{T} \sum_{t=1}^T y^t$        $e_1 = (1, 0, \dots, 0)$

# Why Arora-Kale works?

Since  $y_t \in D_\alpha := \{y : y \geq 0, b \cdot y \leq \alpha\}$

$$\bar{y} \cdot b \leq \frac{\delta \alpha}{R} b_1 + \frac{1}{T} \sum_{t=1}^T y^t \cdot b \leq (1 + \delta) \alpha$$

---

Must check  $\bar{y}$  is feasible

From Oracle, for all  $t$ :  $\text{tr} \left( \left( \sum_{j=1}^m y_j^t A_j - C \right) \rho^t \right) \geq 0$

We need:  $\lambda_{\min} \left( \left( \sum_{j=1}^m \left( \frac{1}{T} \sum_{t=1}^T y_j^t \right) A_j - C \right) \right) \geq 0$

# Matrix Multiplicative Weight

MMW (Arora, Kale '07) Given  $n \times n$  matrices  $0 < M^t < I$  and  $\varepsilon < \frac{1}{2}$ ,

$$\frac{1}{T} \sum_{t=1}^T \text{tr}(M^t \rho^t) \leq \left( \frac{1 + \varepsilon}{T} \right) \lambda_n \left( \sum_{t=1}^T M^t \right) + \frac{\ln(n)}{T\varepsilon}$$

with  $\rho^t = \frac{\exp(-\varepsilon'(\sum_{\tau=1}^{t-1} M^\tau))}{\text{tr}(\dots)}$  and  $\varepsilon' = -\ln(1 - \varepsilon)$

$\lambda_n$  : min eigenvalue

2-player zero-sum game interpretation:

- Player A chooses density matrix  $X^t$
- Player B chooses matrix  $0 < M^t < I$

Pay-off:  $\text{tr}(X^t M^t)$

“ $X^t = \rho^t$  strategy almost as good as global strategy”

# Matrix Multiplicative Weight

MMW (Arora, Kale '07) Given  $n \times n$  matrices  $M^t$  and  $\varepsilon < \frac{1}{2}$ ,

$$\frac{1}{T} \sum_{t=1}^T \text{tr}(M^t \rho^t) \leq \left( \frac{1 + \varepsilon}{T} \right) \lambda_n \left( \sum_{t=1}^T M^t \right) + \frac{\ln(n)}{T\varepsilon}$$

with  $\rho^t = \frac{\exp(-\varepsilon'(\sum_{\tau=1}^{t-1} M^\tau))}{\text{tr}(\dots)}$  and  $\varepsilon' = -\ln(1 - \varepsilon)$

$\lambda_n$  : min eigenvalue

From Oracle:  $\text{tr} \left( \left( \sum_{j=1}^m y_j^t A_j - C \right) \rho^t \right) \geq 0$

By MMW:  $\lambda_{\min} \left( \left( \sum_{j=1}^m \left( \frac{1}{T} \sum_{t=1}^T y_j^t \right) A_j - C \right) \right) \geq 0$

# Quantizing Arora-Kale Algorithm

We make it quantum as follows:

1. Implement ORACLE by Gibbs Sampling to produce  $y^t$  and apply amplitude amplification to solve it in time  $\tilde{O}(s^2 n^{1/2} m^{1/2})$
2. Sparsify  $M^t$  to be a sum of  $O(\log(m))$  terms:

$$\overline{M}^t = \left( \|y^t\|_1 Q^{-1} \sum_{j=1}^Q A_{i_j} - C + RI \right) / 2R \quad \overline{M}^t \approx M^t$$

$$(i_1, \dots, i_Q) \sim y^t / \|y^t\|_1, \quad Q = O(\log(m))$$

3. Quantum Gibbs Sampling + amplitude amplification to prepare

$$\overline{\rho}^t = \exp \left( -\varepsilon' \sum_{\tau=1}^t \overline{M}^\tau \right) / \text{tr}(\dots) \quad \overline{\rho}^t \approx \rho^t$$

in time  $\tilde{O}(s^2 n^{1/2})$ .

# Quantizing Arora-Kale Algorithm

We make it quantum as follows:

1. Implement ORACLE by Gibbs Sampling to produce  $y^t$  and apply amplitude amplification to solve it in time  $\tilde{O}(s^2 n^{1/2} m^{1/2})$

We'll show there is a feasible  $y^t$  of the form  $y^t = Nq^t$  with  $q^t := \exp(h)/\text{tr}(\exp(h))$  and

$$h = \sum_{i=1}^m (\lambda \text{tr}(A_i \rho^t) + \mu b_i) |i\rangle\langle i|$$

We need to simulate an oracle to the entries of  $h$ . We do it by measuring  $\rho^t$  with  $A_i$ .

To prepare each  $\rho^t$  takes time  $\tilde{O}(s^2 n^{1/2})$ . To sample from  $q^t$  requires  $\tilde{O}(m^{1/2})$  calls to oracle for  $h$ . So total time is  $\tilde{O}(s^2 n^{1/2} m^{1/2})$

# Quantizing Arora-Kale Algorithm

We make it quantum as follows:

1. Implement ORACLE by Gibbs Sampling to produce  $y^t$  and apply amplitude amplification to solve it in time  $\tilde{O}(s^2 n^{1/2} m^{1/2})$
2. Sparsify  $M^t$  to be a sum of  $O(\log(m))$  terms:

$$\overline{M}^t = \left( \|y^t\|_1 Q^{-1} \sum_{j=1}^Q A_{i_j} - C + RI \right) / 2R \quad \overline{M}^t \approx M^t$$

$$(i_1, \dots, i_Q) \sim y^t / \|y^t\|_1, \quad Q = O(\log(m))$$

Can show it works by Matrix Hoeffding bound:  $Z_1, \dots, Z_k$  independent  $n \times n$  Hermitian matrices s.t.  $E(Z_i)=0$ ,  $\|Z_i\| < \lambda$ . Then

$$\Pr \left( \left\| \frac{1}{k} \sum_{i=1}^k Z_i \right\| \geq \varepsilon \right) \leq n \cdot \exp \left( -\frac{k\varepsilon^2}{8\lambda^2} \right)$$

# Quantum Arora-Kale, Roughly

Let  $\rho^1 = I/n$ ,  $\varepsilon = \frac{\delta\alpha}{2\omega R}$ ,  $\varepsilon' = -\ln(1 - \varepsilon)$ ,  $T = \frac{8\omega^2 R^2 \ln(n)}{\delta^2 \alpha^2}$

For  $t = 1, \dots, T$

1.  $y^t \leftarrow \text{ORACLE}(\rho^t)$

Gibbs Sampling

2.  $M^t = \sum_{j=1}^m (y_j^t A_j - C + \omega I) / 2\omega$

3. Sparsify  $M^t$  to  $(M')^t$

4.  $\rho^{t+1} = \exp\left(-\varepsilon' \left(\sum_{\tau=1}^t (M')^\tau\right)\right) / \text{tr}(\dots)$

Gibbs Sampling

Output:  $\bar{y} = \frac{\delta\alpha}{R} e_1 + \frac{1}{T} \sum_{t=1}^T y^t$

# Implementing Oracle by Gibbs Sampling

ORACLE( $\rho$ )

Searches for a vector  $y$  s.t.

i)  $y \in D_\alpha := \{y : y \geq 0, b \cdot y \leq \alpha\}$

ii)  $\sum_{j=1}^m \text{tr}(A_j \rho) y_j - \text{tr}(C \rho) \geq 0$

# Implementing Oracle by Gibbs Sampling

Searches for (non-normalized) probability distribution  $y$  satisfying two linear constraints:

$$\text{tr}(BY) \leq \alpha, \quad \text{tr}(AY) \geq \text{tr}(C\rho)$$

$$Y = \sum_i y_i |i\rangle\langle i|, \quad B = \sum_i b_i |i\rangle\langle i|, \quad A = \sum_i \text{tr}(A_i \rho) |i\rangle\langle i|$$

**Claim:** We can take  $Y$  to be Gibbs: There are constants  $N, \lambda, \mu$  s.t.

$$Y = N \frac{\exp(\lambda A + \mu B)}{\text{tr}(\dots)}$$

# Jaynes' Principle

(Jaynes 57) Let  $\rho$  be a quantum state s.t.  $\text{tr}(\rho M_i) = c_i$

Then there is a Gibbs state of the form  $\exp\left(\sum_i \lambda_i M_i\right) / \text{tr}(\dots)$

with same expectation values.

**Drawback:** no control over size of the  $\lambda_i$ 's.

# Finitary Jaynes' Principle

(Lee, Raghavendra, Steurer '15) Let  $\rho$  s.t.  $\text{tr}(\rho M_i) = c_i$

Then there is a  $\sigma := \frac{\exp(\sum_i \lambda_i M_i)}{\text{tr}(\dots)}$

with  $|\lambda_i| \leq 2 \ln(\dim(\rho)) / \varepsilon$

s.t.  $|\text{tr}(M_i \sigma) - c_i| \leq \varepsilon$

(Note: Used to prove limitations of SDPs for approximating constraints satisfaction problems; [see James Lee's talk](#))

# Implementing Oracle by Gibbs Sampling

**Claim** There is a  $Y$  of the form  $Y = N \frac{\exp(\lambda A + \mu B)}{\text{tr}(\dots)}$

with  $\lambda, \mu < \log(n)/\varepsilon$  and  $N < \alpha$  s.t.

$$\text{tr}(BY) \leq \alpha + N\varepsilon, \quad \text{tr}(AY) \geq \text{tr}(C\rho) - N\varepsilon$$

$$Y = \sum_i y_i |i\rangle\langle i|, \quad B = \sum_i b_i |i\rangle\langle i|, \quad A = \sum_i \text{tr}(A_i \rho) |i\rangle\langle i|$$

# Implementing Oracle by Gibbs Sampling

**Claim** There is a  $Y$  of the form  $Y = N \frac{\exp(\lambda A + \mu B)}{\text{tr}(\dots)}$

with  $\lambda, \mu < \log(n)/\varepsilon$  and  $N < \alpha$  s.t.

$$\text{tr}(BY) \leq \alpha + N\varepsilon, \quad \text{tr}(AY) \geq \text{tr}(C\rho) - N\varepsilon$$

Can implement oracle by exhaustive searching over  $x, y, N$  for a Gibbs distribution satisfying constraints above

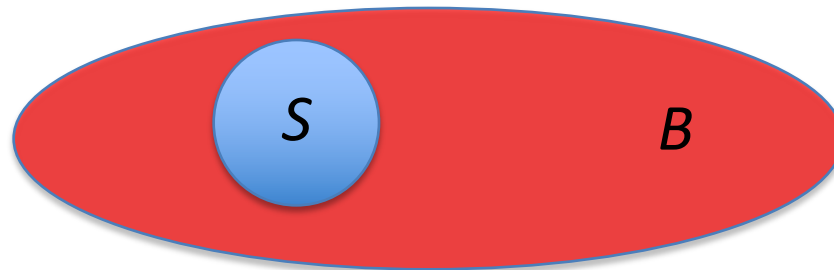
(only  $\alpha \log^2(n)/\varepsilon^3$  different triples needed to be checked)

# Part 2

**Can we prepare thermal states?**  $\rho_\beta = e^{-\beta H} / Z(\beta)$

(Terhal and diVincenzo '00, ...)

Easy? Couple to a bath of the right temperature and wait.



But size of environment might be huge. Maybe not efficient

**Warning:** In general it's impossible to prepare thermal states efficiently, even at *constant* temperature and of classical models, but defined on general graphs (PCP Theorem, Arora *et al* '98)

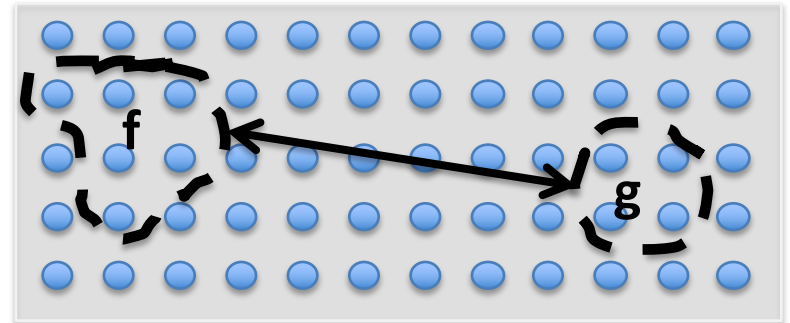
**Warning 2:** Spin glasses not expected to thermalize.

# Classical Result

(Stroock, Zergalinski '92; Martinelli, Olivieri '94, ...) Classically, Gibbs state with finite correlation length can be efficiently simulated with Metropolis

The Gibbs state has correlation length  $\xi$  if for every  $f, g$

$$\begin{aligned} & \sup_{\tau \in B} |\mu_V^\tau(fg) - \mu_V^\tau(f)\mu_V^\tau(g)| \\ & \leq \|f\|_\infty \|g\|_\infty e^{-\text{dist}(f,g)/\xi} \end{aligned}$$



# Preparing Quantum Gibbs States Efficiently

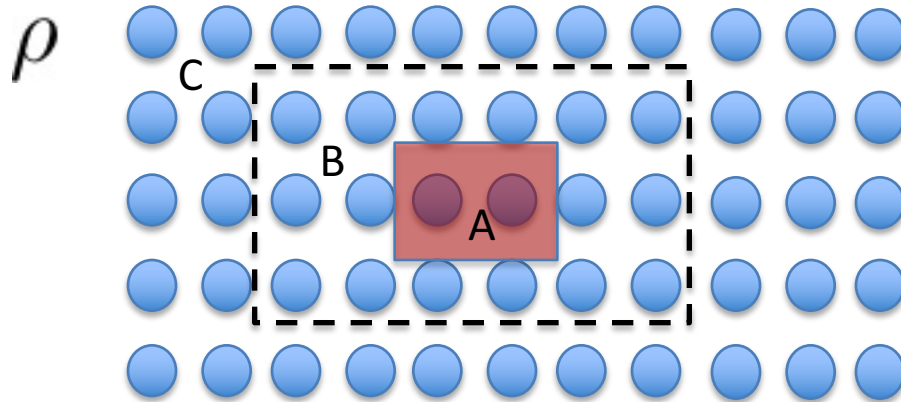
We **conjecture** the same holds for quantum systems.

Partial result

**Thm** (B. Kastoryano '16) For a Hamiltonian  $H$  on a  $d$ -dimensional lattice with  $N$  qubits,  $\rho_T$  can be prepared on a quantum computer in time  $\exp(\log^d(N))$  if

- (i)  $\rho_T$  has finite correlation length
- (ii)  $\rho_T$  is an approximate Markov State

# Q. Approximate Markov States



$\rho$  quantum approximate Markov if for every A, B, C

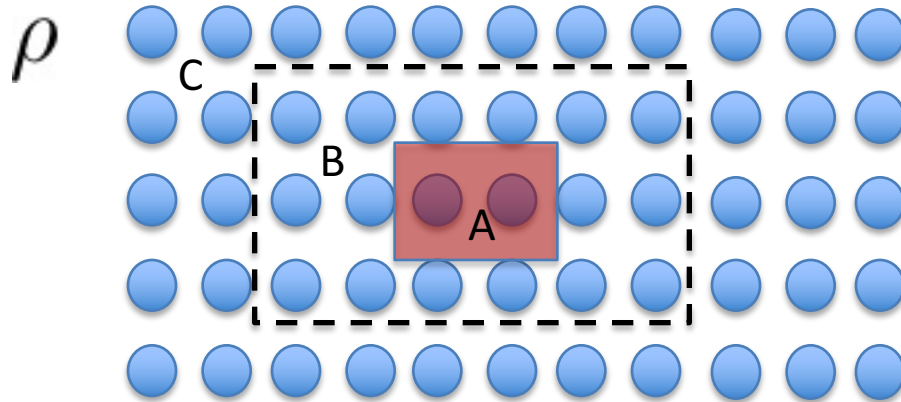
$$I(A : C|B) \leq \exp(-\text{dist}(A, C)^\gamma)$$

- Every Gibbs state of a commuting Hamiltonian is a Markov state

$$H = \sum_k H_k, \quad [H_k, H'_k] = 0$$

- (B., Kato '16) Conjecture true in 1 dim
- Conjecture strengthening of area law for thermal states

# Strengthening of Area Law



(Wolf, Verstraete, Hastings, Cirac '07) 
$$I(A : BC)_{\rho_T} \leq \frac{c}{T} |\partial A|$$

From conjecture:

$$I(A : BC) = I(A : B) + I(A : C|B) \approx I(A : B)$$

Gives rate of saturation of area law

# Conclusion and Open Problems

The theory of quantum Gibbs state is at least as rich as for classical systems. Still at the beginning...

# Conclusion and Open Problems

The theory of quantum Gibbs state is at least as rich as for classical systems. Still at the beginning...

- Can we improve the parameters of the SDP q. algorithm?
- Q. computer only used for Gibbs Sampling. Application of small-sized q. computer?
- Are there more meaningful exponential speed-ups?
- Are quantum Gibbs states approximate Markov?
- Is there a local version of quantum metropolis?  
(Yes if Gibbs state is Markov)

**Thanks!**