Quantum Recommendation Systems.

Iordanis Kerenidis [1] Anupam Prakash [2]

[1] CNRS, Université Paris Diderot, Paris, France.

[2] Nanyang Technological University, Singapore.

January 17, 2017

The *HHL* algorithm

- Utilize intrinsic linear algebra capabilities of quantum computers for *exponential* speedups.

THE *HHL* ALGORITHM

- Utilize intrinsic linear algebra capabilities of quantum computers for *exponential* speedups.
- Vector state $|x\rangle = \sum_i x_i |i\rangle$ where $x \in \mathbb{R}^n$ is a unit vector.

The *HHL* algorithm

- Utilize intrinsic linear algebra capabilities of quantum computers for *exponential* speedups.
- Vector state $|x\rangle = \sum_i x_i |i\rangle$ where $x \in \mathbb{R}^n$ is a unit vector.
- Given sparse matrix $A \in \mathbb{R}^{n \times n}$ and $|b\rangle$ there is a quantum algorithm to prepare $|A^{-1}b\rangle$ in time polylog(n). [Harrow, Hassidim, Lloyd]

THE *HHL* ALGORITHM

- Utilize intrinsic linear algebra capabilities of quantum computers for *exponential* speedups.
- Vector state $|x\rangle = \sum_i x_i |i\rangle$ where $x \in \mathbb{R}^n$ is a unit vector.
- Given sparse matrix $A \in \mathbb{R}^{n \times n}$ and $|b\rangle$ there is a quantum algorithm to prepare $|A^{-1}b\rangle$ in time polylog(n). [Harrow, Hassidim, Lloyd]
- Assumptions: $|b\rangle$ can be prepared polylog(n) time and A is polylog(n) sparse.

The *HHL* algorithm

- Utilize intrinsic linear algebra capabilities of quantum computers for *exponential* speedups.
- Vector state $|x\rangle = \sum_i x_i |i\rangle$ where $x \in \mathbb{R}^n$ is a unit vector.
- Given sparse matrix $A \in \mathbb{R}^{n \times n}$ and $|b\rangle$ there is a quantum algorithm to prepare $|A^{-1}b\rangle$ in time polylog(n). [Harrow, Hassidim, Lloyd]
- Assumptions: $|b\rangle$ can be prepared polylog(n) time and A is polylog(n) sparse.
- Incomparable to classical linear system solver which returns vector $x \in \mathbb{R}^n$ as opposed to $|x\rangle$.

Quantum Machine Learning

- *HHL* led to several proposals for quantum machine learning algorithms.

Quantum Machine Learning

- *HHL* led to several proposals for quantum machine learning algorithms.
- Principal components analysis, classification with ℓ_2-SVMs, k-means clustering, perceptron, nearest neighbors... [Lloyd, Mohseni, Rebentrost, Wiebe, Kapoor, Svore]

Quantum Machine Learning

- *HHL* led to several proposals for quantum machine learning algorithms.
- Principal components analysis, classification with ℓ_2-SVMs, k-means clustering, perceptron, nearest neighbors... [Lloyd, Mohseni, Rebentrost, Wiebe, Kapoor, Svore]
- Algorithms achieve exponential speedups only for sparse/well-conditioned data.

Quantum Machine Learning

- *HHL* led to several proposals for quantum machine learning algorithms.
- Principal components analysis, classification with ℓ_2-SVMs, k-means clustering, perceptron, nearest neighbors... [Lloyd, Mohseni, Rebentrost, Wiebe, Kapoor, Svore]
- Algorithms achieve exponential speedups only for sparse/well-conditioned data.
- Sometimes a variant of the classical problem is solved: ℓ_1 vs ℓ_2-SVM.

Quantum Machine Learning

- *HHL* led to several proposals for quantum machine learning algorithms.
- Principal components analysis, classification with ℓ_2-SVMs, k-means clustering, perceptron, nearest neighbors... [Lloyd, Mohseni, Rebentrost, Wiebe, Kapoor, Svore]
- Algorithms achieve exponential speedups only for sparse/well-conditioned data.
- Sometimes a variant of the classical problem is solved: ℓ_1 vs ℓ_2-SVM.
- Incomparable with classical.

Quantum Recommendation Systems

- *Open problem:* A quantum machine learning algorithm with *exponential worst case* speedup for classical problem.

Quantum Recommendation Systems

- *Open problem:* A quantum machine learning algorithm with *exponential worst case* speedup for classical problem.
- *Quantum recommendation systems*.

Quantum Recommendation Systems

- *Open problem:* A quantum machine learning algorithm with *exponential worst case* speedup for classical problem.
- *Quantum recommendation systems*.
- An exponential speedup over classical with similar assumptions and guarantees.

Quantum Recommendation Systems

- *Open problem:* A quantum machine learning algorithm with *exponential worst case* speedup for classical problem.
- *Quantum recommendation systems*.
- An exponential speedup over classical with similar assumptions and guarantees.
- An end to end application with no assumptions on the data set.

Quantum Recommendation Systems

- *Open problem:* A quantum machine learning algorithm with *exponential worst case* speedup for classical problem.
- *Quantum recommendation systems*.
- An exponential speedup over classical with similar assumptions and guarantees.
- An end to end application with no assumptions on the data set.
- Solves the 'same' problem as a classical recommendation system.

The Recommendation Problem

- The preference matrix P.

	P_1	P_2	P_3	P_4	$\cdots$	$\cdots$	P_{n-1}	P_n
U_1	.1	.4	?	?	$\cdots$	$\cdots$	?	.9
U_2	.2	?	.6	?	$\cdots$	$\cdots$	.85	?
U_3	?	?	.8	.9	$\cdots$	$\cdots$	?	.2
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
U_m	?	.75	?	?	$\cdots$	$\cdots$	?	.2

The Recommendation Problem

- The preference matrix P.

	P_1	P_2	P_3	P_4	$\cdots$	$\cdots$	P_{n-1}	P_n
U_1	.1	.4	?	?	$\cdots$	$\cdots$	?	.9
U_2	.2	?	.6	?	$\cdots$	$\cdots$	.85	?
U_3	?	?	.8	.9	$\cdots$	$\cdots$	?	.2
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
U_m	?	.75	?	?	$\cdots$	$\cdots$	?	.2

- P_{ij} is the value of item j for user i. Samples from P arrive in an online manner.

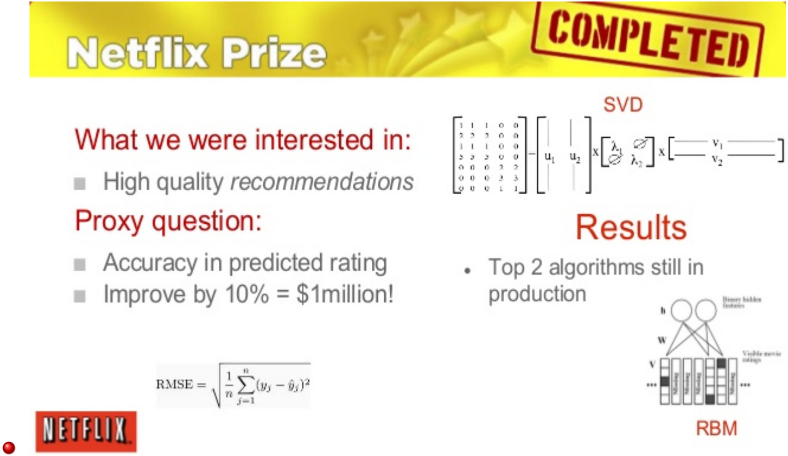
The Recommendation Problem

- The preference matrix P.

	P_1	P_2	P_3	P_4	$\cdots$	$\cdots$	P_{n-1}	P_n
U_1	.1	.4	?	?	$\cdots$	$\cdots$	?	.9
U_2	.2	?	.6	?	$\cdots$	$\cdots$	.85	?
U_3	?	?	.8	.9	$\cdots$	$\cdots$	?	.2
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
U_m	?	.75	?	?	$\cdots$	$\cdots$	?	.2

- P_{ij} is the value of item j for user i. Samples from P arrive in an online manner.
- The assumption that P has a good rank-k approximation for small k is widely used.

The Netflix problem



Reconstruction vs sampling

- Matrix reconstruction algorithms reconstruct $\widetilde{P} \approx P$ using the low rank assumption and require time poly(mn).

Reconstruction vs sampling

- Matrix reconstruction algorithms reconstruct $\widetilde{P} \approx P$ using the low rank assumption and require time poly(mn).
- A reconstruction based recommendation system requires time poly(n), even with pre-computation.

Reconstruction vs sampling

- Matrix reconstruction algorithms reconstruct $\widetilde{P} \approx P$ using the low rank assumption and require time poly(mn).
- A reconstruction based recommendation system requires time poly(n), even with pre-computation.
- Matrix sampling suffices to obtain good recommendations.

Reconstruction vs sampling

- Matrix reconstruction algorithms reconstruct $\widetilde{P} \approx P$ using the low rank assumption and require time poly(mn).
- A reconstruction based recommendation system requires time poly(n), even with pre-computation.
- Matrix sampling suffices to obtain good recommendations.
- Quantum algorithms can perform matrix sampling.

Reconstruction vs sampling

- Matrix reconstruction algorithms reconstruct $\widetilde{P} \approx P$ using the low rank assumption and require time poly(mn).
- A reconstruction based recommendation system requires time poly(n), even with pre-computation.
- Matrix sampling suffices to obtain good recommendations.
- Quantum algorithms can perform matrix sampling.

Theorem

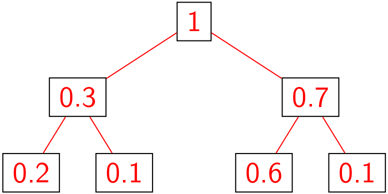
There is a quantum recommendation algorithm with running time $O(poly(k)polylog(mn))$.

Computational Model

- Samples from P arrive in an online manner and are stored in data structure with update time $O(\log^2 mn)$.

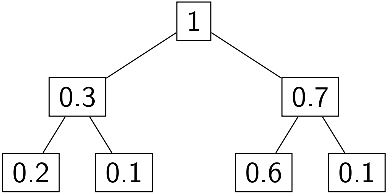
Computational Model

- Samples from P arrive in an online manner and are stored in data structure with update time $O(\log^2 mn)$.
- The quantum algorithm has oracle access to binary tree data structure storing additional metadata.



Computational Model

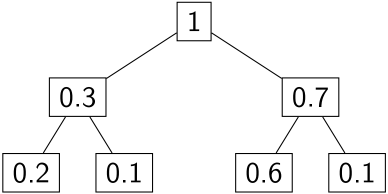
- Samples from P arrive in an online manner and are stored in data structure with update time $O(\log^2 mn)$.
- The quantum algorithm has oracle access to binary tree data structure storing additional metadata.



- We use the standard memory model used for algorithms like Grover search.

Computational Model

- Samples from P arrive in an online manner and are stored in data structure with update time $O(\log^2 mn)$.
- The quantum algorithm has oracle access to binary tree data structure storing additional metadata.



- We use the standard memory model used for algorithms like Grover search.
- Users arrive into system in an online manner and system provides recommendations in time $poly(k)polylog(mn)$.

Singular value estimation

- The singular value decomposition for matrix A is written as $A = \sum_i \sigma_i u_i v_i^t$.

Singular value estimation

- The singular value decomposition for matrix A is written as $A = \sum_i \sigma_i u_i v_i^t$.
- The rank-k approximation $A_k = \sum_{i \in [k]} \sigma_i u_i v_i^t$ minimizes $\|A - A_k\|_F$.

Singular value estimation

- The singular value decomposition for matrix A is written as $A = \sum_i \sigma_i u_i v_i^t$.
- The rank-k approximation $A_k = \sum_{i \in [k]} \sigma_i u_i v_i^t$ minimizes $\|A - A_k\|_F$.
- Quantum singular value estimation:

Singular value estimation

- The singular value decomposition for matrix A is written as $A = \sum_i \sigma_i u_i v_i^t$.
- The rank-k approximation $A_k = \sum_{i \in [k]} \sigma_i u_i v_i^t$ minimizes $\|A - A_k\|_F$.
- Quantum singular value estimation:

Theorem

There is an algorithm with running time $O(polylog(mn)/\epsilon)$ that transforms $\sum_i \alpha_i |v_i\rangle \rightarrow \sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$ where $\overline{\sigma_i} \in \sigma_i \pm \epsilon \|A\|_F$ with probability at least $1 - 1/poly(n)$.

Matrix Sampling

- Let T be a $0/1$ matrix such that $T_{ij} = 1$ if item j is 'good' recommendation for user i.

	P_1	P_2	P_3	P_4	$\cdots$	$\cdots$	P_{n-1}	P_n
U_1	0	0	?	?	$\cdots$	$\cdots$	?	1
U_2	0	?	0	?	$\cdots$	$\cdots$	1	?
U_3	?	?	1	1	$\cdots$	$\cdots$	?	0
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
U_m	?	1	?	?	$\cdots$	$\cdots$	?	0

Matrix Sampling

- Let T be a 0/1 matrix such that $T_{ij} = 1$ if item j is 'good' recommendation for user i.

	P_1	P_2	P_3	P_4	$\cdots$	$\cdots$	P_{n-1}	P_n
U_1	0	0	?	?	$\cdots$	$\cdots$	?	1
U_2	0	?	0	?	$\cdots$	$\cdots$	1	?
U_3	?	?	1	1	$\cdots$	$\cdots$	?	0
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
U_m	?	1	?	?	$\cdots$	$\cdots$	?	0

- Set the ?s to 0 and rescale to obtain a *subsample* matrix $\widehat{T}$.

Matrix Sampling

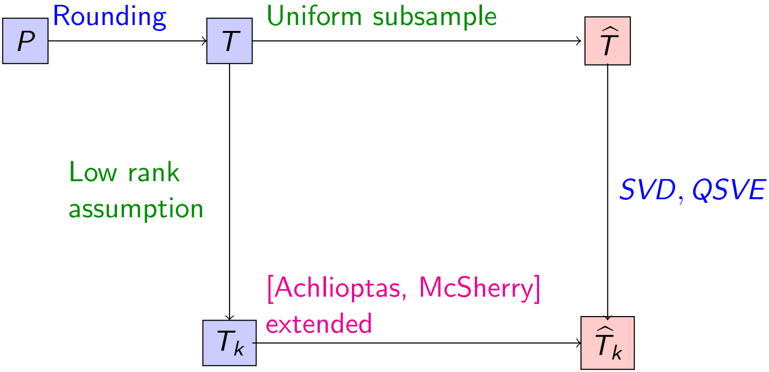


Figure: Matrix sampling based recommendation system.

Matrix Sampling

- T is the binary recommendation matrix obtained by rounding P.

Matrix Sampling

- T is the binary recommendation matrix obtained by rounding P.
- $\widehat{T}$ is a uniform subsample of T:

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/p & \text{[with probability } p\text{]} \\ 0 & \text{[otherwise]} \end{cases}$$

Matrix Sampling

- T is the binary recommendation matrix obtained by rounding P.
- $\widehat{T}$ is a uniform subsample of T:

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/p & \text{[with probability } p\text{]} \\ 0 & \text{[otherwise]} \end{cases}$$

- T_k and $\widehat{T}_k$ are rank-k approximations for T and $\widehat{T}$.

Matrix Sampling

- T is the binary recommendation matrix obtained by rounding P.
- $\widehat{T}$ is a uniform subsample of T:

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/p & \text{[with probability } p\text{]} \\ 0 & \text{[otherwise]} \end{cases}$$

- T_k and $\widehat{T}_k$ are rank-k approximations for T and $\widehat{T}$.
- The low rank assumption implies that $\|T - T_k\| \leq \epsilon \|T\|_F$ for small k.

Matrix Sampling

- T is the binary recommendation matrix obtained by rounding P.
- $\widehat{T}$ is a uniform subsample of T:

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/p & \text{[with probability } p\text{]} \\ 0 & \text{[otherwise]} \end{cases}$$

- T_k and $\widehat{T}_k$ are rank-k approximations for T and $\widehat{T}$.
- The low rank assumption implies that $\|T - T_k\| \leq \epsilon \|T\|_F$ for small k.
- Analysis: Sampling from matrix 'close to' $\widehat{T}_k$ yields good recommendations.

Analysis

- Samples from T_k are good recommendations, for large fraction of 'typical' users.

Analysis

- Samples from T_k are good recommendations, for large fraction of 'typical' users.
- Sampling from $\widehat{T}_k$ suffices.

Analysis

- Samples from T_k are good recommendations, for large fraction of 'typical' users.
- Sampling from $\widehat{T}_k$ suffices.

Theorem (AM02)

If $\widehat{A}$ is obtained from a $0/1$ matrix A by subsampling with probability $p = 16n/\eta \|A\|_F^2$ then with probability at least $1 - exp(-19(\log n)^4)$, for all k,

$$||A - \widehat{A}_k||_F \leq ||A - A_k||_F + 3\sqrt{\eta}k^{1/4}||A||_F$$

Analysis

- The quantum algorithm samples from $\widehat{T}_{\geq \sigma, \kappa}$, a projection onto all singular values $\geq \sigma$ and some in the range $[(1-\kappa)\sigma, \sigma)$.

Analysis

- The quantum algorithm samples from $\widehat{T}_{\geq \sigma, \kappa}$, a projection onto all singular values $\geq \sigma$ and some in the range $[(1-\kappa)\sigma, \sigma)$.
- We extend *AM*02 to this setting showing that:

$$||T - \widehat{T}_{\sigma,\kappa}||_F \leq 9\epsilon \, \| T \|_F$$

Analysis

- The quantum algorithm samples from $\widehat{T}_{\geq\sigma,\kappa}$, a projection onto all singular values $\geq \sigma$ and some in the range $[(1-\kappa)\sigma, \sigma)$.
- We extend $AM02$ to this setting showing that:

$$||T - \widehat{T}_{\sigma,\kappa}||_F \leq 9\epsilon \, \| T \|_F$$

- For most typical users, samples from $(\widehat{T}_{\sigma,\kappa})_i$ are good recommendations with high probability.

Quantum Recommendation Algorithm

- Prepare state $|\widehat{T}_i\rangle$ corresponding to row for user i.

Quantum Recommendation Algorithm

- Prepare state $|\widehat{T}_i\rangle$ corresponding to row for user i.
- Apply quantum projection algorithm to $|\widehat{T}_i\rangle$ to obtain $|(\widehat{T}_{\geq\sigma,\kappa})_i\rangle$.

Quantum Recommendation Algorithm

- Prepare state $|\widehat{T}_i\rangle$ corresponding to row for user i.
- Apply quantum projection algorithm to $|\widehat{T}_i\rangle$ to obtain $|(\widehat{T}_{\geq\sigma,\kappa})_i\rangle$.
- Measure projected state in computational basis to get recommendation.

Quantum Recommendation Algorithm

- Prepare state $|\widehat{T}_i\rangle$ corresponding to row for user i.
- Apply quantum projection algorithm to $|\widehat{T}_i\rangle$ to obtain $|(\widehat{T}_{\geq\sigma,\kappa})_i\rangle$.
- Measure projected state in computational basis to get recommendation.
- The threshold $\sigma = \frac{\epsilon\sqrt{p}\|A\|_F}{\sqrt{2k}}$ and $\kappa = \frac{1}{3}$.

Quantum Recommendation Algorithm

- Prepare state $|\widehat{T}_i\rangle$ corresponding to row for user i.
- Apply quantum projection algorithm to $|\widehat{T}_i\rangle$ to obtain $|(\widehat{T}_{\geq\sigma,\kappa})_i\rangle$.
- Measure projected state in computational basis to get recommendation.
- The threshold $\sigma = \frac{\epsilon\sqrt{p}\|A\|_F}{\sqrt{2k}}$ and $\kappa = \frac{1}{3}$.
- Running time depends on the threshold and not the condition number.

The projection algorithm

- Let $A = \sum_i \sigma_i u_i v_i^t$ be the singular value decomposition, write input $|x\rangle = \sum_i \alpha_i |v_i\rangle$.

The projection algorithm

- Let $A = \sum_i \sigma_i u_i v_i^t$ be the singular value decomposition, write input $|x\rangle = \sum_i \alpha_i |v_i\rangle$.
- Estimate singular values $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$ to additive error $\kappa\sigma/2$.

The projection algorithm

- Let $A = \sum_i \sigma_i u_i v_i^t$ be the singular value decomposition, write input $|x\rangle = \sum_i \alpha_i |v_i\rangle$.
- Estimate singular values $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$ to additive error $\kappa\sigma/2$.
- Map to $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle |t\rangle$ where $t = 1$ if $\overline{\sigma_i} \geq (1 - \kappa/2)\sigma$ and erase $\overline{\sigma_i}$.

The projection algorithm

- Let $A = \sum_i \sigma_i u_i v_i^t$ be the singular value decomposition, write input $|x\rangle = \sum_i \alpha_i |v_i\rangle$.
- Estimate singular values $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$ to additive error $\kappa\sigma/2$.
- Map to $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle |t\rangle$ where $t = 1$ if $\overline{\sigma_i} \geq (1 - \kappa/2)\sigma$ and erase $\overline{\sigma_i}$.
- Post-select on $t = 1$.

THE PROJECTION ALGORITHM

- Let $A = \sum_i \sigma_i u_i v_i^t$ be the singular value decomposition, write input $|x\rangle = \sum_i \alpha_i |v_i\rangle$.
- Estimate singular values $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$ to additive error $\kappa\sigma/2$.
- Map to $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle |t\rangle$ where $t = 1$ if $\overline{\sigma_i} \geq (1-\kappa/2)\sigma$ and erase $\overline{\sigma_i}$.
- Post-select on $t = 1$.
- The output $|A_{\geq \sigma,\kappa} x\rangle$ a projection the space of singular vectors with singular values $\geq \sigma$ and some in the range $[(1-\kappa)\sigma, \sigma)$.

Open Questions

- Find a classical algorithm matrix sampling based recommendation algorithm that runs in time $O(\text{poly}(k)\text{polylog}(mn))$.

  OR

  Prove a lower bound to rule out such an algorithm.
- Find more quantum machine learning algorithms.