A polynomial-time algorithm for approximating degenerate ground states of gapped spin chains.

Christopher T. Chubb and Steven T. Flammia

ARC CENTRE OF EXCELLENCE FOR
ENGINEERED QUANTUM SYSTEMS

THE UNIVERSITY OF SYDNEY

Australian Institute of Physics Congress
2014

1D Ground state approximation

Problem

Given a spin-system with local interactions, approximate the ground state(s) of that system efficiently on a classical computer.

- This is not possible[1] for a general system[2]. We're going to be considering gapped 1D systems, which serve as a nice toy-model in which it is.
- Local and gapped interactions give us the area law[3], a conjectured structural bound on the complexity of ground states.
- Restricting further to 1D allows us to use a rigorous proof of the area law[4].

[1] Given standard complexity theoretic assumptions, analogous to P $\neq$ NP.

[2] J. Kempe, A. Kitaev and O. Regev, doi:10/dqbscx, arXiv:quant-ph/0406180, 2004.

[3] J. Eisert, M. Cramer and M.B. Plenio, doi:10/bts7tp, arXiv:0808.3773, 2008.

[4] Y. Huang, arXiv:1403.0327, 2014

1D Ground state approximation

Problem

Given a spin-system with local interactions, approximate the ground state(s) of that system efficiently on a classical computer.

- This is not possible[1] for a general system[2]. We're going to be considering gapped 1D systems, which serve as a nice toy-model in which it is.
- Local and gapped interactions give us the area law[3], a conjectured structural bound on the complexity of ground states.
- Restricting further to 1D allows us to use a rigorous proof of the area law[4].

[1] Given standard complexity theoretic assumptions, analogous to P $\neq$ NP.

[2] J. Kempe, A. Kitaev and O. Regev, doi:10/dqbscx, arXiv:quant-ph/0406180, 2004.

[3] J. Eisert, M. Cramer and M.B. Plenio, doi:10/bts7tp, arXiv:0808.3773, 2008.

[4] Y. Huang, arXiv:1403.0327, 2014

1D Ground state approximation

Problem

Given a spin-system with local interactions, approximate the ground state(s) of that system efficiently on a classical computer.

- This is not possible[1] for a general system[2]. We're going to be considering gapped 1D systems, which serve as a nice toy-model in which it is.
- Local and gapped interactions give us the area law[3], a conjectured structural bound on the complexity of ground states.
- Restricting further to 1D allows us to use a rigorous proof of the area law[4].

[1] Given standard complexity theoretic assumptions, analogous to P $\neq$ NP.

[2] J. Kempe, A. Kitaev and O. Regev, doi:10/dqbscx, arXiv:quant-ph/0406180, 2004.

[3] J. Eisert, M. Cramer and M.B. Plenio, doi:10/bts7tp, arXiv:0808.3773, 2008.

[4] Y. Huang, arXiv:1403.0327, 2014

1D Ground state approximation

Problem

Given a spin-system with local interactions, approximate the ground state(s) of that system efficiently on a classical computer.

- This is not possible[1] for a general system[2]. We're going to be considering gapped 1D systems, which serve as a nice toy-model in which it is.
- Local and gapped interactions give us the area law[3], a conjectured structural bound on the complexity of ground states.
- Restricting further to 1D allows us to use a rigorous proof of the area law[4].

[1] Given standard complexity theoretic assumptions, analogous to P $\neq$ NP.

[2] J. Kempe, A. Kitaev and O. Regev, doi:10/dqbscx, arXiv:quant-ph/0406180, 2004.

[3] J. Eisert, M. Cramer and M.B. Plenio, doi:10/bts7tp, arXiv:0808.3773, 2008.

[4] Y. Huang, arXiv:1403.0327, 2014

Efficiency

- By efficient we mean that the run-time of the algorithm is polynomial in the system size.
- There are heuristic methods – such as the density matrix renormalisation group – which are typically efficient, but are inefficient in the worst-case[1].
- How and why these heuristic methods fail however is poorly understood, our goal is provable-efficiency.
- May leads to a practical algorithm with efficiency guarantees (c.f. linear programming).
- Also sheds more light on which quantum systems can and cannot be classically simulated.

[1] J. Eisert, doi:10/dtkfcr, arXiv:quant-ph/0609051, 2006.

Efficiency

- By efficient we mean that the run-time of the algorithm is polynomial in the system size.
- There are heuristic methods – such as the density matrix renormalisation group – which are typically efficient, but are inefficient in the worst-case[1].
- How and why these heuristic methods fail however is poorly understood, our goal is provable-efficiency.
- May leads to a practical algorithm with efficiency guarantees (c.f. linear programming).
- Also sheds more light on which quantum systems can and cannot be classically simulated.

[1] J. Eisert, doi:10/dtkfcr, arXiv:quant-ph/0609051, 2006.

Efficiency

- By efficient we mean that the run-time of the algorithm is polynomial in the system size.
- There are heuristic methods – such as the density matrix renormalisation group – which are typically efficient, but are inefficient in the worst-case[1].
- How and why these heuristic methods fail however is poorly understood, our goal is provable-efficiency.
- May leads to a practical algorithm with efficiency guarantees (c.f. linear programming).
- Also sheds more light on which quantum systems can and cannot be classically simulated.

[1] J. Eisert, doi:10/dtkfcr, arXiv:quant-ph/0609051, 2006.

Efficiency

- By efficient we mean that the run-time of the algorithm is polynomial in the system size.
- There are heuristic methods – such as the density matrix renormalisation group – which are typically efficient, but are inefficient in the worst-case[1].
- How and why these heuristic methods fail however is poorly understood, our goal is provable-efficiency.
- May leads to a practical algorithm with efficiency guarantees (c.f. linear programming).
- Also sheds more light on which quantum systems can and cannot be classically simulated.

[1] J. Eisert, doi:10/dtkfcr, arXiv:quant-ph/0609051, 2006.

Efficiency

- By efficient we mean that the run-time of the algorithm is polynomial in the system size.
- There are heuristic methods – such as the density matrix renormalisation group – which are typically efficient, but are inefficient in the worst-case[1].
- How and why these heuristic methods fail however is poorly understood, our goal is provable-efficiency.
- May leads to a practical algorithm with efficiency guarantees (c.f. linear programming).
- Also sheds more light on which quantum systems can and cannot be classically simulated.

[1] J. Eisert, doi:10/dtkfcr, arXiv:quant-ph/0609051, 2006.

Area Law

- One barrier to an efficient algorithm is entanglement, it turns out the states we consider only have limited entanglement however.
- For general states the entanglement entropy S of an arbitrary region A obeys a volume law, for ground states of gapped/local systems it is conjectured to obey an area law.

Volume: $S(A) = \mathcal{O}(|A|)$

Area: $S(A) = \mathcal{O}(|\partial A|)$

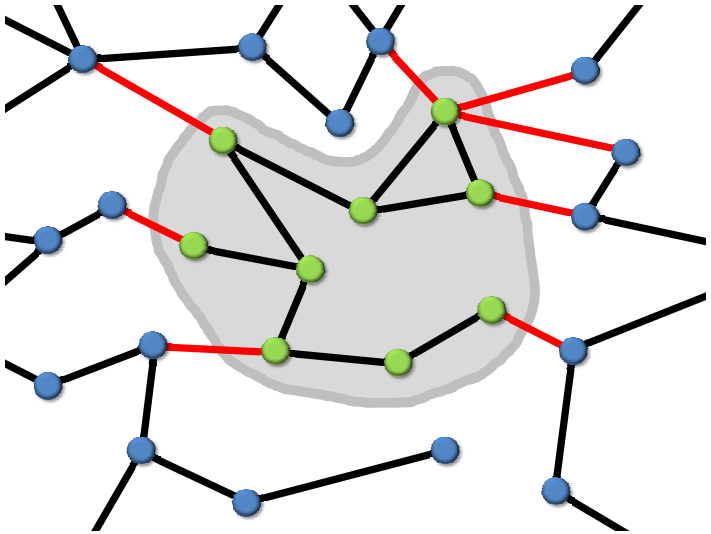
- Restricting to area law states allows us to use the matrix product state ansatz, an efficient state representation.

Area Law

- One barrier to an efficient algorithm is entanglement, it turns out the states we consider only have limited entanglement however.
- For general states the entanglement entropy S of an arbitrary region A obeys a volume law, for ground states of gapped/local systems it is conjectured to obey an area law.

Volume: $S(A) = \mathcal{O}(|A|)$

Area: $S(A) = \mathcal{O}(|\partial A|)$



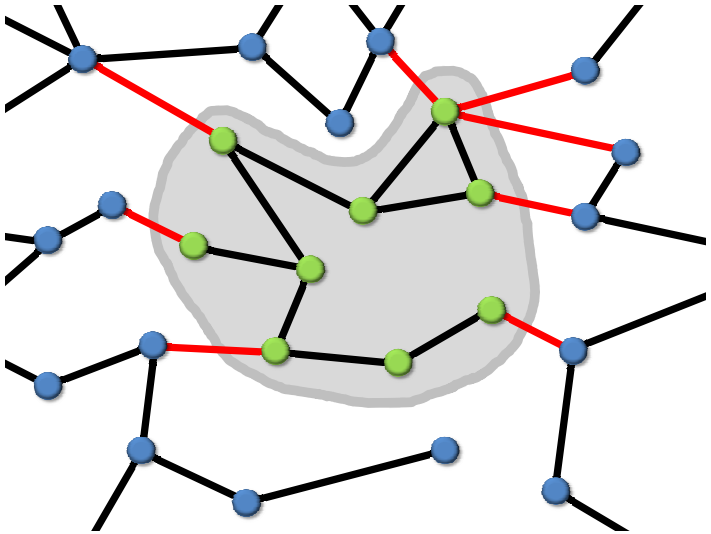
- Restricting to area law states allows us to use the matrix product state ansatz, an efficient state representation.

Area Law

- One barrier to an efficient algorithm is entanglement, it turns out the states we consider only have limited entanglement however.
- For general states the entanglement entropy S of an arbitrary region A obeys a volume law, for ground states of gapped/local systems it is conjectured to obey an area law.

Volume: $S(A) = \mathcal{O}(|A|)$

Area: $S(A) = \mathcal{O}(|\partial A|)$



- Restricting to area law states allows us to use the matrix product state ansatz, an efficient state representation.

A naïve method

- We could simply optimise the Hamiltonian: find a mixed state σ by the convex program

$$\begin{aligned} &\min \ \operatorname{Tr}(H\sigma) \\ &\text{where } \sigma \geq 0,\ \operatorname{Tr}\sigma = 1\,. \end{aligned}$$

  then take $|\Gamma\rangle$ to be the leading eigenvector of σ.
- The domain of σ is however exponentially large, meaning this cannot be computed efficiently.
- This approach can be salvaged if we had some polynomial-sized subspace[1].

[1] Zeph Landau, Umesh Vazirani and Thomas Vidick, doi:10/xd2, arXiv:1307.5143, 2013.

A naïve method

- We could simply optimise the Hamiltonian: find a mixed state σ by the convex program

$$\begin{aligned} &\min \ \operatorname{Tr}(H\sigma) \\ &\text{where } \sigma \geq 0,\ \operatorname{Tr}\sigma = 1\,. \end{aligned}$$

  then take $|\Gamma\rangle$ to be the leading eigenvector of σ.
- The domain of σ is however exponentially large, meaning this cannot be computed efficiently.
- This approach can be salvaged if we had some polynomial-sized subspace[1].

[1] Zeph Landau, Umesh Vazirani and Thomas Vidick, doi:10/xd2, arXiv:1307.5143, 2013.

A naïve method

- We could simply optimise the Hamiltonian: find a mixed state σ by the convex program

$$\begin{aligned} &\min \ \operatorname{Tr}(H\sigma) \\ &\text{where } \sigma \geq 0,\ \operatorname{Tr}\sigma = 1\,. \end{aligned}$$

  then take $|\Gamma\rangle$ to be the leading eigenvector of σ.
- The domain of σ is however exponentially large, meaning this cannot be computed efficiently.
- This approach can be salvaged if we had some polynomial-sized subspace[1].

[1] Zeph Landau, Umesh Vazirani and Thomas Vidick, doi:10/xd2, arXiv:1307.5143, 2013.

Viable sets

A viable set captures the 'local part' of an approximate ground state:

Viable set

A set of states S is (i,δ)-viable if

- The set can be efficiently described.
- The states are defined on the first i spins.
- There exists a witness state $|\psi\rangle$ – which is an approximate ground state of error δ – such that the parts of $|\psi\rangle$ on those first i spins is contained in Span(S).

The span of a $i = n$ viable set forms the desired polynomial subspace for the previous optimisation. Such a set can be constructed inductively.

Viable sets

A viable set captures the 'local part' of an approximate ground state:

Viable set

A set of states S is (i,δ)-viable if

- The set can be efficiently described.
- The states are defined on the first i spins.
- There exists a witness state $|\psi\rangle$ – which is an approximate ground state of error δ – such that the parts of $|\psi\rangle$ on those first i spins is contained in Span(S).

The span of a $i = n$ viable set forms the desired polynomial subspace for the previous optimisation. Such a set can be constructed inductively.

Algorithm structure

Stepping from a viable set on $i-1$ spins to i spins is done via three steps:

1) Extension: Tensor product the viable set with a basis on the ith spin, causing the set to grow whilst the error is preserved.
2) Trimming: Use convex optimisations to remove locally high energy states, bringing the cardinality back down at some error cost.
3) Error Reduction: Using approximate ground state projectors, bringing down the error at some small cardinality cost.

Along with the final optimisation this give a ground state approximation of inverse-polynomial error in run-time

$$T = n^{\mathcal{O}(1/\epsilon)}$$

where ϵ is the gap.

Algorithm structure

Stepping from a viable set on $i-1$ spins to i spins is done via three steps:

1) Extension: Tensor product the viable set with a basis on the ith spin, causing the set to grow whilst the error is preserved.
2) Trimming: Use convex optimisations to remove locally high energy states, bringing the cardinality back down at some error cost.
3) Error Reduction: Using approximate ground state projectors, bringing down the error at some small cardinality cost.

Along with the final optimisation this give a ground state approximation of inverse-polynomial error in run-time

$$T = n^{\mathcal{O}(1/\epsilon)}$$

where ϵ is the gap.

Algorithm structure

Stepping from a viable set on $i-1$ spins to i spins is done via three steps:

1) Extension: Tensor product the viable set with a basis on the ith spin, causing the set to grow whilst the error is preserved.
2) Trimming: Use convex optimisations to remove locally high energy states, bringing the cardinality back down at some error cost.
3) Error Reduction: Using approximate ground state projectors, bringing down the error at some small cardinality cost.

Along with the final optimisation this give a ground state approximation of inverse-polynomial error in run-time

$$T = n^{\mathcal{O}(1/\epsilon)}$$

where ϵ is the gap.

Algorithm structure

Stepping from a viable set on $i-1$ spins to i spins is done via three steps:

1) Extension: Tensor product the viable set with a basis on the ith spin, causing the set to grow whilst the error is preserved.
2) Trimming: Use convex optimisations to remove locally high energy states, bringing the cardinality back down at some error cost.
3) Error Reduction: Using approximate ground state projectors, bringing down the error at some small cardinality cost.

Along with the final optimisation this give a ground state approximation of inverse-polynomial error in run-time

$$T = n^{\mathcal{O}(1/\epsilon)}$$

where ϵ is the gap.

Algorithm structure

Stepping from a viable set on $i-1$ spins to i spins is done via three steps:

1) Extension: Tensor product the viable set with a basis on the ith spin, causing the set to grow whilst the error is preserved.
2) Trimming: Use convex optimisations to remove locally high energy states, bringing the cardinality back down at some error cost.
3) Error Reduction: Using approximate ground state projectors, bringing down the error at some small cardinality cost.

Along with the final optimisation this give a ground state approximation of inverse-polynomial error in run-time

$$T = n^{\mathcal{O}(1/\epsilon)}$$

where ϵ is the gap.

Degeneracy: The problem

- The new result is an extension of this algorithm to degenerate systems, with a run-time of the same scaling.
- The main problem is the size-trimming step, which involves optimising over part of the Hamiltonian.
- Degenerate viable sets requires multiple witnesses and simply performing one optimisation only guarantees the existence of one.
- As we are only optimising parts of the Hamiltonian, the problem of distinguishability arises.

Degeneracy: The problem

- The new result is an extension of this algorithm to degenerate systems, with a run-time of the same scaling.
- The main problem is the size-trimming step, which involves optimising over part of the Hamiltonian.
- Degenerate viable sets requires multiple witnesses and simply performing one optimisation only guarantees the existence of one.
- As we are only optimising parts of the Hamiltonian, the problem of distinguishability arises.

Degeneracy: The problem

- The new result is an extension of this algorithm to degenerate systems, with a run-time of the same scaling.
- The main problem is the size-trimming step, which involves optimising over part of the Hamiltonian.
- Degenerate viable sets requires multiple witnesses and simply performing one optimisation only guarantees the existence of one.
- As we are only optimising parts of the Hamiltonian, the problem of distinguishability arises.

Degeneracy: The problem

- The new result is an extension of this algorithm to degenerate systems, with a run-time of the same scaling.
- The main problem is the size-trimming step, which involves optimising over part of the Hamiltonian.
- Degenerate viable sets requires multiple witnesses and simply performing one optimisation only guarantees the existence of one.
- As we are only optimising parts of the Hamiltonian, the problem of distinguishability arises.

Degeneracy: The solution

Consider a two-fold degeneracy:

- The first step is to perform the original optimisation:

$$\begin{aligned} &\min \ \mathrm{Tr}(H_L \sigma_1) \\ &\text{where } \sigma_1 \geq 0,\ \mathrm{Tr}\,\sigma_1 = 1\,, \end{aligned}$$

  where H_L is the part of the Hamiltonian defined on the first i spins, and the viable set S_1 is constructed from σ_1.

- The second step it to restrict to low energies and project away from this viable set, this corresponds to the convex optimisation

$$\begin{aligned} &\min \ \mathrm{Tr}(P_1 \sigma_2) \\ &\text{where } \sigma_2 \geq 0,\ \mathrm{Tr}\,\sigma_2 = 1, \\ &\qquad \mathrm{Tr}(H_L \sigma_2) \leq \mathrm{Tr}(H_L \sigma_1) + \text{small error}\,, \end{aligned}$$

  where P_1 is the projector onto $\mathrm{Span}(S_1)$, and S_2 is constructed from σ_2, analogous to S_1.

Degeneracy: The solution

Consider a two-fold degeneracy:

- The first step is to perform the original optimisation:

$$\begin{aligned} &\min \ \operatorname{Tr}(H_L \sigma_1) \\ &\text{where } \sigma_1 \geq 0,\ \operatorname{Tr} \sigma_1 = 1\,, \end{aligned}$$

  where H_L is the part of the Hamiltonian defined on the first i spins, and the viable set S_1 is constructed from σ_1.

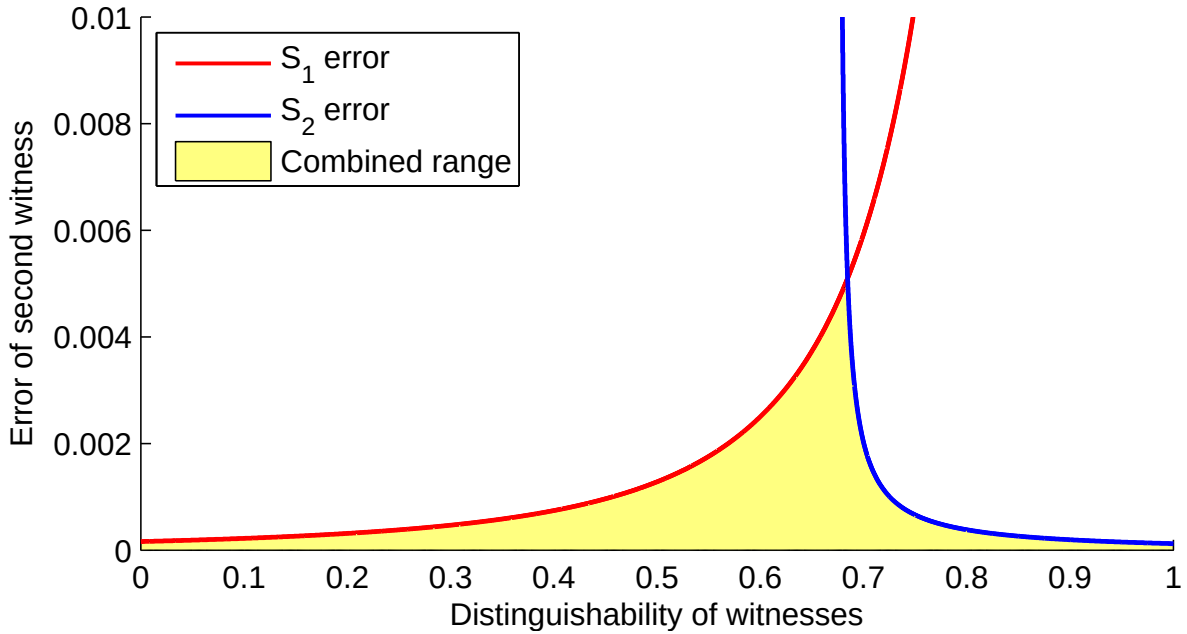
- The second step it to restrict to low energies and project away from this viable set, this corresponds to the convex optimisation

$$\begin{aligned} &\min \ \operatorname{Tr}(P_1 \sigma_2) \\ &\text{where } \sigma_2 \geq 0,\ \operatorname{Tr} \sigma_2 = 1, \\ &\qquad \operatorname{Tr}(H_L \sigma_2) \leq \operatorname{Tr}(H_L \sigma_1) + \text{small error}\,, \end{aligned}$$

  where P_1 is the projector onto $\operatorname{Span}(S_1)$, and S_2 is constructed from σ_2, analogous to S_1.

Degeneracy: The result

It turns out that the error of the second witness in S_1/S_2 depends on the distinguishability.



By taking the union the error induced by trimming can be kept low, and the error of the algorithm kept down.

Conclusion

- The ground states of any gapped and 1D local spin-chain can be approximated with inverse-polynomial error in run-time

$$T = n^{\mathcal{O}(1/\epsilon)}$$

  where ϵ is the gap.
- As such 1D gapped systems are able to be classically simulated to some extent.
- This may lead to practical algorithms with efficiency guarantees.