Quantum Bootstrapping

Christopher Granade

Joint work with:
Nathan Wiebe D. G. Cory

Institute for Quantum Computing
University of Waterloo, Ontario, Canada

6 January 2015

arXiv:1409.1524

www.cgranade.com/research/talks/iqc/01-06-2015/ • *10/abc* → *doi.org/abc*

We want to build a quantum computer.

We want to build a quantum computer.

Need to push past what a classical computer can do. How do we get to 50 qubits?

Building Large Systems: Computational Limits

Computational limits affect many aspects of building large quantum systems:

- Characterization of H

Building Large Systems: Computational Limits

Computational limits affect many aspects of building large quantum systems:

- Characterization of H
- Calibration of controls

Building Large Systems: Computational Limits

Computational limits affect many aspects of building large quantum systems:

- Characterization of H
- Calibration of controls
- Verification of controls

Bootstrapping to 50 Qubits

Express challenges in terms of *simulation*, then use quantum simulators.

Use small quantum simulators to characterize and verify large devices, bootstrap up to 50-qubit scale.

Bootstrapping to 50 Qubits

Express challenges in terms of *simulation*, then use quantum simulators.

Use small quantum simulators to characterize and verify large devices, bootstrap up to 50-qubit scale.

Outline

- Particle filters: platform for Bayesian inference

Outline

- Particle filters: platform for Bayesian inference
- Hamiltonian learning w/ quantum resources

Outline

- Particle filters: platform for Bayesian inference
- Hamiltonian learning w/ quantum resources
- Bootstrapping Hamiltonian learning

Outline

- Particle filters: platform for Bayesian inference
- Hamiltonian learning w/ quantum resources
- Bootstrapping Hamiltonian learning
- Learning control distortions

Modeling Experiments

Likelihood Function

Model data collection as a probability distribution:

$$\Pr(\text{data}|\text{model}; \text{experiment})$$

Modeling Experiments

Likelihood Function

Model data collection as a probability distribution:

$$\Pr(\text{data}|\text{model}; \text{experiment})$$

The likelihood function *describes* an experiment and its possible outcomes.

Born's Rule: Quintessential Likelihood

Can interpret Born's Rule as the likelihood for state-learning experiments:

$$\Pr(\text{click}|\psi;\phi) = |\langle\phi|\psi\rangle|^2$$

Born's Rule: Quintessential Likelihood

Can interpret Born's Rule as the likelihood for state-learning experiments:

$$\Pr(\text{click}|\psi;\phi) = |\langle\phi|\psi\rangle|^2$$

data	click or no click
model	preparation $\|\psi\rangle$
experiment	measurement $\langle\phi\|$

Hamiltonian Learning Likelihood

Consider Larmor precession at an unknown ω and T_2:

$$H(\omega) = \frac{\omega}{2}\sigma_z, \quad |\psi_{\text{in}}\rangle = |+\rangle, \quad M = \{|+\rangle\langle+|, |-\rangle\langle-|\}$$

$$\Pr(d = 0|\text{model} = (\omega, T_2); \exp = t) = \frac{1 - e^{-t/T_2}}{2} + e^{-t/T_2}\cos^2(\omega t/2)$$

Hamiltonian Learning Likelihood

Consider Larmor precession at an unknown ω and T_2:

$$H(\omega) = \frac{\omega}{2}\sigma_z, \quad |\psi_{\text{in}}\rangle = |+\rangle, \quad M = \{|+\rangle\langle+|, |-\rangle\langle-|\}$$

$$\Pr(d = 0|\text{model} = (\omega, T_2); \exp = t) = \frac{1 - e^{-t/T_2}}{2} + e^{-t/T_2}\cos^2(\omega t/2)$$

Parameterize model as $\underline{x} = (\omega, T_2)$, experiment as $\underline{e} = (t)$.

Updating Knowledge

Once we have a likelihood, we can now reason about

$$\Pr(\underline{x}|d, \underline{e}),$$

what we know having seen some data.

Updating Knowledge

Once we have a likelihood, we can now reason about

$$\Pr(\underline{x}|d, \underline{e}),$$

what we know having seen some data.

By Bayes' Rule: $\Pr(\underline{x}|d, \underline{e}) = \frac{\Pr(d|\underline{x};\underline{e})}{\Pr(d|\underline{e})} \Pr(\underline{x}).$

$\Longrightarrow$ Simulation is a resource for learning.

Updating Knowledge

Once we have a likelihood, we can now reason about

$$\Pr(\underline{x}|d, \underline{e}),$$

what we know having seen some data.

By Bayes' Rule: $\Pr(\underline{x}|d, \underline{e}) = \frac{\Pr(d|\underline{x};\underline{e})}{\Pr(d|\underline{e})} \Pr(\underline{x}).$

$\Longrightarrow$ Simulation is a resource for learning.

Updating Knowledge

Once we have a likelihood, we can now reason about

$$\Pr(\underline{x}|d, \underline{e}),$$

what we know having seen some data.

By Bayes' Rule: $\Pr(\underline{x}|d, \underline{e}) = \frac{\Pr(d|\underline{x};\underline{e})}{\Pr(d|\underline{e})} \Pr(\underline{x}).$

$\Longrightarrow$ Simulation is a resource for learning.

Updating Knowledge

Once we have a likelihood, we can now reason about

$$\Pr(\underline{x}|d, \underline{e}),$$

what we know having seen some data.

By Bayes' Rule: $\Pr(\underline{x}|d, \underline{e}) = \frac{\Pr(d|\underline{x};\underline{e})}{\Pr(d|\underline{e})} \Pr(\underline{x}).$

$\Longrightarrow$ Simulation is a resource for learning.

Estimate $\hat{\underline{x}}$ as the expectation over $\underline{x}$,

$$\hat{\underline{x}} = \mathbb{E}[\underline{x}] = \int \underline{x} \Pr(\underline{x}) \, \mathrm{d}\underline{x}.$$

Updating Knowledge

Once we have a likelihood, we can now reason about

$$\Pr(\underline{x}|d, \underline{e}),$$

what we know having seen some data.

By Bayes' Rule: $\Pr(\underline{x}|d, \underline{e}) = \frac{\Pr(d|\underline{x};\underline{e})}{\Pr(d|\underline{e})} \Pr(\underline{x})$.
$\implies$ Simulation is a resource for learning.

Estimate $\hat{\underline{x}}$ as the expectation over $\underline{x}$,

$$\hat{\underline{x}} = \mathbb{E}[\underline{x}] = \int \underline{x} \Pr(\underline{x}) \, \mathrm{d}\underline{x}.$$

In many cases, difficult to perform analytically...

Sequential Monte Carlo

SMC (aka *particle filter*)**:** numerical algorithm for generating samples from a distribution, using a transition kernel.

$$\text{prior} \xrightarrow{\text{Bayes' Rule}} \text{posterior}$$

Posterior samples then approximate $\int / \mathbb{E}$.

SMC Approximation

$$\Pr(\underline{x}) \approx \sum_{i}^{n} w_i \delta(\underline{x} - \underline{x}_i)$$

(Doucet and Johansen 2011; Huszár and Houlsby *10/s86*; *Granade* et al. 2012 *10/s87*)

Sequential Monte Carlo

SMC (aka *particle filter*)**:** numerical algorithm for generating samples from a distribution, using a transition kernel.

$$\text{prior} \xrightarrow{\text{Bayes' Rule}} \text{posterior}$$

Posterior samples then approximate $\int / \mathbb{E}$.

SMC Approximation

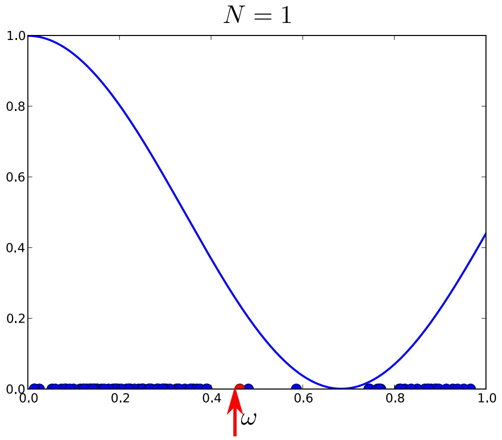
$$\Pr(\underline{x}) \approx \sum_{i}^{n} w_i \delta(\underline{x} - \underline{x}_i)$$

QInfer Open-source implementation for quantum info.

(Doucet and Johansen 2011; Huszár and Houlsby *10/s86*; *Granade* et al. 2012 *10/s87*)

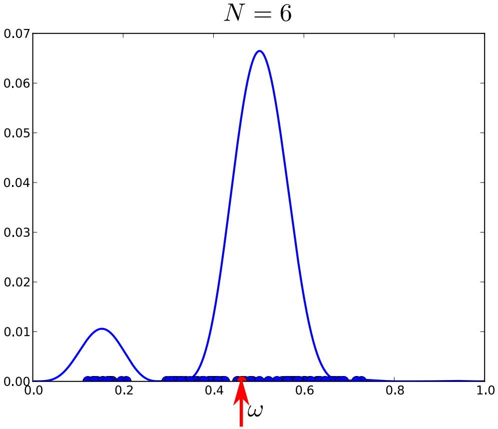
Sequential Monte Carlo

For Larmor precession:



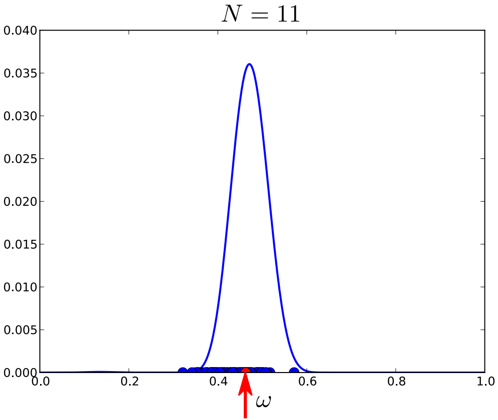
Sequential Monte Carlo

For Larmor precession:



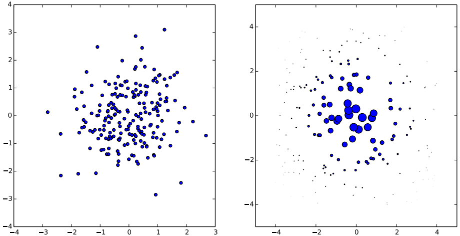
Sequential Monte Carlo

For Larmor precession:



Ambiguity and Impovrishment

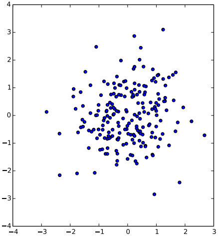
Ambiguity in SMC approximation:



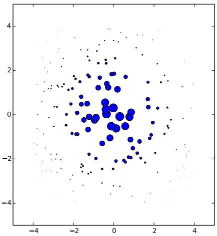
density weight

Ambiguity and Impovrishment

Ambiguity in SMC approximation:



density



weight

Using weight is less numerically stable, results in smaller *effective* number of particles.

$$n_{\text{ess}} := 1/\sum_i w_i^2$$

Numerical Stability and Resampling

As data D is collected, $\Pr(\underline{x}_i|D) \to 0$ for most initial particles $\{x_i\}$.

- $\Rightarrow n_{\text{ess}} \to 0$ as data is collected.

Resampling: move information from weights to the density of SMC particles.

- Resampling when $n_{\text{ess}}/n \leq 0.5$ preserves stability.
- Monitoring n_{ess} can herald some kinds of failures.

Towards Bootstrapping

SMC uses *simulation* as a resource for *learning*.

Simulation calls: main cost to SMC (n each Bayes update).

Towards Bootstrapping

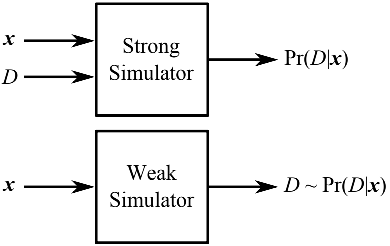
SMC uses *simulation* as a resource for *learning*.

Simulation calls: main cost to SMC (n each Bayes update).

Big Idea

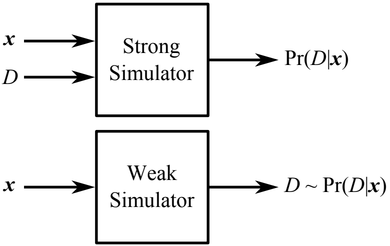
Use quantum simulation to extend SMC past classical resources.

Weak and Strong Simulation



(Ferrie and *Granade* 2014 *10/tdj*)

Weak and Strong Simulation



Quantum simulation produces data, not likelihoods. Must sample to estimate likelihood.

(Ferrie and *Granade* 2014 *10/tdj*)

Adaptive Likelihood Estimation

Solution

Treat estimating the likelihood as a secondary estimation problem:

Learn likelihood of untrusted system from frequencies of trusted system.

Adaptive Likelihood Estimation

Solution

Treat estimating the likelihood as a secondary estimation problem:

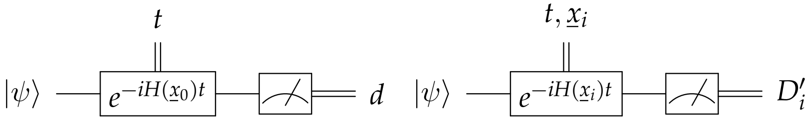
Learn likelihood of untrusted system from frequencies of trusted system.

SMC is robust to likelihood estimation errors.

Quantum Likelihood Evaluation

Compare *classical* outcomes of unknown and trusted systems.

Unknown System Simulator



For each $\underline{x}_i$:

- repeatedly sample from quantum simulation of $\mathrm{e}^{-\mathrm{i}t\underline{x}_i}$, getting D_i'.
- estimate $\hat{\ell}_i$ from D_i'.

SMC update: $w_i \mapsto w_i \hat{\ell}_i / \sum_i w_i \hat{\ell}_i$.

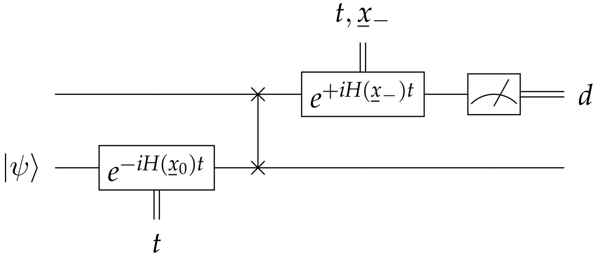
(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

QLE can work, but as $t \to \infty$, $\Pr(d|\underline{x}; t) \rightsquigarrow 1/\dim\mathcal{H}$.
Thus, $t \geq t_{\mathrm{eq}}$ is uninformative.

By the Cramer-Rao Bound, error then scales as $\mathrm{O}(1/Nt_{\mathrm{eq}}^2)$.

Interactive QLE

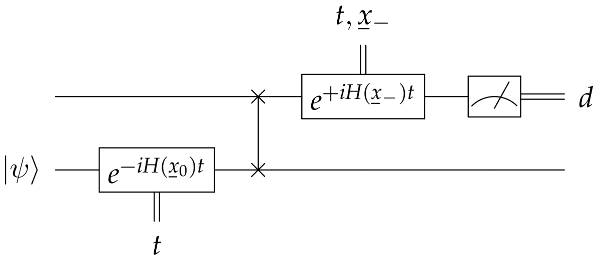
Solution: couple unknown system to a quantum simulator, then invert evolution by hypothesis $\underline{x}_-$.



(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

Interactive QLE

Solution: couple unknown system to a quantum simulator, then invert evolution by hypothesis $\underline{x}_-$.



Echo

If $\underline{x}_- \approx \underline{x}_0$, then $\left|\langle\psi|\mathrm{e}^{-\mathrm{i}t(H(\underline{x}_0)-H(\underline{x}_-))}|\psi\rangle\right|^2 \approx 1$.

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

Particle Guess Heuristic

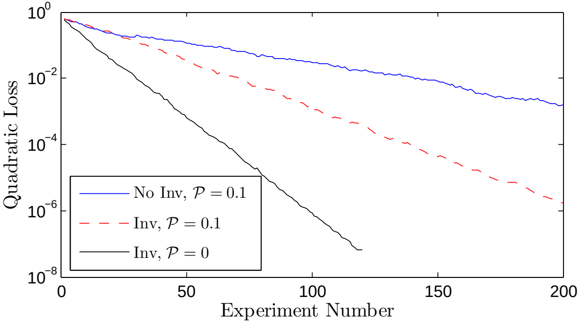
Inversion connects the model and experiment spaces.
Use this duality as a heuristic for experiment design.

- Choose $\underline{x}_-, \underline{x}'_- \sim \Pr(\underline{x})$, the most recent posterior.
- Choose $t = 1/\|\underline{x}_- - \underline{x}'_-\|$.
- Return $\underline{e} = (\underline{x}_-, t)$.

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

Ising Model on Spin Chains

Hamiltonian: nearest-neighbor Ising models on a chain of nine qubits.
Interactivity allows for dramatic improvements over QLE.

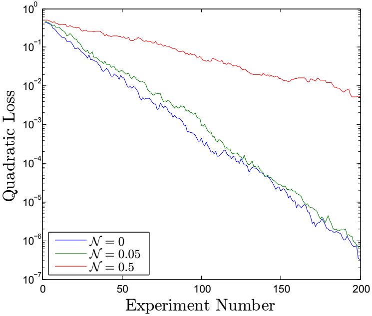


$\mathcal{P}$: adaptive likelihood estimation tolerance.

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

Ising Model on the Complete Graph

With IQLE, can also learn on complete interaction graphs. We show the performance as a function of the depolarization strength $\mathcal{N}$.

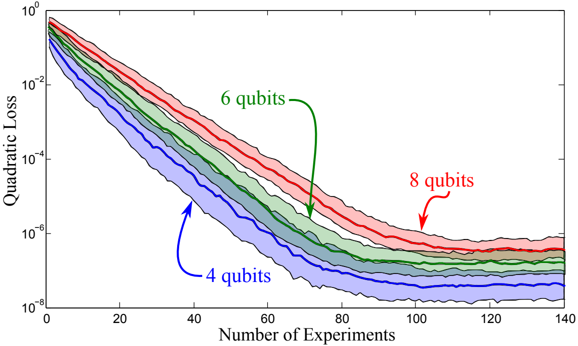


$\mathcal{N}$: depolarizing noise following SWAP gate.

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tdk*)

Ising Model with the Wrong Graph

Simulate with spin chains, suppose "true" system is complete, with non-NN couplings $O(10^{-4})$.



(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tdk*)

Scaling Parameter

dim $\underline{x}$, not dim $\mathcal{H}$, determines scaling of IQLE.

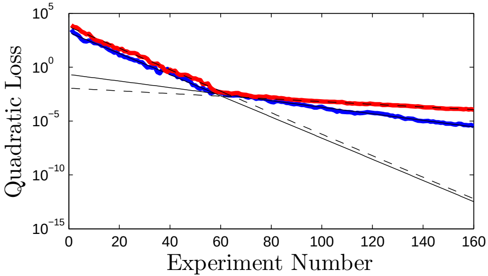


Figure: 4 qubit (red) and 6 qubit (blue) complete graph IQLE

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

Scaling and Dimensionality

In spin-chain and complete graph, average error decays exponentially,

$$L(N) \propto e^{-\gamma N}$$

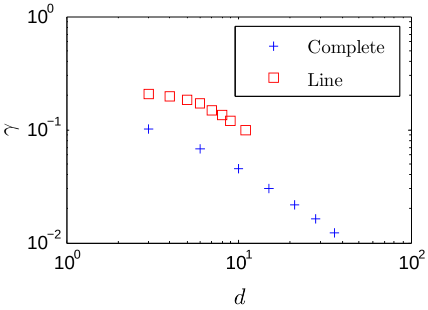
(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

Scaling and Dimensionality

In spin-chain and complete graph, average error decays exponentially,

$$L(N) \propto e^{-\gamma N}$$

Assess scaling by finding $\gamma = \gamma(\dim \underline{x})$:



With quantum simulation, learning *may* scale efficiently.

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tf3*)

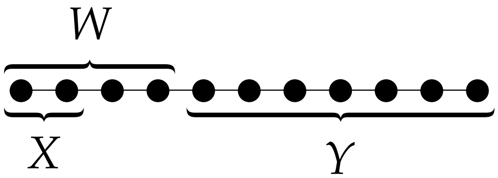
SMC + IQLE:

- Possibly scalable with quantum resources.
- Robust to finite sampling.
- Robust to approximate models.

Still requires simulator be at least as large as system of interest.

Information Locality

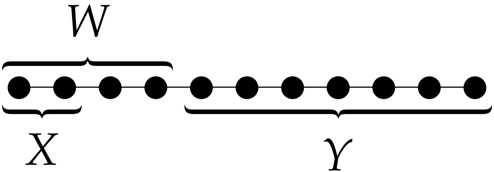
To go further, we want to *localize* our experiment, such that we can simulate on a smaller system.



Measure on X, simulate on W, and ignore all terms with support over Y.

Information Locality

To go further, we want to *localize* our experiment, such that we can simulate on a smaller system.

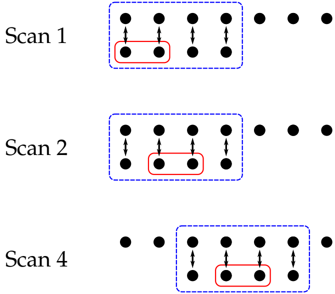


Measure on X, simulate on W, and ignore all terms with support over Y.

Gives *approximate* model that can be used to learn Hamiltonian restricted to X.

Local and Global Particle Clouds

To reconstruct the entire system, we need to combine data from different partitions.



Separate out one partition at a time, maintain a *global* cloud of particles.

Local and Global Particle Clouds

Initialize $\{\underline{x}_i\}$ over entire system. Then, for each simulated subregister W_k:

1. Make “local” particle cloud $\{\underline{x}_i|_{W_k}\}$ by slicing $\{\underline{x}_i\}$.
2. Run SMC+IQLE with $\{\underline{x}_i|_{W_k}\}$ as a prior.
3. Ensure that the final “local” cloud has been resampled (has equal weights).
4. Overwrite parameters in “global” cloud $\{\underline{x}_i\}$ corresponding to post-resampling $\{\underline{x}_i|_{W_k}\}$.

In this way, all parameters are updated by an SMC run.

Q50 Example

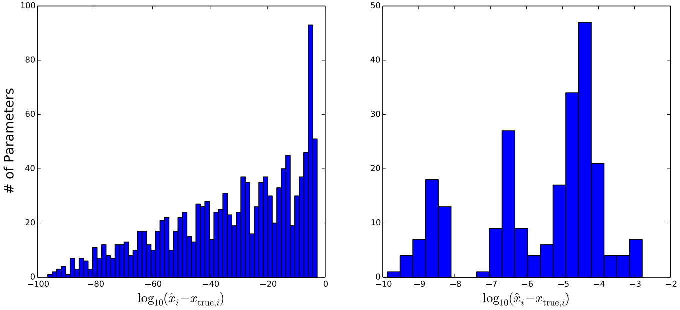
Goal: characterize a 50-qubit Ising model (complete graph) with unknown ZZ couplings.

All Hamiltonian terms commute, but initial state doesn't. Let A_X be observable, $A_{X'}$ be sim. observable.

$$\begin{aligned}\|A_X(t) - A_{X'}(t)\| &\le \|A_X(t)\|(\mathrm{e}^{2\|H|_Y\|t} - 1)\\ \Rightarrow t &\le \ln\left(\frac{\delta}{\|A_X(t)\|} + 1\right)(2\|H|_Y\|)^{-1},\end{aligned}$$

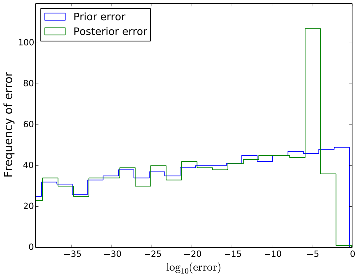
where δ is the tolerable likelihood error.

Example Q50 Run



$|X_k| = 4$, $|W_k| = 8$, $n = 20,000$, $N = 500$, exp. decaying interactions.
NB: 1225 parameter model, L_2 error of 0.3%.

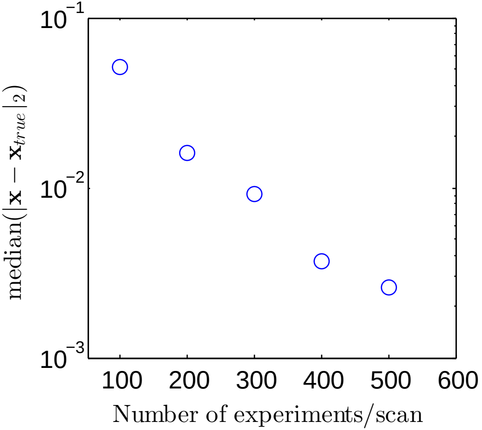
Example Q50 Run



$|X_k| = 4$, $|W_k| = 8$, $n = 20,000$, $N = 500$, exp. decaying interactions.
NB: 1225 parameter model, L_2 error of 0.3%.

Scaling With N

We expect from uncompressed quantum Hamiltonian learning that the error decays exponentially with more data. This remains the case even with compression.



Lieb-Robinson Bounds

More generally, for $[H|_W, H_Y] \neq 0$, use *Lieb-Robinson bound*. If interactions between X and Y decay sufficiently quickly, then there exists C, μ and v s. t. for any observables $A_X(t)$, B_Y:

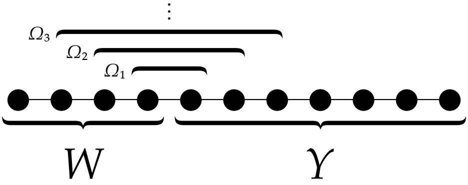
$$\|[A_X(t), B_Y]\| \leq C\|A_X(t)\|\|B_Y\||X||Y|(\mathrm{e}^{v|t|} - 1)\mathrm{e}^{-\mu d(X,Y)}$$

This *guarantees* that error due to truncation is bounded if we choose small t.

(Hastings and Koma 2006 *10/cddqgz*; Nachtergale and Sims 2006 *10/d9xwfg*)

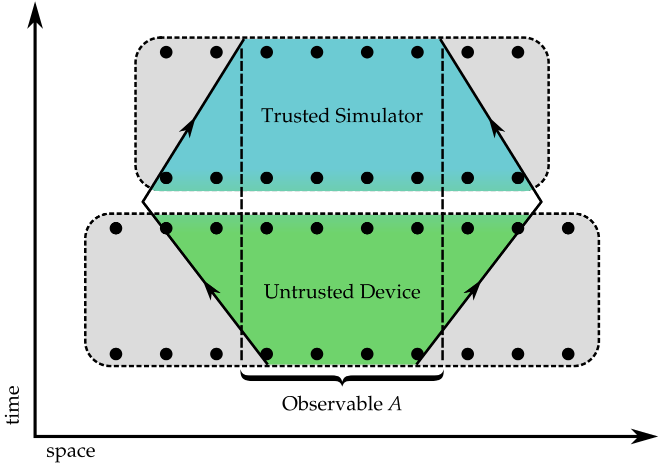
Lieb-Robinson Bounds

Can find bound in terms of Hamiltonian by considering H site-by-site.



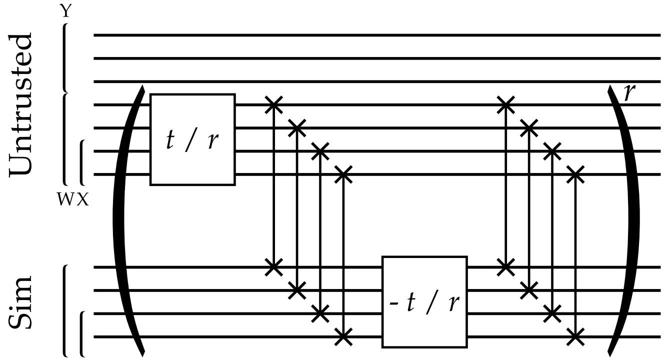
Let H_j be the Hamiltonian term containing distance-j interactions between W and Y, acting on sites Ω_j.

$$\|A(t) - e^{iH|_W t} A e^{-iH|_W t}\| \leq \sum_j C\|A\|\|H_j\|\|X\||\Omega_j| e^{-\mu j}(e^{v|t|} - 1)$$


Trusted Simulator
Untrusted Device
Observable A
time
space

"Shaking"

Can improve the Lieb-Robinson bound by alternating between simulator and system. Using $r \approx vt$ SWAP gates, error is $O(t)$.



Bootstrapping Algorithm

Consider H an affine map $H(\underline{C})$ of control settings $\underline{C}$:

$$H(\underline{C}) = \underline{C} \cdot (H_1, H_2, \ldots, H_M) + H_0. \tag{1}$$

E.g.: cross-talk.
We can learn this with compressed IQLE:

- Learn $H(\underline{0})$ to estimate $\hat{H}_0$.
- Learn $H(\underline{e}_j)$ for $j \in \{1, \ldots, M\}$.
- Subtract $\hat{H}_0$ from each of the learned Hamiltonians to estimate the other terms.
- Use the pseudoinverse to derive control settings to generate desired Hamiltonians.

Example: Controlling NN Ising Couplings

Consider $H(\underline{C})$ such that C_i nominally controls the coupling $H_i = \sigma_z^{(i)} \sigma_z^{(i+1)}$. For a 50-qubit device, $\dim \underline{C} = 49$, so this is a $(49+1) \times 1225 \approx 61 \times 10^3$ parameter model.

We collect 200 bits of data per scan, for a total of $50 \times 49 \times 200 = 490 \times 10^3$ bits of data. We use 20×10^3 particles, for a total of 10 million likelihood calls.

Results for Bootstrapping 50-Qubit Simulator

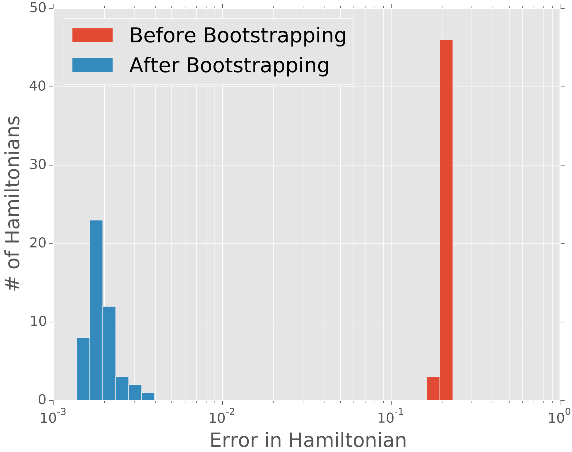


Figure: Frequencies of error $\|H(\hat{\underline{C}}_i) - H_i\|_2$ for Q50 bootstrapping.

- Bayesian inference: simulation as a characterization/validation resource.

- Bayesian inference: simulation as a characterization/validation resource.
- Sequential Monte Carlo: numerical algorithm for inference.

- Bayesian inference: simulation as a characterization/validation resource.
- Sequential Monte Carlo: numerical algorithm for inference.
- Robust to many practical concerns.

- Bayesian inference: simulation as a characterization/validation resource.
- Sequential Monte Carlo: numerical algorithm for inference.
- Robust to many practical concerns.
- Can use quantum simulation to offer potential scaling.

- Bayesian inference: simulation as a characterization/validation resource.
- Sequential Monte Carlo: numerical algorithm for inference.
- Robust to many practical concerns.
- Can use quantum simulation to offer potential scaling.
- Using robustness of SMC, can truncate simulation $\rightarrow$ bootstrapping.

Further Information

Slides, a journal reference for this work, a full bibliography and a software implementation can be found at *http://www.cgranade.com/research/talks/iqc/01-06-2015/*.

Thank you for your kind attention!

Decision Theory

A few definitions help us evaluate estimates $\hat{\underline{x}}$ of $\underline{x}$:

Loss How well have we learned?
$L_{\underline{\underline{Q}}}(\hat{\underline{x}}, \underline{x}) := (\hat{\underline{x}} - \underline{x})^{\mathrm{T}} \underline{\underline{Q}} (\hat{\underline{x}} - \underline{x})$

Decision Theory

A few definitions help us evaluate estimates $\underline{\hat{x}}$ of $\underline{x}$:

- **Loss** How well have we learned?
  $L_{\underline{\underline{Q}}}(\underline{\hat{x}}, \underline{x}) := (\underline{\hat{x}} - \underline{x})^{\mathrm{T}} \underline{\underline{Q}} (\underline{\hat{x}} - \underline{x})$
- **Risk** On average, how well will we learn a particular model?
  $R(\underline{\hat{x}}, \underline{x}) := \mathbb{E}_D[L(\underline{\hat{x}}(D), \underline{x})]$

Decision Theory

A few definitions help us evaluate estimates $\underline{\hat{x}}$ of $\underline{x}$:

Loss How well have we learned?
$L_{\underline{\underline{Q}}}(\underline{\hat{x}}, \underline{x}) := (\underline{\hat{x}} - \underline{x})^{\mathrm{T}} \underline{\underline{Q}} (\underline{\hat{x}} - \underline{x})$

Risk On average, how well will we learn a particular model?
$R(\underline{\hat{x}}, \underline{x}) := \mathbb{E}_D[L(\underline{\hat{x}}(D), \underline{x})]$

Bayes risk On average, how well will we learn a range of models?
$r(\underline{\hat{x}}, \pi) = \mathbb{E}_{\underline{x} \sim \pi}[R(\underline{\hat{x}}, \underline{x})]$

Decision Theory

A few definitions help us evaluate estimates $\underline{\hat{x}}$ of $\underline{x}$:

Loss How well have we learned?
$L_{\underline{\underline{Q}}}(\underline{\hat{x}}, \underline{x}) := (\underline{\hat{x}} - \underline{x})^{\mathrm{T}} \underline{\underline{Q}} (\underline{\hat{x}} - \underline{x})$

Risk On average, how well will we learn a particular model?
$R(\underline{\hat{x}}, \underline{x}) := \mathbb{E}_D[L(\underline{\hat{x}}(D), \underline{x})]$

Bayes risk On average, how well will we learn a range of models?
$r(\underline{\hat{x}}, \pi) = \mathbb{E}_{\underline{x} \sim \pi}[R(\underline{\hat{x}}, \underline{x})]$

Cramér-Rao Bound On average, how well *can* we learn?

Cramér-Rao Bound

Fisher Information

How much information about $\underline{x}$ is obtained by sampling data?

$$\underline{\underline{I}}(\underline{x}) = \mathbb{E}_D[(\underline{\nabla}_{\underline{x}} \log \Pr(D|\underline{x}))(\underline{\nabla}_{\underline{x}} \log \Pr(D|\underline{x}))^{\mathrm{T}}]$$

(Ferrie, *Granade* and Cory 2013 *10/tfx*)

Cramér-Rao Bound

Fisher Information

How much information about $\underline{x}$ is obtained by sampling data?

$$\underline{\underline{I}}(\underline{x}) = \mathbb{E}_D[(\underline{\nabla}_{\underline{x}} \log \Pr(D|\underline{x}))(\underline{\nabla}_{\underline{x}} \log \Pr(D|\underline{x}))^{\mathrm{T}}]$$

The Cramér-Rao Bound tells how well any unbiased estimator can do. If $\underline{\underline{Q}} = \mathbb{1}$, then

$$R(\hat{\underline{x}}, \underline{x}) = \mathrm{Tr}(\mathrm{Cov}(\hat{\underline{x}})) \geq \mathrm{Tr}(\underline{\underline{I}}(\underline{x})^{-1}).$$

(Ferrie, *Granade* and Cory 2013 *10/tfx*)

Bayesian Cramér-Rao Bound

Expectation of Fisher information over prior π: the *Bayesian* Cramér-Rao bound.

$$\underline{\underline{B}} := \mathbb{E}_{\underline{x}\sim\pi}[\underline{\underline{I}}(\underline{x})], \quad r(\pi) \geq \underline{\underline{B}}^{-1}$$

For adaptive experiments, the posterior is used instead of the prior.

The BCRB can be computed iteratively: useful for tracking optimality online.

$$\underline{\underline{B}}_{k+1} = \underline{\underline{B}}_k + \begin{cases} \mathbb{E}_{\underline{x}\sim\pi}[\underline{\underline{I}}(\underline{x}; \underline{e}_{k+1})] & \text{(non-adaptive)} \\ \mathbb{E}_{\underline{x}|d_1,\dots,d_k}[\underline{\underline{I}}(\underline{x}; \underline{e}_{k+1})] & \text{(adaptive)} \end{cases}$$

(Gill and Levit 1995; Ferrie, *Granade* et al. 2012 *10/s87*)

Liu and West Algorithm

Draw new particles $\underline{x}'$ from kernel density estimate:

$$\Pr(\underline{x}') \propto \sum_i w_i \exp\left((\underline{x}' - \underline{\mu}_i)^{\mathrm{T}} \underline{\underline{\Sigma}} (\underline{x}' - \underline{\mu}_i)\right)$$

$$\underline{\mu}_i := a\underline{x}_i + (1-a)\mathbb{E}[\underline{x}] \qquad \underline{\underline{\Sigma}} := h^2 \operatorname{Cov}[\underline{x}] \qquad w_i' := 1/n$$

(West 1993; Isard and Blake 1998 *10/cc76f6*; Liu and West 2001)

Liu and West Algorithm

Draw new particles $\underline{x}'$ from kernel density estimate:

$$\Pr(\underline{x}') \propto \sum_i w_i \exp\left((\underline{x}' - \underline{\mu}_i)^{\mathrm{T}} \underline{\underline{\Sigma}} (\underline{x}' - \underline{\mu}_i) \right)$$

$$\underline{\mu}_i := a\underline{x}_i + (1-a)\mathbb{E}[\underline{x}] \qquad \underline{\underline{\Sigma}} := h^2 \operatorname{Cov}[\underline{x}] \qquad w_i' := 1/n$$

Parameters a and h can be set based on application:

- $a = 1, h = 0$: Bootstrap filter, used in state-space applications like CONDENSATION.
- $a^2 + h^2 = 1$: Ensures $\mathbb{E}[\underline{x}'] = \mathbb{E}[\underline{x}]$ and $\operatorname{Cov}(\underline{x}') = \operatorname{Cov}(\underline{x})$, but assumes unimodality.
- $a = 1, h \geq 0$: Allows for multimodality, emulating state-space with synthesized noise.

(West 1993; Isard and Blake 1998 *10/cc76f6*; Liu and West 2001)

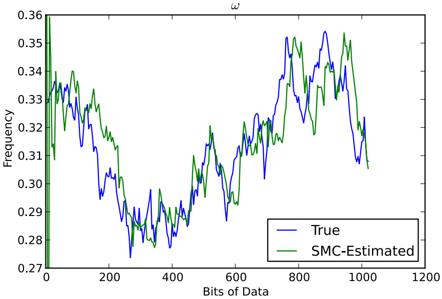
Putting it All Together: The SMC Algorithm

1. Draw $\{\underline{x}_i\} \sim \pi$, set $\{w_i\} = 1/n$.
2. For each datum $d_j \in D$:
   1. $w_i \leftarrow w_i \times \Pr(d_j | \underline{x}_i; \underline{e}_j)$.
   2. Renormalize $\{w_i\}$.
   3. If $n_{\text{ess}}/n \leq 0.5$, resample.
3. Report $\hat{\underline{x}} := \mathbb{E}[\underline{x}] \approx \sum_i w_i \underline{x}_i$.

State-Space SMC

Can move particles at each timestep $\underline{x}(t_k) \sim \Pr(\underline{x}(t_k)|\underline{x}(t_{k-1}))$.

This represents *tracking* of a stochastic process.



Confidence and Credible Regions

Characterizing uncertainty of estimates is critical for many applications:

Definition (Confidence Region)

X_α is an α-confidence region if $\Pr_D(\underline{x}_0 \in X_\alpha(D)) \geq \alpha$.

(*Granade* et al. 2012 *10/s87*; Ferrie 2014 *10/tb4*)

Confidence and Credible Regions

Characterizing uncertainty of estimates is critical for many applications:

Definition (Credible Region)

X_α is an α-credible region if $\Pr_{\underline{x}}(\underline{x} \in X_\alpha | D) \geq \alpha$.

(*Granade* et al. 2012 *10/s87*; Ferrie 2014 *10/tb4*)

Confidence and Credible Regions

Characterizing uncertainty of estimates is critical for many applications:

Definition (Credible Region)

X_α is an α-credible region if $\Pr_{\underline{x}}(\underline{x} \in X_\alpha | D) \geq \alpha$.

Credible regions can be calculated from posterior $\Pr(\underline{x}|D)$ by demanding

$$\int_{X_\alpha} \mathrm{d}\Pr(\underline{x}|D) \geq \alpha.$$

(*Granade* et al. 2012 *10/s87*; Ferrie 2014 *10/tb4*)

High Posterior Density

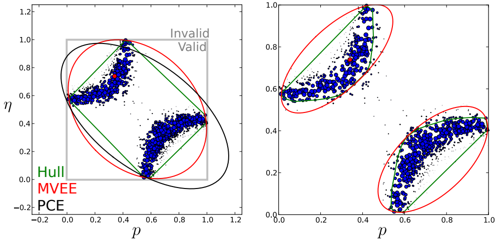
Want credible regions that are *small* (most powerful).

- Posterior covariance ellipses (PCE)— good for approximately normal posteriors
- Convex hull— very general, but verbose description
- Minimum volume enclosing ellipses (MVEE)— good approximation to hull

(*Granade* et al. 2012 *10/s87*; Ferrie 2014 *10/tb4*)

Comparison of HPD Estimators

For multimodal distributions, clustering can be used to exclude regions of small support.
For a noisy coin model (heads probability p, visibility η):



Left, no clustering. Right, DBSCAN.

Plot courtesy of Chris Ferrie. (Ferrie 2014 *10/tb4*)

Bayes Factors and Model Selection

Drunk Under the Streetlights

In SMC update $w_i \mapsto w_i \times \Pr(d|\underline{x};\underline{e})/\mathcal{N}$,

$$\mathcal{N} = \mathcal{N}(d) \approx \Pr(d|\underline{e}).$$

Is this useful?

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tdk*)

Bayes Factors and Model Selection

Drunk Under the Streetlights

In SMC update $w_i \mapsto w_i \times \Pr(d|\underline{x};\underline{e})/\mathcal{N}$,

$$\mathcal{N} = \mathcal{N}(d) \approx \Pr(d|\underline{e}).$$

Is this useful?

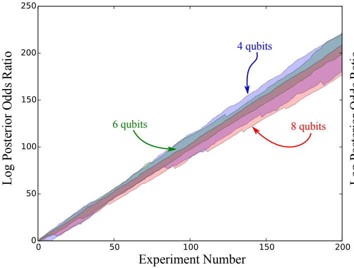
Collecting normalizations $\mathcal{N}_A$ and $\mathcal{N}_B$ for models A, B at each step gives

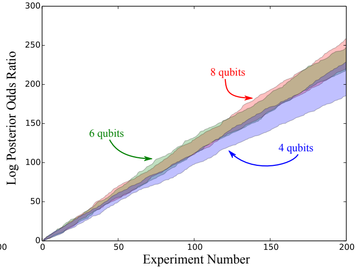
$$\text{Bayes factor} = \frac{\Pr(D|A;\underline{e})\Pr(A)}{\Pr(D|B;\underline{e})\Pr(B)} \approx \frac{\prod_{d\in D}\mathcal{N}_A(d)}{\prod_{d\in D}\mathcal{N}_B(d)} \times \frac{\Pr(A)}{\Pr(B)}$$

For full data record, can multiply normalization records to select A versus B.

(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tdk*)

For example, deciding between linear- (left) and complete-graph (right) Ising models:





(Wiebe, *Granade*, Ferrie and Cory 2014 *10/tdk*)

Method of Hyperparameters

If "true" model $\underline{x} \sim \Pr(\underline{x}|\underline{y})$, for some *hyperparameters* $\underline{y}$, can est. $\underline{y}$ directly:

$$\Pr(d|\underline{y};\underline{e}) = \int \Pr(d|\underline{x},\underline{y};\underline{e}) \Pr(\underline{x}|\underline{y};\underline{e}) \, \mathrm{d}\underline{x}.$$

(*Granade* et al. 2012 *10/s87*)

Method of Hyperparameters

If "true" model $\underline{x} \sim \Pr(\underline{x}|\underline{y})$, for some *hyperparameters* $\underline{y}$, can est. $\underline{y}$ directly:

$$\Pr(d|\underline{y};\underline{e}) = \int \Pr(d|\underline{x},\underline{y};\underline{e}) \Pr(\underline{x}|\underline{y};\underline{e}) \,\mathrm{d}\underline{x}.$$

Example

For Larmor precession with $\omega \sim \mathrm{Cauchy}(\omega_0, T_2^{-1})$,

$$\Pr(d|(\omega_0, T_2^{-1}); t) = \mathrm{e}^{-tT_2^{-1}} \cos^2(\omega_0 t/2) + (1 - \mathrm{e}^{-tT_2^{-1}})/2.$$

Let $\underline{y} = (\omega_0, T_2^{-1})$.

(*Granade* et al. 2012 *10/s87*)

Hyperparameters and Region Estimation

In some hyperparameter models, can also express as region estimator on underlying parameters.

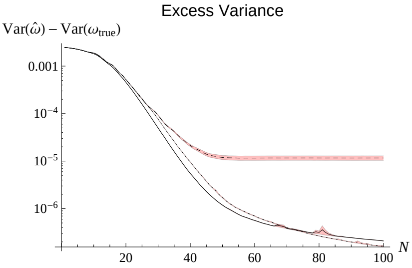


Figure: Larmor precession model w/ $\omega \sim \mathrm{N}(\mu, \sigma^2)$, three exp. design strategies

Critically, the covariance region for ω is not smaller than the true covariance given by the hyperparameter σ^2.

(*Granade* et al. 2012 *10/s87*)