Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 128 (2018)

A quantum-inspired classical algorithm for recommendation systems

Ewin Tang

April 13, 2019

Abstract

We give a classical analogue to Kerenidis and Prakash's quantum recommendation system, previously believed to be one of the strongest candidates for provably exponential speedups in quantum machine learning. Our main result is an algorithm that, given an $m \times n$ matrix in a data structure supporting certain ℓ^2-norm sampling operations, outputs an ℓ^2-norm sample from a rank-k approximation of that matrix in time $O(\text{poly}(k)\log(mn))$, only polynomially slower than the quantum algorithm. As a consequence, Kerenidis and Prakash's algorithm does not in fact give an exponential speedup over classical algorithms. Further, under strong input assumptions, the classical recommendation system resulting from our algorithm produces recommendations exponentially faster than previous classical systems, which run in time linear in m and n.

The main insight of this work is the use of simple routines to manipulate ℓ^2-norm sampling distributions, which play the role of quantum superpositions in the classical setting. This correspondence indicates a potentially fruitful framework for formally comparing quantum machine learning algorithms to classical machine learning algorithms.

Contents

ISSN 1433-8092

1 Introduction

1.1 Quantum Machine Learning

This work stems from failed efforts to prove that Kerenidis and Prakash's quantum recommendation system algorithm [KP17b] achieves an exponential speedup over any classical algorithm. Such a result would be interesting because Kerenidis and Prakash's algorithm is a quantum machine learning (QML) algorithm.

Though QML has been studied since 1995 [BJ99], it has garnered significant attention in recent years, beginning in 2008 with Harrow, Hassidim, and Lloyd's quantum algorithm for solving linear systems [HHL09]. This burgeoning field has produced exciting quantum algorithms that give hope for finding exponential speedups outside of the now-established gamut of problems related to period finding and Fourier coefficients. However, in part because of caveats exposited by Aaronson [Aar15], it is not clear whether any known QML algorithm gives a new exponential speedup over classical algorithms for practically relevant instances of a machine learning problem. Kerenidis and Prakash's work was notable for addressing all of these caveats, giving a complete quantum algorithm that could be compared directly to classical algorithms.

When Kerenidis and Prakash's work was published, their algorithm was exponentially faster than the best-known classical algorithms. It was not known whether this was a provably exponential speedup. The goal of this work is to describe a classical algorithm that performs the same task as the quantum recommendation systems algorithm with only polynomially slower runtime, thus answering this question. This removes one of the most convincing examples we have of exponential speedups for machine learning problems (see Section 6.7 of [Pre18]).

How the quantum algorithm works. Outside of the recommendation systems context, the quantum algorithm just samples from the low-rank approximation of an input matrix. It

proceeds as follows. First, an application of phase estimation implicitly estimates singular values and locates singular vectors of the input. A quantum projection procedure then uses this information to project a quantum state with a row of the input to a state with the corresponding row of a low-rank approximation of the input. Measuring this state samples an entry from the row with probability proportional to its magnitude. Kerenidis and Prakash posited that, with a classical algorithm, producing samples following these distributions requires time linear in the input dimensions.

The intuition behind this claim is not that singular value estimation or computing projections is particularly difficult computationally. Rather, it's simply hard to believe that any of these steps can be done without reading the full input. After all, a significant portion of the theory of low-rank matrix approximation only asks for time complexity linear in input-sparsity or sublinear with some number of passes through the data. By comparison to these types of results, what the quantum algorithm achieves (query complexity polylogarithmic in the input size) is impressive.

With this claim in mind, Kerenidis and Prakash then apply their quantum algorithm to make a fast online recommendation system. As information arrives about user-product preferences, we place it into a dynamic data structure that is necessary to run the quantum algorithm. We can then service requests for recommendations as they arrive by running the quantum algorithm and returning the output sample. Under strong assumptions about the input data, this sample is likely to be a good recommendation.

State preparation: the quantum algorithm's assumption. To see why the classical algorithm we present is possible, we need to consider the technique Kerenidis and Prakash use to construct their relevant quantum states.

Kerenidis and Prakash's algorithm is one of many QML algorithms [HHL09; LMR13; LMR14; KP17a] that require quantum state preparation assumptions, which state that given an input vector v, one can quickly form a corresponding quantum state $|v\rangle$. To achieve the desired runtime in practice, an implementation would replace this assumption with either a procedure to prepare a state from an arbitrary input vector (where the cost of preparation could be amortized over multiple runs of the algorithm) or a specification of input vectors for which quantum state preparation is easy. Usually QML algorithms abstract away these implementation details, assuming a number of the desired quantum states are already prepared. The quantum recommendation systems algorithm is unique in that it explicitly comes with a data structure to prepare its states (see Section 3).

These state preparation assumptions are nontrivial: even given ability to query entries of a vector in superposition, preparing states corresponding to arbitrary length-n input vectors is known to take $\Omega(\sqrt{n})$ time (a corollary of quantum search lower bounds [Ben+97]). Thus, the data structure to quickly prepare quantum states is essential for the recommendation systems algorithm to achieve query complexity polylogarithmic in input size.

How a classical algorithm can perform as well as the quantum algorithm. The classical algorithm we present cannot be compared to classical low-rank matrix completion and approximation results requiring linear time. Rather, the key insight is that the data structure used to satisfy state preparation assumptions can also satisfy ℓ^2-norm sampling

assumptions (defined in Section 2.2).

So, a classical algorithm whose goal is to "match" the quantum algorithm can exploit these assumptions. The effectiveness of ℓ^2-norm sampling in machine learning [SWZ16; HKS11] and randomized linear algebra [KV17; DMM08] is well-established. In fact, a work by Frieze, Kannan, and Vempala [FKV04] shows that, with ℓ^2-norm sampling assumptions, a form of singular value estimation is possible in time independent of m and n. Further, in the context of this literature, sampling from the projection of a vector onto a subspace is not outside the realm of feasibility. This work just puts these two pieces together.

The importance of ℓ^2-norm sampling. In an imprecise sense, our algorithm replaces state preparation assumptions with ℓ^2-norm sampling assumptions. In this particular case, while quantum superpositions served to represent data implicitly that would take linear time to write out, this need can be served just as well with probability distributions and subsamples of larger pieces of data.

The correspondence between ℓ^2-norm sampling assumptions and state preparation assumptions makes sense. While the former sidesteps the obvious search problems inherent in linear algebra tasks by pinpointing portions of vectors or matrices with the most weight, the latter sidesteps these search problems by allowing for quantum states that are implicitly aware of weight distribution. We suspect that this connection revealed by the state preparation data structure is somewhat deep, and cannot be fixed by simply finding a state preparation data structure without sampling power.

This work demonstrates one major case where classical computing with ℓ^2-norm sampling is an apt point of comparison for revealing speedups (or, rather, the lack thereof) in QML. We believe that this reference point remains useful, even for QML algorithms that don't specify state preparation implementation details, and thus are formally incomparable to any classical model. So, we suggest a general framework for studying the speedups given by QML algorithms with state preparation assumptions: compare QML algorithms with state preparation to classical algorithms with sampling. Indeed, a QML algorithm using state preparation assumptions *should* aim to surpass the capabilities of classical algorithms with ℓ^2-norm sampling, given that in theory, generic state preparation tends to only appear in settings with generic sampling, and in practice, we already know how to implement fast classical sampling on existing hardware.

In summary, we argue for the following guideline: *when QML algorithms are compared to classical ML algorithms in the context of finding speedups, any state preparation assumptions in the QML model should be matched with ℓ^2-norm sampling assumptions in the classical ML model.*

1.2 Recommendation Systems

In addition to our algorithm having interesting implications for QML, it also can be used as a recommendation system.

To formalize the problem of recommending products to users, we use the following model, first introduced in 1998 by Kumar et al. [Kum+01] and refined further by Azar et al. [Aza+01]

and Drineas et al. [DKR02]. We represent the sentiments of m users towards n products with an $m \times n$ *preference matrix* T, where T_{ij} is large if user i likes product j. We further assume that T is close to a matrix of small rank k (constant or logarithmic in m and n), reflecting the intuition that users tend to fall into a small number of classes based on their preferences. Given a matrix A containing only a subset of entries of T, representing our incomplete information about user-product preferences, our goal is to output high-value entries of T, representing good recommendations.

Modern work on recommendation systems uses matrix completion to solve this (which works well in practice[1]), but these techniques must take linear time to produce a recommendation. Kerenidis and Prakash's recommendation system (and, consequently, this work) follows in an older line of research, which experiments with very strong assumptions on input with the hope of finding a new approach that can drive down runtime to sublinear in m and n. In Kerenidis and Prakash's model, finding a good recommendation for a user i reduces to sampling from the ith row of a low rank *approximation* of the subsampled data A, instead of a low-rank completion. Using our classical analogue to Kerenidis and Prakash's algorithm, we can get recommendations in $O(\text{poly}(k)\,\text{polylog}(m,n))$ time, exponentially faster than the best-known in the literature. Sublinear-time sampling for good recommendations has been proposed before (see introduction of [DKR02]), but previous attempts to implement it failed to circumvent the bottleneck of needing linear time to write down input-sized vectors.

For context, our model is most similar to the model given in 2002 by Drineas et al. [DKR02]. However, that algorithm's main goal is minimizing the number of user preferences necessary to generate good recommendations; we discuss in Appendix B how to adapt our algorithm to that model to get similar results. Other approaches include combinatorial techniques [Kum+01; Awe+05] and the use of mixture models [KS08].

We note two major assumptions present in our model that differ from the matrix completion setting. (The full list of assumptions is given in Section 5.) The first assumption is that our subsample A is contained in a data structure. This makes sense in the setting of an online recommendation system, where we can amortize the cost of our preprocessing. Recommendation systems in practice tend to be online, so building a system that keeps data in this data structure to satisfy this assumption seems reasonable. The second assumption is that we know a constant fraction of the full preference matrix T. This is impractical and worse than matrix completion, for which we can prove correctness given as little as an $\tilde{\Theta}(\frac{k}{m+n})$ fraction of input data [Rec11]. This seems to be an issue common among works reducing recommendation problems to low-rank matrix approximation [DKR02; Aza+01]. Nevertheless, we hope that this algorithm, by presenting a novel sampling-based technique with a much faster asymptotic runtime, inspires improved practical techniques and provides an avenue for further research.

[1]Work on the Netflix Prize is a useful reference for practical recommendation systems: see Bennett et al. [BL07] for a broad overview of techniques, Koren et al. [KBV09] for a high-level exposition of the SVD technique, and Bell et al. [BK07] for more technical details.

1.3 Algorithm Sketch

We first state the main result, a classical algorithm that can sample a high-value entry from a given row of a low-rank approximation of a given matrix. The formal statement can be found in Section 4.3.

Theorem (1, informal)**.** *Suppose we are given as input a matrix A supporting query and ℓ^2-norm sampling operations, a row $i \in [m]$, a singular value threshold σ, an error parameter $\eta > 0$, and a sufficiently small $\varepsilon > 0$. There is a classical algorithm whose output distribution is ε-close in total variation distance to the distribution given by ℓ^2-norm sampling from the ith row of a low-rank approximation D of A in query and time complexity*

$$O\left(\text{poly}\left(\frac{\|A\|_F}{\sigma}, \frac{1}{\varepsilon}, \frac{1}{\eta}\frac{\|A_i\|}{\|D_i\|},\right)\right),$$

where the quality of D depends on η.

This makes the runtime *independent* of m and n. Here, $(\|A\|_F/\sigma)^2$ is a bound on the rank of the low-rank approximation, so we think of σ as something like $\|A\|_F/\sqrt{k}$. To implement the needed sampling operations, we will use the data structure described in Section 3, which adds at most an additional $O(\log(mn))$ factor in overhead. This gives a time complexity of

$$\tilde{O}\left(\frac{\|A\|^{24}}{\sigma^{24}\varepsilon^{12}\eta^6}\log(mn)\frac{\|A_i\|^2}{\|D_i\|^2}\right).$$

This is a large slowdown versus the quantum algorithm in some exponents. However, we suspect that these exponents can be improved significantly through standard techniques.

The only difference between Theorem 1 and its quantum equivalent in [KP17b] is that the quantum algorithm has only logarithmic dependence[2] on ε. Thus, we can say that our algorithm performs just as well, up to polynomial slowdown and ε approximation factors (the quantum algorithm still has polynomial dependence on η). These ε's don't affect the classical recommendation system guarantees:

Theorem (2, informal)**.** *Applying Theorem 1 to the recommendation systems model with the quantum state preparation data structure achieves identical bounds on recommendation quality as the quantum algorithm in [KP17b] up to constant factors, for sufficiently small ε.*

To prove Theorem 1 (Section 4), we present and analyze Algorithm 3. It combines a variety of techniques, all relying on sampling access to relevant input vectors. The main restriction to keep in mind is that we need to perform linear algebra operations without incurring the cost of reading a full row or column.

The algorithm begins by using the given support for ℓ^2-norm sampling to run a sampling routine (called ModFKV, see Section 4.2) based on Frieze, Kannan, and Vempala's 1998 algorithm [FKV04] to find a low-rank approximation of A. It doesn't have enough time to output the matrix in full; instead, it outputs a succinct description of the matrix. This

[2]The analysis of the phase estimation in the original paper has some ambiguities, but subsequent work [Gil+18] demonstrates that essentially the same result can be achieved with a different algorithm.

description is S, a normalized constant-sized subset of rows of A, along with some constant-sized matrices $\hat{U}$ and $\hat{\Sigma}$, which implicitly describe $\hat{V} := S^T\hat{U}\hat{\Sigma}^{-1}$, a matrix whose columns are approximate right singular vectors of A. The corresponding low-rank approximation is $D := A\hat{V}\hat{V}^T$, the projection of the rows of the input matrix onto the low-dimensional subspace spanned by $\hat{V}$. Though computing $\hat{V}$ directly takes too much time, we can sample from and query to its columns. Since rows of S are normalized rows of A, we have sampling access to S with our input data structure. We can translate such samples to samples from $\hat{V}$ using the simple sampling routines discussed in Section 4.1.

Though we could use this access to $\hat{V}$ for the rest of our algorithm, we take a more direct approach. To sample from the ith row of D, $A_i(S^T\hat{U}\hat{\Sigma}^{-1})(S^T\hat{U}\hat{\Sigma}^{-1})^T$, given D's description, we first estimate A_iS^T. This amounts to estimating a constant number of inner products and can be done with sampling access to A_i by Proposition 4.2. Then, we multiply this estimate by $\hat{U}\hat{\Sigma}^{-1}(\hat{\Sigma}^{-1})^T\hat{U}^T$, which is a constant-sized matrix. Finally, we sample from the product of the resulting vector with S and output the result. This step uses rejection sampling: given the ability to sample and query to a constant-sized set of vectors (in this case, rows of S), we can sample from a linear combination of them (Proposition 4.3).

This completes the broad overview of the algorithm. The correctness and runtime analysis is elementary; most of the work is in showing that ModFKV's various outputs truly behave like approximate large singular vectors and values (Proposition 4.6 and Theorem 4.7).

To prove Theorem 2 and show that the quality bounds on the recommendations are the same (see Section 5.3), we just follow Kerenidis and Prakash's analysis and apply the model assumptions and theorems (Section 5) in a straightforward manner. The ℓ^2-norm sampling operations needed to run Algorithm 3 are instantiated with the data structure Kerenidis and Prakash use (Section 3).

1.4 Further Questions

Since this algorithm is associated both with recommendation systems and quantum machine learning, two lines of questioning naturally follow.

First, we can continue to ask whether any quantum machine learning algorithms have provably exponential speedups over classical algorithms. We believe that a potentially enlightening approach is to investigate how state preparation assumptions can be satisfied and whether they are in some way comparable to classical sampling assumptions. After all, we find it unlikely that a quantum exponential speedup can be reinstated just with a better state preparation data structure. However, we are unaware of any research in this area in particular, which could formalize a possible connection between QML algorithms with state preparation assumptions and classical ML algorithms with sampling assumptions.

Second, while the recommendation system algorithm we give is asymptotically exponentially faster than previous algorithms, there are several aspects of this algorithm that make direct application infeasible in practice. First, the model assumptions are somewhat constrictive. It is unclear whether the algorithm still performs well when such assumptions are not satisfied. Second, the exponents and constant factors are large (mostly as a result of using Frieze,

Kannan, and Vempala's algorithm [FKV04]). We believe that the "true" exponents are much smaller, but our technique is fairly roundabout and likely compounds exponents unnecessarily. These issues could be addressed with more straightforward analysis combined with more sophisticated techniques (see, for example, [DV06]).

2 Definitions

Throughout, we obey the following conventions. We assume that basic operations on input data (e.g. adding, multiplying, reading, and writing) take $O(1)$ time. $[n] := \{1, \ldots, n\}$. $f \lesssim g$ denotes the ordering $f = O(g)$ (and correspondingly for $\gtrsim$ and $\eqsim$). For a matrix A, A_i and $A^{(i)}$ will refer to its ith row and column, respectively. $\|A\|_F$ and $\|A\|_2$ will refer to Frobenius and spectral norm, respectively. Norm of a vector v, denoted $\|v\|$, will always refer to ℓ^2-norm. The absolute value of $x \in \mathbb{R}$ will be denoted $|x|$. Occasionally, matrix and vector inequalities of the form $\|x - y\| \leq \varepsilon$ will be phrased in the form $x = y + E$, where $\|E\| \leq \varepsilon$. Thus, the letter E will always refer to some form of perturbation or error.

For a matrix $A \in \mathbb{R}^{m \times n}$, let $A = U\Sigma V^T = \sum_{i=1}^{\min m,n} \sigma_i u_i v_i^T$ be the SVD of A. Here, $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are unitary matrices with columns $\{u_i\}_{i \in [m]}$ and $\{v_i\}_{i \in [n]}$, the left and right singular vectors, respectively. $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal with $\sigma_i := \Sigma_{ii}$ and the σ_i nonincreasing and nonnegative.

We will use the function ℓ to indicate splitting the singular vectors along a singular value:

$$\ell(\lambda) := \max\{i \mid \sigma_i \geq \lambda\}.$$

For example, σ_1 through $\sigma_{\ell(\lambda)}$ gives all of the singular values that are at least λ. This notation suppresses ℓ's dependence on σ_i, but it will always be clear from context.

Π will always refer to an orthogonal projector. That is, if $\beta = \{b_1, \ldots, b_d\}$ is an orthonormal basis for $\operatorname{im} \Pi$, then $\Pi = \sum_{i=1}^{d} b_i b_i^T = BB^T$ for B the matrix whose columns are the elements of β. We will often conflate B, the matrix of basis vectors, and the basis β itself.

2.1 Low-Rank Approximations

We will use various techniques to describe low-rank approximations of A. All of these techniques will involve projecting the rows onto some span of right singular vectors.

$$\begin{aligned} A_k &:= A\Pi_k & \operatorname{im} \Pi_k &:= \operatorname{span}\{v_i \mid i \in [k]\} \\ A_\sigma &:= A\Pi_\sigma & \operatorname{im} \Pi_\sigma &:= \operatorname{span}\{v_i \mid i \in [\ell(\sigma)]\} \end{aligned}$$

A_k and A_σ correspond to the standard notions of low-rank approximations of A. Thus, $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ and is a rank-k matrix minimizing the Frobenius norm distance from A. Similarly, A_σ is just A_t for $t = \ell(\sigma)$. Notice that $\operatorname{rank} A_{\frac{\|A\|_F}{\sqrt{\lambda}}} \leq \lambda$.

We will need to relax this notion for our purposes, and introduce error $\eta \in [0, 1]$. Define $A_{\sigma,\eta} := AP_{\sigma,\eta}$ where $P_{\sigma,\eta}$ is some Hermitian matrix satisfying $\Pi_{\sigma(1+\eta)} \preceq P_{\sigma,\eta} \preceq \Pi_{\sigma(1-\eta)}$ and $\preceq$ is the Loewner order.

In words, $A_{\sigma,\eta}$ is the class of matrices "between" $A_{\sigma(1+\eta)}$ and $A_{\sigma(1-\eta)}$: $P_{\sigma,\eta}$ is the identity on A's singular vectors with value are larger than $\sigma(1+\eta)$, the zero map on A's singular vectors with value at most $\sigma(1-\eta)$, and some PSD matrix with norm at most one on singular vectors with values between $\sigma(1-\eta)$ and $\sigma(1+\eta)$. Such a form of error could arise from having η-like error in estimating the singular values used to compute a low-rank matrix approximation. η should be thought of as constant (1/5 will be the eventual value), and σ should be thought of as very large (say, a constant multiple of $\|A\|_F$), so $A_{\sigma,\eta}$ always has low rank.

2.2 Sampling

For a nonzero vector $x \in \mathbb{R}^n$, we denote by $\mathcal{D}_x$ the distribution over $[n]$ whose probability density function is

$$\mathcal{D}_x(i) = \frac{x_i^2}{\|x\|^2}$$

We will call a sample from $\mathcal{D}_x$ a sample from x.

We make two basic observations. First, $\mathcal{D}_x$ is the distribution resulting from measuring the quantum state $|x\rangle := \frac{1}{\|x\|}\sum x_i |i\rangle$ in the computational basis. Second, sampling access to $\mathcal{D}_x$ makes easy some tasks that are hard given just query access to x. For example, while finding a hidden large entry of $x \in \mathbb{R}^n$ takes $\Omega(n)$ queries with just query access, it takes a constant number of samples with query and sample access.

In all situations, sampling access will be present alongside query access, and accordingly, we will conflate samples $i \sim \mathcal{D}_x$ with the corresponding entries x_i. Note that knowledge of $\|x\|$ is also relevant and useful in this sampling context, since it allows for computing probabilities from $\mathcal{D}_x$ and yet is hard to compute even with query and sampling access to x.

For probability distributions P, Q (as density functions) over a (discrete) universe X, the total variation distance between them is defined as

$$\|P - Q\|_{TV} := \frac{1}{2}\sum_{x \in X} \Big|P(x) - Q(x)\Big|.$$

For a set S, we denote pulling an $s \in S$ uniformly at random by $s \sim_u S$. We will continue to conflate a distribution with its density function.

3 Data Structure

Since we are interested in achieving sublinear bounds for our algorithm, we need to concern ourselves with how the input is given.

In the recommendation systems context, entries correspond to user-product interactions, so we might expect that the input matrix $A \in \mathbb{R}^{m \times n}$ is given as an unordered stream of entries (i, j, A_{ij}). However, if the entries are given in such an unprocessed format, then clearly linear time is required even to parse the input into a usable form. Even when the input is relatively structured (for example, if we are given the known entries of T sorted by row and column),

there is no hope to sample the low-rank approximation of a generic matrix in sublinear time because of the time needed to locate a nonzero entry.

To avoid these issues, we will instead consider our input matrix stored in a low-overhead data structure. We define it first for a vector, then for a matrix.

Lemma 3.1. There exists a data structure storing a vector $v \in \mathbb{R}^n$ with w nonzero entries in $O(w \log(n))$ space, supporting the following operations:

- Reading and updating an entry of v in $O(\log n)$ time;
- Finding $\|v\|^2$ in $O(1)$ time;
- Sampling from $\mathcal{D}_v$ in $O(\log n)$ time.

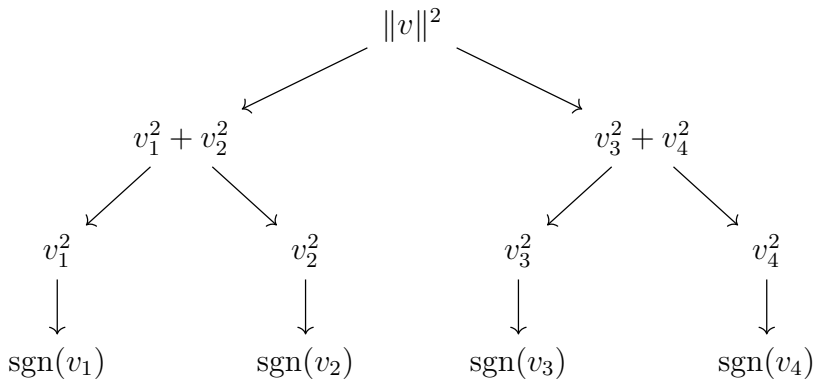

Figure 1: Binary search tree (BST) data structure for $v \in \mathbb{R}^4$. The leaf nodes store v_i via its weight v_i^2 and sign $\operatorname{sgn}(v_i)$, and the weight of an interior node is just the sum of the weights of its children. To update an entry, update all of the nodes above the corresponding leaf. To sample from $\mathcal{D}_v$, start from the top of the tree and randomly recurse on a child with probability proportional to its weight. To take advantage of sparsity, prune the tree to only nonzero nodes.

Proposition 3.2. *Consider a matrix $A \in \mathbb{R}^{m \times n}$. Let $\tilde{A} \in \mathbb{R}^m$ be a vector whose ith entry is $\|A_i\|$. There exists a data structure storing a matrix $A \in \mathbb{R}^{m \times n}$ with w nonzero entries in $O(w \log mn)$ space, supporting the following operations:*

- *Reading and updating an entry of A in $O(\log mn)$ time;*
- *Finding $\tilde{A}_i$ in $O(\log m)$ time;*
- *Finding $\|A\|_F^2$ in $O(1)$ time;*
- *Sampling from $\mathcal{D}_{\tilde{A}}$ and $\mathcal{D}_{A_i}$ in $O(\log mn)$ time.*

This can be done by having a copy of the data structure specified by Lemma 3.1 for each row of A and $\tilde{A}$ (which we can think of as the roots of the BSTs for A's rows). This has all of the desired properties, and in fact, is the data structure Kerenidis and Prakash use

to prepare arbitrary quantum states (Theorem A.1 in [KP17b]). Thus, our algorithm can operate on the same input, although any data structure supporting the operations detailed in Proposition 3.2 will also suffice.

This data structure and its operations are not as ad hoc as they might appear. The operations listed above appear in other work as an effective way to endow a matrix with ℓ^2-norm sampling assumptions (see [DKR02] and [FKV04]).

4 Main Algorithm

Our goal is to prove Theorem 1:

Theorem (1)**.** *There is a classical algorithm that, given a matrix A with query and sampling assumptions as described in Proposition 3.2, along with a row $i \in [m]$, threshold σ, $\eta \in (0, 1]$, and sufficiently small $\varepsilon > 0$, has an output distribution ε-close in total variation distance to $\mathcal{D}_{D_i}$ where $D \in \mathbb{R}^{m \times n}$ satisfies $\|D - A_{\sigma,\eta}\|_F \leq \varepsilon \|A\|_F$ for some $A_{\sigma,\eta}$, in query and time complexity*

$$O\left(\text{poly}\left(\frac{\|A\|_F}{\sigma}, \frac{1}{\varepsilon}, \frac{1}{\eta}, \frac{\|A_i\|}{\|D_i\|}\right)\right).$$

We present the algorithm (Algorithm 3) and analysis nonlinearly. First, we give two algorithms that use ℓ^2-norm sampling access to their input vectors to perform basic linear algebra. Second, we present ModFKV, a sampling algorithm to find the description of a low-rank matrix approximation. Third, we use the tools we develop to get from this description to the desired sample.

4.1 Vector Sampling

Recall how we defined sampling from a vector.

Definition. For a vector $x \in \mathbb{R}^n$, we denote by $\mathcal{D}_x$ the distribution over $[n]$ with density function $\mathcal{D}_x(i) = x_i^2/\|x\|^2$. We call a sample from $\mathcal{D}_x$ a sample from x.

We will need that closeness of vectors in ℓ^2-norm implies closeness of their respective distributions in TV distance:

Lemma 4.1. For $x, y \in \mathbb{R}^n$ satisfying $\|x - y\| \leq \varepsilon$, the corresponding distributions $\mathcal{D}_x$, $\mathcal{D}_y$ satisfy $\|\mathcal{D}_x - \mathcal{D}_y\|_{TV} \leq 2\varepsilon/\|x\|$.

Proof. Let $\hat{x}$ and $\hat{y}$ be the normalized vectors $x/\|x\|$ and $y/\|y\|$.

$$\begin{aligned}\|\mathcal{D}_x - \mathcal{D}_y\|_{TV} &= \frac{1}{2}\sum_{i=1}^{n} \left|\hat{x}_i^2 - \hat{y}_i^2\right| = \frac{1}{2}\left\langle |\hat{x} - \hat{y}|, |\hat{x} + \hat{y}| \right\rangle \leq \frac{1}{2}\|\hat{x} - \hat{y}\|\|\hat{x} + \hat{y}\| \\ &\leq \|\hat{x} - \hat{y}\| = \frac{1}{\|x\|}\left\|x - y - (\|x\| - \|y\|)\hat{y}\right\| \leq \frac{1}{\|x\|}\Big(\|x - y\| + \big|\|x\| - \|y\|\big|\Big) \leq \frac{2\varepsilon}{\|x\|}\end{aligned}$$

The first inequality follows from Cauchy-Schwarz, and the rest follow from triangle inequality. □

Now, we give two subroutines that can be performed, assuming some vector sampling access. First, we show that we can estimate the inner product of two vectors well.

Proposition 4.2. *Given query access to $x, y \in \mathbb{R}^n$, sample access to $\mathcal{D}_x$, and knowledge of $\|x\|$, $\langle x, y\rangle$ can be estimated to additive error $\|x\|\|y\|\varepsilon$ with at least $1-\delta$ probability using $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ queries and samples (and the same time complexity).*

Proof. Perform samples in the following way: for each i, let the random variable Z be y_i/x_i with probability $x_i^2/\|x\|^2$ (so we sample i from $\mathcal{D}_x$). We then have:

$$\mathrm{E}[Z] = \sum \frac{y_i}{x_i}\frac{x_i^2}{\|x\|^2} = \frac{\sum x_i y_i}{\|x\|^2} = \frac{\langle x, y\rangle}{\|x\|^2},$$
$$\mathrm{Var}[Z] \le \sum \left(\frac{y_i}{x_i}\right)^2 \frac{x_i^2}{\|x\|^2} = \frac{\sum y_i^2}{\|x\|^2} = \frac{\|y\|^2}{\|x\|^2}.$$

Since we know $\|x\|$, we can normalize by it to get a random variable whose mean is $\langle x, y\rangle$ and whose standard deviation is $\sigma = \|x\|\|y\|$.

The rest follows from standard techniques: we take the median of $6\log\frac{1}{\delta}$ copies of the mean of $\frac{9}{2\varepsilon^2}$ copies of Z to get within $\varepsilon\sigma = \varepsilon\|x\|\|y\|$ of $\langle x, y\rangle$ with probability at least $1-\delta$ in $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ accesses. All of the techniques used here take linear time. □

Second, we show that, given sample access to some vectors, we can sample from a linear combination of them.

Proposition 4.3. *Suppose we are given query and sample access to the columns of $V \in \mathbb{R}^{n\times k}$, along with their norms. Then given $w \in \mathbb{R}^k$ (as input), we can output a sample from Vw in $O(k^2 C(V, w))$ expected query complexity and expected time complexity, where*

$$C(V, w) := \frac{\sum_{i=1}^k \|w_i V^{(i)}\|^2}{\|Vw\|^2}.$$

C measures the amount of cancellation for Vw. For example, when the columns of V are orthogonal, $C = 1$ for all nonzero w, since there is no cancellation. Conversely, when the columns of V are linearly dependent, there is a choice of nonzero w such that $\|Vw\| = 0$, maximizing cancellation. In this context, C is undefined, which matches with sampling from the zero vector also being undefined. By perturbing w we can find vectors requiring arbitrarily large values of C.

Proof. We use rejection sampling: see Algorithm 1. Given sampling access to a distribution P, rejection sampling allows for sampling from a "close" distribution Q, provided we can compute some information about their corresponding distributions.

If $r_i \le 1$ for all i, then the above procedure is well-defined and outputs a sample from Q in M iterations in expectation.[3]

[3]The number of iterations is a geometric random variable, so this can be converted into a bound

Algorithm 1: Rejection Sampling

Pull a sample s from P;
Compute $r_s = \frac{Q(s)}{MP(s)}$ for some constant M;
Output s with probability r_s and restart otherwise;

In our case, P is the distribution formed by first sampling a row j with probability proportional to $\|w_j V^{(j)}\|^2$ and then sampling from $\mathcal{D}_{V^{(j)}}$; Q is the target $\mathcal{D}_{Vw}$. We choose

$$r_i = \frac{(Vw)_i^2}{k\sum_{j=1}^k (V_{ij}w_j)^2},$$

which we can compute in k queries[4]. This expression is written in a way the algorithm can directly compute, but it can be put in the form of the rejection sampling procedures stated above:

$$M = \frac{Q(i)k\sum_{j=1}^k (V_{ij}w_j)^2}{P(i)(Vw)_i^2} = \frac{k(\sum_{j=1}^k \|w_j V^{(j)}\|^2)}{\|Vw\|^2} = kC(V,w).$$

M is independent of i, so it is a constant as desired. To prove correctness, all we need to show is that our choice of r_i is always at most 1. This follows from Cauchy-Schwarz:

$$r_i = \frac{(Vw)_i^2}{k\sum_{j=1}^k (V_{ij}w_j)^2} = \frac{(\sum_{j=1}^k V_{ij}w_j)^2}{k\sum_{j=1}^k (V_{ij}w_j)^2} \leq 1.$$

Each iteration of the procedure takes $O(k)$ queries, leading to a query complexity of $O(k^2 C(V,w))$. Time complexity is linear in the number of queries. □

4.2 Finding a Low-Rank Approximation

Now, we describe the low-rank approximation algorithm that we use at the start of the main algorithm.

Theorem 4.4. *Given a matrix $A \in \mathbb{R}^{m\times n}$ supporting the sample and query operations described in Proposition 3.2, along with parameters $\sigma \in (0, \|A\|_F], \varepsilon \in (0, \sqrt{\sigma/\|A\|_F}/4], \eta \in [\varepsilon^2, 1]$, there is an algorithm that outputs a succinct description (of the form described below) of some D satisfying $\|D - A_{\sigma,\eta}\|_F \leq \varepsilon\|A\|_F$ with probability at least $1-\delta$ and*

$$O\left(\text{poly}\left(\frac{\|A\|_F^2}{\sigma^2}, \frac{1}{\varepsilon}, \frac{1}{\eta}, \log\frac{1}{\delta}\right)\right)$$

query and time complexity.

guaranteeing a sample in $M \log 1/\delta$ iterations with failure probability $1-\delta$, provided the algorithm knows M. All expected complexity bounds we deal with can be converted to high probability bounds in the manner described.

[4]Notice that we can compute r_i without directly computing the probabilities $Q(i)$. This helps us because computing $Q(i)$ involves computing $\|Vw\|$, which is nontrivial.

To prove this theorem, we modify the algorithm given by Frieze, Kannan, and Vemapa [FKV04] and show that it satisfies the desired properties. The modifications are not crucial to the correctness of the full algorithm: without them, we simply get a different type of low-rank approximation bound. They come into play in Section 5 when proving guarantees about the algorithm as a recommendation system.

Algorithm 2: ModFKV

Input: Matrix $A \in \mathbb{R}^{m \times n}$ supporting operations in Proposition 3.2, threshold σ, error parameters ε, η

Output: A description of an output matrix D

Set $K = \|A\|_F^2/\sigma^2$ and $\bar{\varepsilon} = \eta\varepsilon^2$;

Set $q = \Theta\big(\frac{K^4}{\bar{\varepsilon}^2}\big)$;

Sample rows $i_1, \ldots, i_q$ from $\mathcal{D}_{\tilde{A}}$;

Let $\mathcal{F}$ denote the distribution given by choosing an $s \sim_u [q]$, and choosing a column from $\mathcal{D}_{A_{i_s}}$;

Sample columns $j_1, \ldots, j_q$ from $\mathcal{F}$;

Let W be the resulting $q \times q$ row-and-column-normalized submatrix $W_{rc} := \frac{A_{i_r j_c}}{q\sqrt{\mathcal{D}_{\tilde{A}}(i_r)\mathcal{F}(j_c)}}$;

Compute the left singular vectors of W $u^{(1)}, \ldots, u^{(k)}$ that correspond to singular values $\sigma^{(1)}, \ldots, \sigma^{(k)}$ larger than σ;

Output $i_1, \ldots, i_q$, $\hat{U} \in \mathbb{R}^{q \times k}$ the matrix whose ith column is $u^{(i)}$, and $\hat{\Sigma} \in \mathbb{R}^{k \times k}$ the diagonal matrix whose ith entry on the diagonal is $\sigma^{(i)}$. This is the description of the output matrix D;

The algorithm, ModFKV, is given in Algorithm 2. It subsamples the input matrix, computes the subsample's large singular vectors and values, and outputs them with the promise that they give a good description of the singular vectors of the full matrix. We present the algorithm as the original work does, aiming for a constant failure probability. This can be amplified to δ failure probability by increasing q by a factor of $O(\log \frac{1}{\delta})$ (the proof is slightly nontrivial, using a martingale inequality; see Theorem 1 of [DKM06]). More of the underpinnings are explained in Frieze, Kannan, and Vempala's paper [FKV04].

We get the output matrix D from its description in the following way. Let S be the submatrix given by restricting the rows to $i_1, \ldots, i_q$ and renormalizing row i by $1/\sqrt{q\mathcal{D}_{\tilde{A}}(i)}$ (so they all have the same norm). Then $\hat{V} := S^T\hat{U}\hat{\Sigma}^{-1} \in \mathbb{R}^{n \times k}$ is our approximation to the large right singular vectors of A; this makes sense if we think of S, $\hat{U}$, and $\hat{\Sigma}$ as our subsampled low-rank approximations of A, U, and Σ (from A's SVD). Appropriately, D is the "projection" of A onto the span of $\hat{V}$:

$$D := A\hat{V}\hat{V}^T = AS^T\hat{U}\hat{\Sigma}^{-2}\hat{U}^TS.$$

The query complexity of ModFKV is dominated by querying all of the entries of W, which is $O(q^2)$, and the time complexity is dominated by computing W's SVD, which is $O(q^3)$. We can convert this to the input parameters using that $q = O(\frac{\|A\|^8}{\sigma^8\varepsilon^4\eta^2})$.

ModFKV differs from FKV only in that σ is taken as input instead of k, and is used as the threshold for the singular vectors. As a result of this change, K replaces k in the subsampling steps, and σ replaces k in the SVD step. Notice that the number of singular vectors taken (denoted k) is at most K, so in effect, we are running FKV and just throwing away some of the smaller singular vectors. Because we ignore small singular values that FKV had to work to find, we can sample a smaller submatrix, speeding up our algorithm while still achieving an analogous low-rank approximation bound:

Lemma 4.5. The following bounds hold for the output matrix D (here, k is the width of $\hat{V}$, and thus a bound on rank D):

$$\|A - D\|_F^2 \leq \|A - A_k\|_F^2 + \varepsilon\|A\|_F^2 \tag{♢}$$

$$\text{and } \ell((1 + \bar{\varepsilon}\sqrt{K})\sigma) \leq k \leq \ell((1 - \bar{\varepsilon}\sqrt{K})\sigma) \tag{♡}$$

The following property will be needed to prove correctness of ModFKV and Algorithm 3. The estimated singular vectors in $\hat{V}$ behave like singular vectors, in that they are close to orthonormal.

Proposition 4.6. *The output vectors $\hat{V}$ satisfy*

$$\|\hat{V} - \Lambda\|_F = O(\bar{\varepsilon})$$

for Λ a set of orthonormal vectors with the same image as $\hat{V}$.

As an easy corollary, $\hat{V}\hat{V}^T$ is $O(\bar{\varepsilon})$-close in Frobenius norm to the projector $\Lambda\Lambda^T$, since $\hat{V}\hat{V}^T = (\Lambda + E)(\Lambda + E)^T$ and $\|\Lambda E^T\|_F = \|E\Lambda^T\|_F = \|E\|_F$. The proofs of the above lemma and proposition delve into FKV's analysis, so we defer them to the appendix.

The guarantee on our output matrix D is (♢), but for our recommendation system, we want that D is close to some $A_{\sigma,\eta}$. Now, we present the core theorem showing that the former kind of error implies the latter.

Theorem 4.7. *If Π a k-dimensional orthogonal projector satisfies*

$$\|A_k\|_F^2 \leq \|A\Pi\|_F^2 + \varepsilon\sigma_k^2,$$

then

$$\|A\Pi - A_{\sigma_k,\eta}\|_F^2 \lesssim \varepsilon\sigma_k^2/\eta,$$

where $\varepsilon \leq \eta \leq 1$.[5]

The proof is somewhat involved, so we defer it to the appendix. To our knowledge, this is a novel translation of a typical FKV-type bound as in (♢) to a new, useful type of bound, so we believe this theorem may find use elsewhere. Now, we use this theorem to show that D is close to some $A_{\sigma,\eta}$.

Corollary 4.8. $\|D - A_{\sigma,\eta}\|_F \lesssim \varepsilon\|A\|_F/\sqrt{\eta}$.

[5] An analogous proof gives the more general bound $\|\Pi - \Pi_{\sigma,\eta}\|_F^2 \lesssim \varepsilon/\eta$.

Proof. Throughout the course of this proof, we simplify and apply theorems based on the restrictions on the parameters in Theorem 4.4.

First, notice that the bound ($\diamondsuit$) can be translated to the type of bound in the premise of Theorem 4.7, using Proposition 4.6.

$$\begin{aligned}
\|A - D\|_F^2 &\leq \|A - A_k\|_F^2 + \bar{\varepsilon}\|A\|_F^2 \\
\|A - A(\Pi + E)\|_F^2 &\leq \|A - A_k\|_F^2 + \bar{\varepsilon}\|A\|_F^2 \\
(\|A - A\Pi\|_F - \bar{\varepsilon}\|A\|_F)^2 &\lesssim \|A - A_k\|_F^2 + \bar{\varepsilon}\|A\|_F^2 \\
\|A - A\Pi\|_F^2 &\lesssim \|A - A_k\|_F^2 + \bar{\varepsilon}\|A\|_F^2 \\
\|A\|^2 - \|A\Pi\|_F^2 &\lesssim \|A\|^2 - \|A_k\|_F^2 + \bar{\varepsilon}\|A\|_F^2 \\
\|A_k\|_F^2 &\lesssim \|A\Pi\|_F^2 + (\bar{\varepsilon}\|A\|_F^2/\sigma_k^2)\sigma_k^2
\end{aligned}$$

The result of the theorem is that

$$\left\| A\Pi - A_{\sigma_k, \frac{\eta - \bar{\varepsilon}\sqrt{K}}{1 - \bar{\varepsilon}\sqrt{K}}} \right\|_F^2 \lesssim \frac{(\bar{\varepsilon}\|A\|_F^2/\sigma_k^2)\sigma_k^2}{\frac{\eta - \bar{\varepsilon}\sqrt{K}}{1 - \bar{\varepsilon}\sqrt{K}}} \lesssim \frac{\bar{\varepsilon}}{\eta}\|A\|_F^2.$$

The bound on k ($\heartsuit$) implies that any $A_{\sigma_k, \frac{\eta - \bar{\varepsilon}\sqrt{K}}{1 - \bar{\varepsilon}\sqrt{K}}}$ is also an $A_{\sigma,\eta}$ (the error of the former is contained in the latter), so we can conclude

$$\|D - A_{\sigma,\eta}\|_F = \|A(\Pi + E) - A_{\sigma,\eta}\|_F \lesssim \|A\Pi - A_{\sigma,\eta}\|_F + \bar{\varepsilon}\|A\|_F \lesssim \sqrt{\frac{\bar{\varepsilon}}{\eta}}\|A\|_F.$$

$\bar{\varepsilon}$ was chosen so that the final term is bounded by $\varepsilon\|A\|_F$. □

This completes the proof of Theorem 4.4. To summarize, after this algorithm we are left with the description of our low-rank approximation $D = AS^T\hat{U}\Sigma^{-2}\hat{U}^TS$, which consists of the following:

- $\hat{U} \in \mathbb{R}^{q \times k}$, explicit orthonormal vectors;
- $\hat{\Sigma} \in \mathbb{R}^{k \times k}$, an explicit diagonal matrix whose diagonal entries are in $(\sigma, \|A\|_F]$;
- $S \in \mathbb{R}^{q \times n}$, which is *not* output explicitly, but whose rows are rows of A normalized to equal norm $\|A\|_F/\sqrt{q}$ (so we can sample from S's rows); and
- $\hat{V} \in \mathbb{R}^{n \times k}$, a close-to-orthonormal set of vectors implicitly given as $S^T\hat{U}\hat{\Sigma}^{-1}$.

This description will suffice to generate samples from rows of D.

4.3 Proof of Theorem 1

Theorem 1. *There is a classical algorithm that, given a matrix A with query and sampling assumptions as described in Proposition 3.2, along with a row $i \in [m]$, threshold σ, $\eta \in (0, 1]$, and sufficiently small $\varepsilon > 0$, has an output distribution ε-close in total variation distance*

to $\mathcal{D}_{D_i}$ where $D \in \mathbb{R}^{m\times n}$ satisfies $\|D - A_{\sigma,\eta}\|_F \le \varepsilon\|A\|_F$ for some $A_{\sigma,\eta}$, in query and time complexity

$$O\left(\text{poly}\left(\frac{\|A\|_F}{\sigma}, \frac{1}{\varepsilon}, \frac{1}{\eta}, \frac{\|A_i\|}{\|D_i\|}\right)\right).$$

Proof. We will give an algorithm (Algorithm 3) where the error in the output distribution is $O(\varepsilon\|A_i\|/\|D_i\|)$-close to $\mathcal{D}_{D_i}$, and there is no dependence on $\|A_i\|/\|D_i\|$ in the runtime, and discuss later how to modify the algorithm to get the result in the theorem.

Algorithm 3: Low-rank approximation sampling

Input: Matrix $A \in \mathbb{R}^{m\times n}$ supporting the operations in 3.2, user $i \in [m]$, threshold σ, $\varepsilon > 0$, $\eta \in (0, 1]$

Output: Sample $s \in [n]$

Run ModFKV (2) with parameters $(\sigma, \varepsilon, \eta)$ to get a description of $D = A\hat{V}\hat{V}^T = AS^T\hat{U}\hat{\Sigma}^{-2}\hat{U}^TS$;

Estimate A_iS^T entrywise by using Proposition 4.2 with parameter $\frac{\varepsilon}{\sqrt{K}}$ to estimate $\langle A_i, S_t^T\rangle$ for all $t \in [q]$. Let est be the resulting $1 \times q$ vector of estimates;

Compute est$\hat{U}\hat{\Sigma}^{-2}\hat{U}^T$ with matrix-vector multiplication;

Sample s from (est$\hat{U}\hat{\Sigma}^{-2}\hat{U}^T)S$ using Proposition 4.3;

Output s;

Correctness: By Theorem 4.4, for sufficiently small[6] ε, the output matrix D satisfies

$$\|D - A_{\sigma,\eta}\|_F \le \varepsilon\|A\|_F.$$

So, all we need is to approximately sample from the ith row of D, given its description.

Recall that the rows of S_t^T have norm $\|A\|_F/\sqrt{q}$. Thus, the guarantee from Proposition 4.2 states that each estimate of an entry has error at most $\frac{\varepsilon}{\sqrt{Kq}}\|A_i\|\|A\|_F$, meaning that $\|\text{est} - A_iS^T\| \le \frac{\varepsilon}{\sqrt{K}}\|A_i\|\|A\|_F$. Further, using that $\hat{V}$ is close to orthonormal (Proposition 4.6) and $\|\hat{U}\Sigma^{-1}\| \le \frac{1}{\sigma}$, we have that the vector we sample from is close to D_i:

$$\begin{aligned}\|(\text{est} - A_iS^T)\hat{U}\Sigma^{-1}\hat{V}^T\| &\le (1 + O(\varepsilon^4))\|\text{est}\hat{U}\Sigma^{-1} - A_iS^T\hat{U}\Sigma^{-1}\| \\ &\lesssim \frac{1}{\sigma}\|\text{est} - A_iS^T\| \le \frac{\varepsilon}{\sigma\sqrt{K}}\|A_i\|\|A\|_F = \varepsilon\|A_i\|\end{aligned}$$

Finally, by Lemma 4.1, we get the desired bound: that the distance from the output distribution to $\mathcal{D}_{D_i}$ is $O(\varepsilon\|A_i\|/\|D_i\|)$.

Runtime: Applying Proposition 4.2 q times takes $O(\frac{Kq}{\varepsilon^2}\log\frac{q}{\delta})$ time; the naive matrix-vector

[6]This is not a strong restriction: $\varepsilon \lesssim \min\{\sqrt{\eta}, \sqrt{\sigma/\|A\|_F}\}$ works. This makes sense: for ε any larger, the error can encompass addition or omission of full singular vectors.

multiplication takes $O(Kq)$ time; and applying Proposition 4.3 takes time $O(Kq^2)$, since

$$\begin{aligned} C(S^T, \hat{U}\hat{\Sigma}^{-2}\hat{U}^T\text{est}^T) &= \frac{\sum_{j=1}^{q} \|(\text{est}\hat{U}\hat{\Sigma}^{-2}\hat{U}^T)_j S_j\|^2}{\|\text{est}\hat{U}\hat{\Sigma}^{-2}\hat{U}^T S\|^2} \\ &\leq \frac{\|\text{est}\hat{U}\hat{\Sigma}^{-2}\hat{U}^T\|^2 \|S\|_F^2}{\|\text{est}\hat{U}\hat{\Sigma}^{-2}\hat{U}^T S\|^2} \\ &\leq \frac{\|\hat{\Sigma}^{-1}\hat{U}^T\|^2 \|S\|_F^2}{\min_{x:\|x\|=1} \|x\Sigma^{-1}\hat{U}^T S\|^2} \\ &\leq \frac{\|A\|_F^2}{\sigma^2(1-\varepsilon^4)^2} = O(K) \end{aligned}$$

using Cauchy-Schwarz, Proposition 4.6, and the basic facts about D's description[7].

The query complexity is dominated by the use of Proposition 4.3, resulting in

$$\tilde{O}\left(\frac{\|A\|^2}{\sigma^2}\left(\frac{\|A\|^8}{\sigma^8\varepsilon^4\eta^2}\right)^2\right) = \tilde{O}\left(\frac{\|A\|^{18}}{\sigma^{18}\varepsilon^8\eta^4}\right),$$

where the $\tilde{O}$ hides the log factors incurred by amplifying the failure probability to δ. The time complexity is dominated by the $O(q^3)$ SVD computation in ModFKV in $O(q^3)$ time, resulting in

$$\tilde{O}\left(\frac{\|A\|^{24}}{\sigma^{24}\varepsilon^{12}\eta^6}\right).$$

□

Finally, we briefly discuss variants of this algorithm.

- To get the promised bound in the theorem statement, we can repeatedly estimate $A_i S^T$ (creating est_1, est_2, etc.) with exponentially decaying ε, eventually reducing the error of the first step to $O(\varepsilon\|A_i S^T\|)$. This procedure decreases the total variation error to $O(\varepsilon)$ and increases the runtime by $\tilde{O}(\|A_i\|^2/\|D_i\|^2)$, as desired. Further, we can ignore δ by choosing $\delta = \varepsilon$ and outputting an arbitrary $s \in [n]$ upon failure. This only changes the output distribution by ε in total variation distance and increase runtime by polylog $\frac{1}{\varepsilon}$.
- While the input is a row $i \in [m]$ (and thus supports query and sampling access), it need not be. More generally, given query and sample access to orthonormal vectors $V \in \mathbb{R}^{n\times k}$, and query access to $x \in \mathbb{R}$, one can approximately sample from a distribution $O(\varepsilon\|x\|/\|VV^T x\|)$-close to $\mathcal{D}_{VV^T x}$, the projection of x onto the span of V, in $O(\frac{k^2}{\varepsilon^2}\log\frac{k}{\delta})$ time.
- While the SVD dominates the time complexity of Algorithm 3, the same description output by ModFKV can be used for multiple recommendations, amortizing the cost down to the query complexity (since the rest of the algorithm is linear in the number of queries).

[7]We have just proved that, given D's description, we can sample from any vector of the form $\hat{V}x$ in $O(Kq^2)$ time.

5 Application to Recommendations

We now go through the relevant assumptions necessary to apply Theorem 1 to the recommendation systems context. As mentioned above, these are the same assumptions as those in [KP17b]: an exposition of these assumptions is also given there. Then, we prove Theorem 2, which shows that Algorithm 3 gives the same guarantees on recommendations as the quantum algorithm.

5.1 Preference Matrix

Recall that given m users and n products, the preference matrix $T \in \mathbb{R}^{m \times n}$ contains the complete information on user-product preferences. For ease of exposition, we will assume the input data is binary:

Definition. If user i likes product j, then $T_{ij} = 1$. If not, $T_{ij} = 0$.

We can form such a preference matrix from generic data about recommendations, simply by condensing information down to the binary question of whether a product is a good recommendation or not.[8]

We are typically given only a small subsample of entries of T (which we learn when a user purchases or otherwise interacts with a product). Then, finding recommendations for user i is equivalent to finding large entries of the ith row of T given such a subsample.

Obviously, without any restrictions on what T looks like, this problem is ill-posed. We make this problem tractable through the standard assumption that T is close to a matrix of small rank k.

***T* is close to a low-rank matrix.** That is, $\|T - T_k\|_F \leq \rho \|T\|_F$ for some k and $\rho \ll 1$. k should be thought of as constant (at worst, $\text{polylog}(m, n)$).

This standard assumption comes from the intuition that users decide their preference for products based on a small number of factors (e.g. price, quality, and popularity) [DKR02; Aza+01; KBV09].

The low-rank assumption gives T robust structure; that is, only given a small number of entries, T can be reconstructed fairly well.

Many users have approximately the same number of preferences. The low-rank assumption is enough to get some bound on quality of recommendations (see Lemma 3.2 in [KP17b]). However, this bound considers "matrix-wide" recommendations. We would like to give a bound on the probability that an output is a good recommendation *for a particular user*.

It is not enough to assume that $\|T - T_k\|_F \leq \rho \|T\|_F$. In a worst-case scenario, a few users make up the vast majority of the recommendations (say, a few users like every product, and

[8] This algorithm makes no distinction between binary matrices and matrices with values in the interval $[0, 1]$, and the corresponding analysis is straightforward upon defining a metric for success when data is nonbinary.

the rest of the users are only happy with four products). Then, even if we reconstruct T_k exactly, the resulting error, $\rho\|T\|_F$, can exceed the mass of recommendations in the non-heavy users, drowning out any possible information about the vast majority of users that could be gained from the low-rank structure.

In addition to being pathological for user-specific bounds, this scenario is orthogonal to our primary concerns: we aren't interested in providing recommendations to users who desire very few products or who desire nearly all products, since doing so is intractable and trivial, respectively. To avoid considering such a pathological case, we restrict our attention to the "typical user":

Definition. For $T \in \mathbb{R}^{m\times n}$, call $S \subset [m]$ a subset of users (γ, ζ)-*typical* (where $\gamma > 0$ and $\zeta \in [0, 1)$) if $|S| \geq (1-\zeta)m$ and, for all $i \in S$,

$$\frac{1}{1+\gamma}\frac{\|T\|_F^2}{m} \leq \|T_i\|^2 \leq (1+\gamma)\frac{\|T\|_F^2}{m}.$$

γ and ζ can be chosen as desired to broaden or restrict our idea of typical. We can enforce good values of γ and ζ simply by requiring that users have the same number of good recommendations; this can be done by defining a good recommendation to be the top 100 products for a user, regardless of utility to the user.

Given this definition, we can give a guarantee on recommendations for typical users that come from an approximate reconstruction of T.

Theorem 5.1. *For $T \in \mathbb{R}^{m\times n}$, S a (γ, ζ)-typical set of users, and a matrix $\tilde{T}$ satisfying $\|T - \tilde{T}\|_F \leq \varepsilon\|T\|_F$,*

$$\mathop{\mathrm{E}}_{i\sim_u S}\left[\|\mathcal{D}_{T_i} - \mathcal{D}_{\tilde{T}_i}\|_{TV}\right] \leq \frac{2\varepsilon\sqrt{1+\gamma}}{1-\zeta}.$$

Further, for a chosen parameter $\psi \in (0, 1-\zeta)$ there exists some $S' \subset S$ of size at least $(1-\psi-\zeta)m$ such that, for $i \in S'$,

$$\|\mathcal{D}_{T_i} - \mathcal{D}_{\tilde{T}_i}\|_{TV} \leq 2\varepsilon\sqrt{\frac{1+\gamma}{\psi}}.$$

The first bound is an average-case bound on typical users and the second is a strengthening of the resulting Markov bound. Both bound total variation distance from $\mathcal{D}_{T_i}$, which we deem a good goal distribution to sample from for recommendations[9]. We defer the proof of this theorem to the appendix.

When we don't aim for a particular distribution and only want to bound the probability of giving a bad recommendation, we can prove a stronger average-case bound on the failure probability.

[9]When T is not binary, this means that if product X is λ times more preferable than product Y, then it will be chosen as a recommendation λ^2 times more often. By changing how we map preference data to actual values in T, this ratio can be increased. That way, we have a better chance of selecting the best recommendations, which approaches like matrix completion can achieve. However, these transformations must also preserve that T is close-to-low-rank.

Theorem 5.2 (Theorem 3.3 of [KP17b]). *For $T \in \mathbb{R}^{m\times n}$ a* binary *preference matrix, S a (γ,ζ)-typical set of users, and a matrix $\tilde{T}$ satisfying $\|T-\tilde{T}\|_F \le \varepsilon\|T\|_F$, for a chosen parameter $\psi \in (0, 1-\zeta)$ there exists some $S' \subset S$ of size at least $(1-\psi-\zeta)m$ such that*

$$\Pr_{\substack{i\sim_u S'\\ j\sim\tilde{T}_i}}[(i,j) \text{ is bad}] \le \frac{\varepsilon^2(1+\varepsilon)^2}{(1-\varepsilon)^2\left(1/\sqrt{1+\gamma}-\varepsilon/\sqrt{\psi}\right)^2(1-\psi-\zeta)}.$$

For intuition, if ε is sufficiently small compared to the other parameters, this bound becomes $O(\varepsilon^2(1+\gamma)/(1-\psi-\zeta))$. The total variation bound from Theorem 5.1 is not strong enough to prove this: the failure probability we would get is $2\varepsilon\sqrt{1+\gamma}/(1-\psi-\zeta)$. Accounting for T being binary gives the extra ε factor.

We know k. More accurately, a rough upper bound for k will suffice. Such an upper bound can be guessed and tuned from data.

In summary, we have reduced the problem of "find a good recommendation for a user" to "given some entries from a close-to-low-rank matrix T, sample from $\tilde{T}_i$ for some $\tilde{T}$ satisfying $\|T-\tilde{T}\|_F \le \varepsilon\|T\|_F$ for small ε."

5.2 Matrix Sampling

We have stated our assumptions on the full preference matrix T, but we also need assumptions on the information we are given about T. Even though we will assume we have a constant fraction of data about T, this does not suffice for good recommendations. For example, if we are given the product-preference data for only half of our products, we have no hope to give good recommendations on the other half.

We will use a model for subsampling for matrix reconstruction given by Achlioptas and McSherry [AM07]. In this model, the entries we are given are chosen uniformly over all entries. This model has seen use previously in the theoretical recommendation systems literature [DKR02]. Specifically, we have the following:

Definition. For a matrix $T \in \mathbb{R}^{m\times n}$, let $\hat{T}$ be a random matrix i.i.d. on its entries, where

$$\hat{T}_{ij} = \begin{cases} \frac{T_{ij}}{p} & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}. \qquad (\clubsuit)$$

Notice that $E[\hat{T}] = T$. When the entries of T are bounded, $\hat{T}$ is T perturbed by a random matrix E whose entries are independent and bounded random variables. Standard concentration inequalities imply that such random matrices don't have large singular values (the largest singular value is, say, $O(\sqrt{n/p})$). Thus, for some vector v, if $\|Tv\|/\|v\|$ is large (say, $O(\sqrt{mn/k})$), then $\|(T+E)v\|/\|v\|$ will still be large, despite E having large Frobenius norm.

The above intuition suggests that when T has large singular values, its low-rank approximation T_k is not perturbed much by E, and thus, low-rank approximations of $\hat{T}$ are good reconstructions of T. A series of theorems by Achlioptas and McSherry [AM07] and Kerenidis

and Prakash [KP17b] formalizes this intuition. For brevity, we only describe a simplified form of the last theorem in this series, which is the version they (and we) use for analysis. It states that, under appropriate circumstances, it's enough to compute $\hat{T}_{\sigma,\eta}$ for appropriate σ and η.

Theorem 5.3 (4.3 of [KP17b])**.** *Let $T \in \mathbb{R}^{m\times n}$ and let $\hat{T}$ be the random matrix defined in (♣), with $p \geq \frac{3\sqrt{nk}}{2^{9/2}\varepsilon^3\|T\|_F}$ and $\max_{ij}|T_{ij}| = 1$. Let $\sigma = \frac{5}{6}\sqrt{\frac{\varepsilon^2 p}{8k}}\|\hat{T}\|_F$, let $\eta = 1/5$, and assume that $\|T\|_F \geq \frac{9}{\sqrt{2}\varepsilon^3}\sqrt{nk}$. Then with probability at least $1 - \exp(-19(\log n)^4)$,*

$$\|T - \hat{T}_{\sigma,\eta}\|_F \leq 3\|T - T_k\|_F + 3\varepsilon\|T\|_F.$$

With this theorem, we have a formal goal for a recommendation systems algorithm. We are given some subsample $A = \hat{T}$ of the preference matrix, along with knowledge of the size of the subsample p, the rank of the preference matrix k, and an error parameter ε. Given that the input satisfies the premises for Theorem 5.3, for some user i, we can provide a recommendation by sampling from $(A_{\sigma,\eta})_i$ with σ, η specified as described. Using the result of this theorem, $A_{\sigma,\eta}$ is close to T, and thus we can use the results of Section 5.1 to conclude that such a sample is likely to be a good recommendation for typical users.

Now, all we need is an algorithm that can sample from $(A_{\sigma,\eta})_i$. Theorem 1 shows that Algorithm 3 is exactly what we need!

5.3 Proof of Theorem 2

Theorem 2. *Suppose we are given $\hat{T}$ in the data structure in Proposition 3.2, where $\hat{T}$ is a subsample of T where p is constant and T satisfies $\|T - T_k\|_F \leq \rho\|T\|_F$ for a known k. Further suppose that the conditions of Theorem 5.3 hold, the bound in the conclusion holds (which is true with probability $\geq 1 - \exp(-19(\log n)^4)$), and we have S a (γ,ζ)-typical set of users with $1-\zeta$ and γ constant. Then, for sufficiently small ε, sufficiently small ρ (at most a function of ζ and γ), and a constant fraction of users $\bar{S} \subset S$, for all $i \in \bar{S}$ we can output samples from a distribution $\mathcal{O}_i$ with probability $1 - (mn)^{-\Theta(1)}$ satisfying*

$$\|\mathcal{O}_i - \mathcal{D}_{T_i}\|_{TV} \lesssim \varepsilon + \rho$$

in $O(\text{poly}(k, 1/\varepsilon)\,\text{polylog}(mn))$ time.

Kerenidis and Prakash's version of this analysis treats γ and ζ with slightly more care, but does eventually assert that these are constants. Notice that we must assume p is constant.

Proof. We just run Algorithm 3 with parameters as described in Proposition 5.3: $\sigma = \frac{5}{6}\sqrt{\frac{\varepsilon^2 p}{8k}}\|A\|_F$, ε, $\eta = 1/5$. Provided $\varepsilon \lesssim \sqrt{p/k}$, the result from Theorem 1 holds. We can perform the algorithm because A is in the data structure given by Proposition 3.2 (inflating the runtime by a factor of $\log(mn)$).

Correctness: Using Theorem 5.3 and Theorem 1,

$$\begin{aligned}\|T - D\|_F &\leq \|T - A_{\sigma,\eta}\|_F + \|A_{\sigma,\eta} - D\|_F \\ &\leq 3\|T - T_k\|_F + 3\varepsilon\|T\|_F + \varepsilon\|A\|_F.\end{aligned}$$

Applying a Chernoff bound to $\|A\|_F^2$ (a sum of independent random variables), we get that with high probability $1 - e^{-\|T\|_F^2 p/3}$, $\|A\|_F \leq \sqrt{2/p}\|T\|_F$. Since p is constant, and $\|T - T_k\|_F \leq \rho\|T\|_F$, we get that $\|T - D\|_F = O(\rho + \varepsilon)\|T\|_F$.

Then, we can apply Theorem 5.1 to get that, for S a (γ, ζ)-typical set of users of T, and $\mathcal{O}_i$ the output distribution for user i, we can find some $S' \subset S$ of size at least $(1 - \zeta - \psi)m$ such that, for all $i \in S'$,

$$\|\mathcal{O}_i - \mathcal{D}_{T_i}\|_{TV} \leq \|\mathcal{O}_i - \mathcal{D}_{D_i}\|_{TV} + \|\mathcal{D}_{D_i} - \mathcal{D}_{T_i}\|_{TV} \lesssim \varepsilon + (\varepsilon + \rho)\sqrt{\frac{1+\gamma}{\psi}} \lesssim \varepsilon + \rho,$$

which is the same bound that Kerenidis and Prakash achieve.

We can also get the same bound that they get when applying Theorem 5.2: although our total variation error is ε, we can still achieve the same desired $O(\varepsilon^2)$ failure probability as in the theorem. To see this, notice that in this model, Algorithm 3 samples from a vector α such that $\|\alpha - T_i\| \leq \varepsilon$. Instead of using Lemma 4.1, because T_i is binary, we can observe that the probability that an ℓ^2-norm sample from α is a bad recommendation is not ε, but $O(\varepsilon^2)$. From there, everything else follows similarly.

In summary, the classical algorithm has two forms of error that the quantum algorithm does not. However, the error in estimating the low-rank approximation folds into the error between T and $A_{\sigma,\eta}$, and the error in total variation distance folds into the error from sampling from an inexact reconstruction of T. Thus, we can achieve the same bounds.

Runtime: Our algorithm runs in time and query complexity

$$O(\mathrm{poly}(k, 1/\varepsilon, \|A_i\|/\|D_i\|)\,\mathrm{polylog}(mn, 1/\delta)),$$

which is the same runtime as Kerenidis and Prakash's algorithm up to polynomial slowdown.

To achieve the stated runtime, it suffices to show that $\|A_i\|/\|D_i\|$ is constant for a constant fraction of users in S. We sketch the proof here; the details are in Kerenidis and Prakash's proof [KP17b]. We know that $\|T - D\|_F \leq O(\rho + \varepsilon)\|T\|_F$. Through counting arguments we can show that, for a $(1 - \psi')$-fraction of typical users $S'' \subset S$,

$$\mathop{\mathrm{E}}_{i \sim_u S''}\left[\frac{\|A_i\|^2}{\|(A_{\sigma,\eta})_i\|^2}\right] \lesssim \frac{(1+\rho+\varepsilon)^2}{(1-\psi-\zeta)\left(\frac{1}{\sqrt{1+\gamma}} - \frac{\rho+\varepsilon}{\sqrt{\psi}}\right)^2}.$$

For ρ sufficiently small, this is a constant, and so by Markov's inequality a constant fraction S''' of S'' has $\|A_i\|/\|D_i\|$ constant. We choose $\bar{S}$ to be the intersection of S''' with S'. □

Acknowledgments

Thanks to Scott Aaronson for introducing me to this problem, advising me during the research process, and rooting for me every step of the way. His mentorship and help were integral to this work as well as to my growth as a CS researcher, and for this I am deeply grateful. Thanks also to Daniel Liang for providing frequent, useful discussions and for reading through

a draft of this document. Quite a few of the insights in this document were generated during discussions with him.

Thanks to Patrick Rall for the continuing help throughout the research process and the particularly incisive editing feedback. Thanks to everybody who attended my informal presentations and gave me helpful insight at Simons, including András Gilyén, Iordanis Kerenidis, Anupam Prakash, Mario Szegedy, and Ronald de Wolf. Thanks to Fred Zhang for pointing out a paper with relevant ideas for future work. Thanks to Sujit Rao and anybody else that I had enlightening conversations with over the course of the project. Thanks to Prabhat Nagarajan for the continuing support.

References

[Aar15] Scott Aaronson. "Read the fine print". In: *Nature Physics* 11.4 (2015), p. 291.

[AM07] Dimitris Achlioptas and Frank McSherry. "Fast computation of low-rank matrix approximations". In: *Journal of the ACM (JACM)* 54.2 (2007), p. 9.

[Awe+05] Baruch Awerbuch et al. "Improved Recommendation Systems". In: *Symposium on Discrete Algorithms*. 2005.

[Aza+01] Yossi Azar et al. "Spectral analysis of data". In: *Symposium on Theory of Computing*. ACM. 2001.

[BK07] Robert M Bell and Yehuda Koren. "Lessons from the Netflix prize challenge". In: *ACM SIGKDD Explorations Newsletter* 9.2 (2007), pp. 75–79.

[Ben+97] Charles H Bennett et al. "Strengths and weaknesses of quantum computing". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1510–1523.

[BL07] James Bennett and Stan Lanning. "The Netflix prize". In: *KDD Cup and Workshop*. 2007.

[BJ99] Nader H. Bshouty and Jeffrey C. Jackson. "Learning DNF over the Uniform Distribution Using a Quantum Example Oracle". In: *SIAM J. Comput.* 28.3 (1999), pp. 1136–1153.

[DV06] Amit Deshpande and Santosh Vempala. "Adaptive sampling and fast low-rank matrix approximation". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2006, pp. 292–303.

[DKM06] P. Drineas, R. Kannan, and M. Mahoney. "Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication". In: *SIAM Journal on Computing* 36.1 (2006), pp. 132–157.

[DMM08] P. Drineas, M. Mahoney, and S. Muthukrishnan. "Relative-Error CUR Matrix Decompositions". In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 844–881.

[DKR02] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. "Competitive recommendation systems". In: *Symposium on Theory of Computing*. ACM. 2002.

[FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala. "Fast Monte-Carlo algorithms for finding low-rank approximations". In: *Journal of the ACM (JACM)* 51.6 (2004), pp. 1025–1041.

[Gil+18] András Gilyén et al. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In: *arXiv* (2018). arXiv: 1806.01838 [quant-ph].

[HHL09] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". In: *Physical review letters* 103.15 (2009), p. 150502.

[HKS11] Elad Hazan, Tomer Koren, and Nati Srebro. "Beating SGD: Learning SVMs in Sublinear Time". In: *Neural Information Processing Systems*. 2011.

[KV17] Ravindran Kannan and Santosh Vempala. "Randomized algorithms in numerical linear algebra". In: *Acta Numerica* 26 (2017), pp. 95–135.

[KP17a] Iordanis Kerenidis and Anupam Prakash. "Quantum gradient descent for linear systems and least squares". In: *arXiv* (2017). arXiv: 1704.04992 [quant-ph].

[KP17b] Iordanis Kerenidis and Anupam Prakash. "Quantum recommendation systems". In: *Innovations in Theoretical Computer Science*. 2017.

[KS08] Jon Kleinberg and Mark Sandler. "Using mixture models for collaborative filtering". In: *Journal of Computer and System Sciences* 74.1 (2008), pp. 49–69.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems". In: *Computer* 42.8 (2009).

[Kum+01] Ravi Kumar et al. "Recommendation systems: a probabilistic analysis". In: *Journal of Computer and System Sciences* 63.1 (2001), pp. 42–61.

[LMR13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning". In: *arXiv* (2013). arXiv: 1307.0411 [quant-ph].

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis". In: *Nature Physics* 10.9 (2014), p. 631.

[Pre18] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79.

[Rec11] Benjamin Recht. "A simpler approach to matrix completion". In: *Journal of Machine Learning Research* 12 (2011), pp. 3413–3430.

[SWZ16] Zhao Song, David P. Woodruff, and Huan Zhang. "Sublinear Time Orthogonal Tensor Decomposition". In: *Neural Information Processing Systems*. 2016.

A Deferred Proofs

Proof of Lemma 4.5. We can describe ModFKV as FKV run on K with the filter threshold γ raised from $\Theta(\varepsilon/K)$ to $1/K$. The original work aims to output a low-rank approximation similar in quality to A_K, so it needs to know about singular values as low as ε/K. In our case, we don't need as strong of a bound, and can get away with ignoring these singular vectors. To prove our bounds, we just discuss where our proof differs from the original work's proof (Theorem 1 of [FKV04]). First, they show that

$$\Delta(W^T; u^{(t)}, t \in [K]) \geq \|A_K\|_F^2 - \frac{\varepsilon}{2}\|A\|_F^2.$$

The proof of this holds when replacing K with any $K' \leq K$. We choose to replace K with the number of singular vectors taken by ModFKV, k. Then we have that

$$\Delta(W^T; u^{(t)}, t \in T) = \Delta(W^T; u^{(t)}, t \in [k]) \geq \|A_k\|_F^2 - \frac{\varepsilon}{2}\|A\|_F^2.$$

We can complete the proof now, using that $[k] = T$ because our filter accepts the top k singular vectors (though not the top K). Namely, we avoid the loss of $\gamma\|W\|_F^2$ that they incur in this way. This gives the bound ($\diamondsuit$).

Further, because we raise γ, we can correspondingly lower our number of samples. Their analysis requires q (which they denote p) to be $\Omega(\max\{\frac{k^4}{\varepsilon^2}, \frac{k^2}{\varepsilon\gamma^2}, \frac{1}{\varepsilon^2\gamma^2}\})$ (for Lemma 3, Claim 1, and Claim 2, respectively). So, for us we can pick $q = \Theta(k^4/\varepsilon^2)$.

As for bounding k, ModFKV can compute the first k singular values to within a cumulative additive error of $\bar{\varepsilon}\|A\|_F$. This is stated below Theorem 2 in [FKV04]. Thus, ModFKV could only conceivably take a singular vector v such that $\|Av\| \geq \sigma - \bar{\varepsilon}\|A\|_F = \sigma(1 - \bar{\varepsilon}\|A\|_F/\sigma)$, and analogously for the upper bound. □

Proof of Proposition 4.6. We follow the proof of Claim 2 of [FKV04]. For $i \neq j$, we have as follows:

$$\left|\hat{v}_i^T \hat{v}_j\right| = \frac{|u_i^T SS^T u_j|}{\|W^T u_i\|\|W^T u_j\|} \leq \frac{|u_i^T SS^T u_j|}{\sigma^2} \leq \frac{\|S\|_F^2}{\sigma^2\sqrt{q}} = \frac{\bar{\varepsilon}}{K}$$
$$\left|1 - \hat{v}_i^T \hat{v}_i\right| = \frac{|u_i^T WW^T u_i| - |u_i^T SS^T u_j|}{\|W^T u_i\|\|W^T u_j\|} \leq \frac{\|S\|_F^2}{\sigma^2\sqrt{q}} = \frac{\bar{\varepsilon}}{K}$$

Here, we use that $\|WW^T - SS^T\| \leq \|S\|_F^2/\sqrt{q}$ (Lemma 2 [FKV04]) and $\{Wu_i\}$ are orthogonal.

This means that $\hat{V}^T\hat{V}$ is $O(\bar{\varepsilon}/K)$-close entry-wise to the identity. Looking at $\hat{V}$'s singular value decomposition into $A\Sigma B^T$ (treating Σ as square), the entrywise bound implies that $\|\Sigma^2 - I\|_F \lesssim \bar{\varepsilon}$, which in turn implies that $\|\Sigma - I\|_F \lesssim \bar{\varepsilon}$. $\Lambda := AB^T$ is close to $\hat{V}$, orthonormal, and in the same subspace as desired. □

Proof of Theorem 4.7. We will prove a slightly stronger statement: it suffices to choose $A_{\sigma,\eta}$ such that $P_{\sigma,\eta}$ is an orthogonal projector (which we will call $\Pi_{\sigma,\eta}$). In fact, we could have used this restricted version as our definition of $A_{\sigma,\eta}$ if we wanted to. We use the notation $\Pi_E := \Pi_{\sigma,\eta} - \Pi_{\sigma(1+\eta)}$ to refer to the part of $\Pi_{\sigma,\eta}$ that can vary; it can be any orthogonal projector on the span of the singular vectors with values in $[\sigma(1-\eta), \sigma(1+\eta))$.

For simplicity we denote σ_k by σ and $\min m, n$ by N.

$$\begin{aligned}
\|A\Pi - A_{\sigma,\eta}\|_F^2 &= \|U\Sigma V^T(\Pi - \Pi_{\sigma,\eta})\|_F^2 \\
&= \|\Sigma V^T(\Pi - \Pi_{\sigma,\eta})\|_F^2 \\
&= \sum_{i=1}^{N} \sigma_i^2 \|v_i^T\Pi - v_i^T\Pi_{\sigma,\eta}\|^2
\end{aligned}$$

That is, $A\Pi$ and $A_{\sigma,\eta}$ are close when their corresponding projectors behave in the same way. Let $a_i = v_i^T \Pi$, and $b_i = v_i^T \Pi_{\sigma,\eta}$. Note that

$$b_i = \begin{cases} v_i^T & \sigma_i \geq (1+\eta)\sigma \\ v_i^T \Pi_E & (1+\eta)\sigma > \sigma_i \geq (1-\eta)\sigma \\ 0 & (1-\eta)\sigma > \sigma_i \end{cases}.$$

Using the first and third case, and the fact that orthogonal projectors Π satisfy $\|v - \Pi v\|^2 = \|v\|^2 - \|\Pi v\|^2$, the formula becomes

$$\|A\Pi - A_{\sigma,\eta}\|_F^2 = \sum_1^{\ell(\sigma(1+\eta))} \sigma_i^2(1 - \|a_i\|^2) + \sum_{\ell(\sigma(1+\eta))+1}^{\ell(\sigma(1-\eta))} \sigma_i^2 \|a_i - b_i\|^2 + \sum_{\ell(\sigma(1-\eta))+1}^{N} \sigma_i^2(\|a_i\|^2).$$

♠ ♠ ♠

Now, we consider the assumption equation. We reformulate the assumption into the following system of equations:

$$\sum_{i=1}^{k} \sigma_i^2 \leq \sum_{i=1}^{N} \sigma_i^2 \|a_i\|^2 + \varepsilon\sigma^2 \qquad \sigma_i^2 \text{ are nonincreasing}$$
$$\|a_i\|^2 \in [0,1] \qquad \sum \|a_i\|^2 = k$$

The first line comes from the equation. The second line follows from Π being an orthogonal projector on a k-dimensional subspace.

It turns out that this system of equations is enough to show that the $\|a_i\|^2$ behave the way we want them to. We defer the details to Lemma A.2; the results are as follows.

$$\sum_1^{\ell(\sigma_k(1+\eta))} \sigma_i^2(1 - \|a_i\|^2) \leq \varepsilon\Big(1 + \frac{1}{\eta}\Big)\sigma_k^2 \qquad \sum_{\ell(\sigma_k(1-\eta))+1}^{N} \sigma_i^2 \|a_i\|^2 \leq \varepsilon\Big(\frac{1}{\eta} - 1\Big)\sigma_k^2$$
$$\sum_1^{\ell(\sigma_k 1+\eta))} (1 - \|a_i\|^2) \leq \frac{\varepsilon}{\eta} \qquad \sum_{\ell(\sigma_k(1-\eta))+1}^{N} \|a_i\|^2 \leq \frac{\varepsilon}{\eta}$$

Now, applying the top inequalities:

$$\|A\Pi - A_{\sigma,\eta}\|_F^2 \leq \frac{2\varepsilon\sigma^2}{\eta} + \sum_{\ell(\sigma(1+\eta))+1}^{\ell(\sigma(1-\eta))} \sigma_i^2 \|a_i - b_i\|^2.$$

We just need to bound the last term in the above inequality now. Notice the following:

$$\sum_{\ell(\sigma(1+\eta))+1}^{\ell(\sigma(1-\eta))} \sigma_i^2 \|a_i - b_i\|^2 \leq \sigma^2(1+\eta)^2 \|U^T(\Pi - \Pi_E)\|_F^2,$$

where U is the set of vectors $v_{\ell(\sigma(1+\eta))+1}$ through $v_{\ell(\sigma(1-\eta))}$.

♠ ♠ ♠

Notice that Π_E is the error component of the projection, and this error can be any projection onto a subspace spanned by U. Thus, to bound the above we just need to pick an orthogonal projector Π_E making the norm as small as possible. If $UU^T\Pi$ were an orthogonal projection, this would be easy:

$$\|U^T(\Pi - UU^T\Pi)\|_F^2 = 0.$$

However, this is likely not the case. $UU^T\Pi$ is *close* to an orthogonal projector, though, through the following reasoning:

For ease of notation let P_1 be the orthogonal projector onto the first $\ell(\sigma(1+\eta))$ singular vectors, $P_2 = UU^T$, and P_3 be the orthogonal projector onto the the rest of the singular vectors. We are concerned with $P_2\Pi$.

Notice that $P_1 + P_2 + P_3 = I$. Further, $\|(I-\Pi)P_1\|_F^2 \leq \varepsilon/\eta$ and $\|\Pi P_3\|_F^2 \leq \varepsilon/\eta$ from Lemma A.2. Then

$$\begin{aligned} P_2\Pi = (I - P_1 - P_3)\Pi &= \Pi - P_1 + P_1(I-\Pi) - P_3\Pi \\ \|P_2\Pi - (\Pi - P_1)\|_F &= \|P_1(I-\Pi) - P_3\Pi\|_F \leq 2\sqrt{\varepsilon/\eta} \end{aligned}$$

So now it is sufficient to show that $\Pi - P_1$ is close to a projector matrix. This follows from Lemma A.1, since it satisfies the premise:

$$\begin{aligned} (\Pi - P_1)^2 - (\Pi - P_1) &= \Pi - \Pi P_1 - P_1\Pi + P_1 - \Pi + P_1 \\ &= (I-\Pi)P_1 + P_1(I-\Pi) \\ \|(\Pi - P_1)^2 - (\Pi - P_1)\|_F &\leq 2\sqrt{\varepsilon/\eta} \end{aligned}$$

Thus, $UU^T\Pi$ is $(2\sqrt{\varepsilon/\eta} + (2\sqrt{\varepsilon/\eta} + 16\varepsilon/\eta))$-close to an orthogonal projector in Frobenius norm.

♠ ♠ ♠

We can choose Π_E to be M, and plug this into the original equation. We use the assumptions that $\varepsilon/\eta < 1$ and $\eta < 1$ to bound.

$$\begin{aligned} \sum_{\ell(\sigma(1+\eta))+1}^{\ell(\sigma(1-\eta))} \sigma_i^2\|a_i - b_i\|^2 &\leq \sigma^2(1+\eta)^2\|U^T(\Pi - M)\|_F^2 \\ &\leq \sigma^2(1+\eta)^2\|U^T(\Pi - (UU^T\Pi + E))\|_F^2 \\ &\leq \sigma^2(1+\eta)^2\|U^TE\|_F^2 \\ &\lesssim \sigma^2(1+\eta)^2\varepsilon/\eta \\ \|A\Pi - A_{\sigma,\eta}\|_F^2 &\lesssim \frac{2\varepsilon\sigma^2}{\eta} + \sigma^2(1+\eta)^2\frac{\varepsilon}{\eta} \lesssim \varepsilon\sigma^2/\eta \end{aligned}$$

This concludes the proof. (The constant factor is 1602.) □

Lemma A.1. If a Hermitian A satisfies $\|A^2 - A\|_F \le \varepsilon$, then $\|A - P\|_F \le \varepsilon + 4\varepsilon^2$ for some orthogonal projector P.

Proof. Use the fact that Hermitian matrices are normal, so $A = U\Gamma U^T$ for unitary U and diagonal matrix Γ, and

$$A^2 - A = U(\Gamma^2 - \Gamma)U^T \implies \|\Gamma^2 - \Gamma\|_F \le \varepsilon.$$

From here, consider the entries γ_i of Γ, satisfying $\gamma_i^2 - \gamma_i = c_i$ and $\sum c_i^2 = \varepsilon^2$. Thus, $\gamma_i = (1 \pm \sqrt{1 + 4c_i})/2$ which is at most $c_i + 4c_i^2$ off from 0.5 ± 0.5 (aka $\{0, 1\}$), using that

$$1 - x/2 - x^2/2 \le \sqrt{1 - x} \le 1 - x/2.$$

Finally, this means that Γ is off from having only 0's and 1's on the diagonal by $\sqrt{\sum (c_i + 4c_i^2)^2} \le \varepsilon + 4\varepsilon^2$ in Frobenius norm.

If Γ had only 0's and 1's on the diagonal, the resulting $U\Gamma U^T$ would be an orthogonal projector. □

Lemma A.2. The system of equations:

$$\sum_{i=1}^{k} \sigma_i^2 \le \sum_{i=1}^{N} \sigma_i^2 \|a_i\|^2 + \varepsilon\sigma_k^2 \qquad \sigma_i^2 \text{ are nonincreasing}$$
$$\|a_i\|^2 \in [0, 1] \qquad \sum \|a_i\|^2 = k$$

imply the following, for $0 < \eta \le 1$:

$$\sum_{1}^{\ell(\sigma_k(1+\eta))} \sigma_i^2(1 - \|a_i\|^2) \le \varepsilon\Big(1 + \frac{1}{\eta}\Big)\sigma_k^2 \qquad \sum_{\ell(\sigma_k(1-\eta))+1}^{N} \sigma_i^2\|a_i\|^2 \le \varepsilon\Big(\frac{1}{\eta} - 1\Big)\sigma_k^2$$
$$\sum_{1}^{\ell(\sigma_k 1+\eta))} (1 - \|a_i\|^2) \le \frac{\varepsilon}{\eta} \qquad \sum_{\ell(\sigma_k(1-\eta))+1}^{N} \|a_i\|^2 \le \frac{\varepsilon}{\eta}$$

Proof. We are just proving straightforward bounds on a linear system. We will continue to denote σ_k by σ. Thus, $k = \ell(\sigma)$.

The slack in the inequality is always maximized when the weight of the $\|a_i\|^2$ is concentrated on the large-value (small-index) entries. For example, the choice of $\|a_i\|^2$ maximizing slack in the given system of equations is the vector $\{\|a_i\|\}_{i\in[N]} = \mathbf{1}_{\le k}$. Here, $\mathbf{1}_{\le x}$ denotes the vector where

$$(\mathbf{1}_{\le x})_i := \begin{cases} 1 & i \le x \\ 0 & \text{otherwise} \end{cases}.$$

For brevity, we only give the details for the first bound; the others follow similarly. Consider adding the constraint $C = \sum_1^{\ell(\sigma(1+\eta))} \sigma_i^2(1 - \|a_i\|^2)$ to the system of equations. We want to determine for which values of C the modified system is still feasible; we can do this by trying the values that maximize slack.

This occurs when weight is on the smallest possible indices: when $\|a_{\ell(\sigma(1+\eta))}\|^2 = 1 - C/\sigma^2_{\ell(\sigma(1+\eta))}$, $\|a_{\ell(\sigma)+1}\|^2 = C/\sigma^2_{\ell(\sigma(1+\eta))}$, and all other $\|a_i\|^2$ are $\mathbf{1}_{\geq k}$. Notice that $\|a_{\ell(\sigma(1+\eta))}\|^2$ could be negative and $\|a_{\ell(\sigma)+1}\|$ could be larger than one, breaking constraints. However, if there is no feasible solution even when relaxing those two constraints, there is certainly no solution to the non-relaxed system. Thus, we check feasibility (by construction the second equation is satisfied):

$$\begin{aligned}
\sum_{i=1}^{k} \sigma_i^2 &\leq \sum_{i=1}^{k} \sigma_i^2 - C + C\frac{\sigma^2_{\ell(\sigma)+1}}{\sigma_{\ell(\sigma(1+\eta))}} + \varepsilon\sigma^2 \\
C\Big(1 - \frac{\sigma^2_{\ell(\sigma)+1}}{\sigma_{\ell(\sigma(1+\eta))}}\Big) &\leq \varepsilon\sigma^2 \\
C\Big(1 - \frac{1}{(1+\eta)^2}\Big) &\leq \varepsilon\sigma^2
\end{aligned}$$

This gives the bound on C. Repeating for all four cases, we get the following bounds:

$$\sum_{1}^{\ell(\sigma(1+\eta))} \sigma_i^2(1 - \|a_i\|^2) \leq \frac{\varepsilon(1+\eta)^2\sigma^2}{2\eta+\eta^2} \qquad \sum_{\ell(\sigma(1-\eta))+1}^{N} \sigma_i^2\|a_i\|^2 \leq \frac{\varepsilon(1-\eta)^2\sigma^2}{2\eta-\eta^2}$$

$$\sum_{1}^{\ell(\sigma(1+\eta))} (1 - \|a_i\|^2) \leq \frac{\varepsilon}{2\eta+\eta^2} \qquad \sum_{\ell(\sigma(1-\eta))+1}^{N} \|a_i\|^2 \leq \frac{\varepsilon}{2\eta-\eta^2}$$

We get the bounds in the statement by simplifying the above (using that $\eta \leq 1$). □

Proof of Theorem 5.1. The following shows the first, average-case bound (note the use of Lemma 4.1 and Cauchy-Schwarz).

$$\begin{aligned}
\mathrm{E}_{i\sim_u S}\left[\|\mathcal{D}_{T_i} - \mathcal{D}_{\tilde{T}_i}\|_{TV}\right] &= \frac{1}{|S|}\sum_{i\in S} \|\mathcal{D}_{T_i} - \mathcal{D}_{\tilde{T}_i}\|_{TV} \\
&\leq \frac{1}{(1-\zeta)m}\sum_{i\in S}\frac{2\|T_i - \tilde{T}_i\|}{\|T_i\|} \\
&\leq \frac{2(1+\gamma)}{(1-\zeta)\sqrt{m}\|T\|_F}\sum_{i\in S}\|T_i - \tilde{T}_i\| \\
&\leq 2\frac{1+\gamma}{1-\zeta}\Big(\frac{\sum_{i\in[m]}\|T_i - \tilde{T}_i\|}{\sqrt{m}\|T\|_F}\Big) \\
&\leq 2\frac{1+\gamma}{1-\zeta}\Big(\frac{\sqrt{m}\|T - \tilde{T}\|_F}{\sqrt{m}\|T\|_F}\Big) \\
&\leq \frac{2\varepsilon(1+\gamma)}{(1-\zeta)}
\end{aligned}$$

Using that $\|T - \tilde{T}\|_F \leq \varepsilon\|T\|_F$ in combination with a pigeonhole-like argument, we know

that at least a $(1-\psi)$-fraction of users $i \in [m]$ satisfy

$$\|T_i - \tilde{T}_i\|^2 \leq \frac{\varepsilon^2 \|A\|_F^2}{\psi m}.$$

Thus, there is a $S' \subset S$ of size at least $(1-\psi-\zeta)m$ satisfying the above. For such an $i \in S'$, we can argue from Lemma 4.1 and the definition of a (γ, ζ)-typical user that

$$\|\mathcal{D}_{T_i} - \mathcal{D}_{\tilde{T}_i}\|_{TV} \leq \frac{2\|T_i - \tilde{T}_i\|}{\|T_i\|} \leq \frac{2\varepsilon \|T\|_F (1+\gamma)\sqrt{m}}{\sqrt{\psi m}\|T\|_F} = \frac{2\varepsilon(1+\gamma)}{\sqrt{\psi}}.$$

□

B Variant for an Alternative Model

In this section, we describe a variant of our recommendation systems algorithm for the competitive recommendations model, seen in Drineas, Kerenidis, and Raghavan's 2002 paper giving two algorithms for competitive recommendations [DKR02]. The idea is to output good recommendations with as little knowledge about the preference matrix T as possible. Our algorithm is similar to Drineas et al's second algorithm, which has weak assumptions on the form of T, but strong assumptions on how we can gain knowledge about it.

We use a similar model, as follows:

- We begin with no knowledge of our preference matrix T apart from the promise that $\|T - T_k\|_F \leq \rho \|T\|_F$;
- We can request the value of an entry T_{ij} for some cost;
- For some constant $0 < c \leq 1$, we can sample from and compute probabilities from a distribution P over $[m]$ satisfying

  $$P(i) \geq c \frac{\|T_i\|^2}{\|T\|_F^2}.$$

  Further, we can sample from and compute probabilities from distributions Q_i over $[n]$, for $i \in [m]$, satisfying

  $$Q_i(j) \geq c \frac{T_{ij}^2}{\|T_i\|^2}.$$

We discuss the first assumption in Section 5.1. The second assumption is very strong, but we will only need to use it sparingly, for some small set of users and products. In practice, this assumption could be satisfied through paid user surveys.

The last assumption states that the way that we learn about users naturally, via normal user-site interaction, follows the described distributions. For example, consider when T is binary (as in Section 5.1). The assumption about P states that we can sample for users proportional to the number of products they like (with possible error via c). Even though

we don't know the exact number of products a user likes, it is certainly correlated with the amount of purchases/interactions the user has with the site. With this data we can form P. The assumption about Q_i's states that, for a user, we can sample uniformly from the products that user likes. We can certainly assume the ability to sample from the products that a user likes, since such positive interactions are common, intended, and implicit in the user's use of the website. It is not clear whether uniformity is a reasonable assumption, but this can be made more reasonable by making T non-binary and more descriptive of the utility of products to users.

Under these assumptions, our goal is, given a user i, to recommend products to that user *that were not previously known to be good* and are likely to be good recommendations.

To do this, we run Algorithm 3 with T, k, ε as input, the main change being that we use Frieze, Kannan, and Vempala's algorithm as written in their paper instead of ModFKV. As samples and requests are necessary, we can provide them using the assumptions above.

For the FKV portion of the algorithm, this leads to $O(p^2)$ requests to q users about q products, where $q = O(\max\{\frac{k^4}{c^3\varepsilon^6}, \frac{k^2}{c^3\varepsilon^8}\})$. This gives the description of a D such that

$$\|T - D\|_F \leq \sqrt{\|T - T_k\|_F^2 + \varepsilon^2\|T\|_F^2} \leq (\rho + \varepsilon)\|T\|_F.$$

Thus, immediately we can use theorems from Section 5.1 to show that samples from D will give good recommendations.

From here, the next part of Algorithm 3 can output the desired approximate sample from D_i. A similar analysis will show that this approximate sample is likely to be a good recommendation, all while requesting and sampling a number of entries independent of m and n. Such requests and samples will only be needed for the q users chosen by FKV for its subsample, along with the input user. Further, for more recommendations, this process can be iterated with unused information about the q users chosen by FKV. Alternatively, if we can ask the q users for all of their recommendations, we only need $O(\frac{k^2}{\varepsilon^2}\log\frac{k}{\delta})$ samples from the input user to provide that user with an unlimited number of recommendations (we can store and update the estimate of A_iS^T to use when sampling).

This gives good recommendations, only requiring knowledge of $O(\text{poly}(k, 1/\varepsilon, \log 1/\delta))$ entries of T, and with time complexity polynomial in the number of known entries.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il