On differentially private low rank approximation

Michael Kapralov* Kunal Talwar†

October 3, 2012

Abstract

Low rank approximation is a fundamental computational primitive widely used in data analysis. In many applications the dataset that the algorithm operates on may contain sensitive information about contributing individuals (e.g. user/movie ratings in the Netflix challenge), motivating the need to design low rank approximation algorithms that preserve privacy of individual entries of the input matrix.

In this paper, we give a polynomial time algorithm that, given a privacy parameter $\epsilon > 0$, for a symmetric matrix A, outputs an ϵ-differentially approximation to the principal eigenvector of A, and then show how this algorithm can be used to obtain a differentially private rank-k approximation. We also provide lower bounds showing that our utility/privacy tradeoff is close to best possible.

While there has been significant progress on this problem recently for a weaker notion of privacy, namely (ϵ, δ)-differential privacy [HR12, BBDS12], our result is the first to achieve $(\epsilon, 0)$-differential privacy guarantees with a near-optimal utility/privacy tradeoff in polynomial time.

1 Introduction

Low-rank approximation is widely used in statistical analysis and machine learning, where the learner is inferring useful aggregate statistics from a database of records. In a commonly used scenario users are represented by vectors in d dimensional space, corresponding, for example, to their preferences for certain items. A canonical example is the Netflix challenge, where users are present in the database via vectors of their movie ratings. The task of the learner is to infer the ratings of the user for movies that she has not seen. Several successful solutions to the challenge first computed a "covariance" matrix A whose entry a_{ij} corresponds to the correlation between the rankings of movie i and movie j. Intuitively, a user who likes a movie i is also likely to like a movie j if a_{ij} is large. Since the actual reported rankings are rather sparse, the measured matrix A is likely to be noisy, and most geometric learning algorithms first find a low dimensional subspace that best fits the data. This can be done by finding the best rank k approximation to this covariance matrix A, which can be done by taking the top k eigenvectors of A.

In a setting such as the Netflix challenge, privacy may be an important concern. In fact, the deanonymized data released by Netflix led to privacy breaches [15] and led to the cancellation of a planned Netflix 2 challenge. Similar privacy risks have been demonstrated in settings like Hunch and Amazon where no data is directly released (as in the Netflix challenge), but items similar to a given item are shown to the users [5]. This was one of the motivations for the work of McSherry and Mironov [12], who designed a differentially private recommendation system, where the recommendations given to a particular user did not depend significantly on the preferences of any other user (it is of course unavoidable in any useful system that the items recommended to me depend on my preferences). Indeed one of the important steps in that work is a differentially private computation of a low rank approximation of a covariance like matrix. While their work used input perturbation to achieve privacy, in this work we suggest alternate ways to compute a differentially private approximation to this objective.

The simplest version of low rank approximation is the problem of computing the top eigenvector of a matrix. How well can one do that? A little reflection shows that in general, it is impossible to output a vector that is close to the top eigenvector while preserving privacy: Indeed a small perturbation to the identity matrix can cause a large change in the top eigenvector, and if the private mechanism has to hide small perturbations, it

*Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA. Email: kapralov@stanford.edu. Supported by NSF grant NSF grants 0904325 and 0947670. The author also acknowledges financial support from grant #FA9550-12-1-0411 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA). Part of the work was done when the author was an intern at Microsoft Research Silicon Valley.

†Microsoft Research Silicon Valley, Mountain View, CA. Email: kunal@microsoft.com

cannot do much better than outputting a random vector in this case. On the other hand, the reason one wants the eigenvector v in this setting is that it maximized the "energy" in it's direction, i.e. maximized $v^T Cv$ amongst all unit vectors. Can one privately find a vector v such that $v^T Cv$ is close to λ_1? This is the first question we address in this work. For the best rank k approximation, the relevant question then is to find a rank k matrix B_k such that $\|A - B_k\|$ is close to the best possible.

The privacy guarantee in differential privacy is usually with respect to an allowable set of perturbations: those that can be caused by a single user changing her input. This set depends on the application in question. In our setting, if we denote the vector of ratings of a user by x_u, the covariance matrix would be sum $\sum_{u \in D} x_u x_u^T$. Assuming by scaling that $\|x_u\|$ is bounded by 1, a single user can change A by at most 1 in the spectral norm[1]. Thus we will be seeking mechanisms that are differentially private with respect to unit spectral norm perturbations to the matrix A.

Note that one expects the eigenvalues of A to scale with the number of users in the database. While one cannot give useful approximation to (say) the top eigenvector when the corresponding eigenvalue is small, we would like to give good answers when the number of users in the database (and hence the top eigenvector) is large enough.

For rank-1 approximation this motivates the following formulation. We would like to have an ϵ-differentially private algorithm that for a symmetric positive semidefinite $d \times d$ matrix A with largest eigenvalue $\lambda_1 \geq 0$ outputs a vector $v \in \mathbb{S}^{d-1}$ such that

$$\mathbf{E}[v^T Av] \geq (1 - \delta)\lambda_1, \tag{1.1}$$

for some $\delta > 0$ as long as $\lambda_1 \geq \lambda^*(d, \epsilon, \delta)$. What is the smallest $\lambda^*(d, \epsilon, \delta)$ that admits such an algorithm? In this paper, we give an algorithms of this form and provide nearly matching bounds on $\lambda^*(d, \epsilon, \delta)$. We also provide similar bounds for the rank-k approximation problem.

How would one approach this question using known tools in differential privacy? Adding noise to the true eigenvector proportional to the global sensitivity [8] is easily seen to require an unacceptable amount of noise. Adding noise proportional to the smoothed sensitivity [16] is a candidate approach. However it is not clear how to compute a smooth upper bound on the sensitivity. Moreover, analyzing the quality of this mechanism seems non-trivial. Input perturbation is another natural candidate, that was considered in [4]. The approach we use in this work is to use the exponential mechanism [13], which gives a better utility guarantee. It is not a priori clear if one can sample efficiently from this distribution, and showing how to do this is one of the technical contributions of this work.

Our results. We give algorithms for releasing an ϵ-differentially private low-rank approximation to a symmetric matrix. Our algorithm for general k relies on an algorithm for outputting a differentially private vector most of whose mass is concentrated in the few most significant eigenvalues. We then use this primitive to iteratively subtract rank-1 approximations from the input matrix, obtaining an approximate differentially private SVD.

Rank-1 approximation. An algorithm for outputting a differentially private rank-1 approximation to a symmetric matrix A is essentially a distribution on vectors on the unit sphere $\mathbb{S}^d$ together with an efficient algorithm for sampling from this distribution. Our distribution is given by the well-known exponential mechanism [13], which outputs $x \in \mathbb{S}^{d-1}$ with probability proportional to $e^{x^T Ax}, x \in \mathbb{S}^{d-1}$. We show

THEOREM 1.1. *The exponential mechanism, given a symmetric positive definite $d \times d$ matrix A with largest eigenvalue λ, outputs a vector $v \in \mathbb{S}^{d-1}$ such that*

$$\mathbf{E}[v^T Av] \geq (1 - \delta)\lambda_1$$

as long as $\lambda_1 \geq (Cd \log(1/\delta))/(\epsilon\delta)$ for an absolute constant $C > 0$. Furthermore, the exponential mechanism can be implemented to run in time poly$(d, \lambda_1 - \lambda_d)$.

As suggested earlier, the exponential mechanism on the sphere is non-trivial to implement. The main hurdle is that the level sets of the Rayleigh quotient $x^T Ax$ on the sphere $\mathbb{S}^{d-1}$ are not convex, making standard techniques for sampling from convex bodies inapplicable. A brute force solution would take time exponential in the dimension d, which we would like to avoid. Nevertheless, we give a polynomial time algorithm for sampling from this distribution. The sampling algorithm and analysis is the main technical contribution of this part of the paper.

We note that the assumption that A be positive semidefinite is not constraining for us. In fact, the distributions that we use for all our results are invariant under adding a multiple of the identity matrix to A. Thus, the assumption that A is positive semidefinite is without loss of generality.

[1] In [12], the "covariance" matrix is calculated in a different way, giving different weights to different users. Nevertheless, it is done in a way so that the norm of the effect of any single user on the matrix is bounded by an absolute constant

Rank-k approximation. Our algorithms for rank-1 approximation can be applied repeatedly to obtain a differentially private rank-k approximation to symmetric matrices. We state our results for positive semidefinite matrices only for clarity of presentation. However, our techniques easily extend to the case of general symmetric matrices. Our main result for rank-k approximation is

THEOREM 1.2. *Let $A \in \mathbb{R}^{d\times d}$ be a symmetric psd matrix and let $\lambda_1 \geq \ldots \geq \lambda_d$ denote the eigenvalues of A. There exists an ϵ-differentially private polynomial time algorithm for computing a matrix A_k of rank at most k so that $||A - A_k||_2 \leq \lambda_{k+1} + \delta\lambda_1$ as long as $\lambda_1 \geq C_1 dk^3/(\epsilon\delta^6)$ for a constant $C_1 > 0$.*

Note that this improves upon input perturbation whenever $\lambda_1 < d^2/(\epsilon\delta)$ (a more detailed discussion of the guarantees given by input perturbation is given below).

The main idea of the algorithm is simple. Starting with $A_0 = A$, at step $i = 0, \ldots, k-1$ we repeatedly sample a differentially private vector v that nearly maximizes $v^T A v$, and subtract an appropriately scaled rank-1 matrix from A_i and let $A_{i+1} = A_i - (v^T A_i v)vv^T$. The main technical component of the proof is the analysis of the change in the spectral norm of A_i during this 'approximate SVD' process.

Lower bounds. Finally, we provide lower bounds for the problem in section 5. Almost matching lower bounds for the rank-1 case follow by a packing argument on $\mathbb{S}^d$. The lower bounds for rank-k approximation follow by a packing argument on the Grassmannian manifold, and show that the additive term in the approximation necessarily has to be at least on the order of $\Omega(\frac{1}{\epsilon}dk\log(1/\delta))$, showing that our guarantees are close to optimal for constant k.

Related work. The problem of constructing differentially private low rank approximation primitives has received substantial attention in the literature. To the best of our knowledge, the first solution, which uses input perturbation, was given in in [4]. The work of McSherry and Mironov[12] applied input perturbation techniques to the Netflix dataset. Since the work of [12] the problem of improving upon the accuracy/privacy tradeoff achieved by input perturbation has been an important open problem. Besides being interesting in its own right, the question of obtaining better tradeoffs for differentially private low-rank approximation has been shown to be important for the closely related problem of answering *cut queries* [9].

Comparison to [10]. The recent work of [10] gives the first algorithm to providing guarantees superior to input perturbation for low-rank approximation under the additional *incoherence assumption*. Their algorithm, given a $n \times m$ matrix A, outputs an accurate rank-k approximation while satisfying (ϵ, δ)-differential privacy with respect to unit l_2-norm changes to a single rows of A. We give a stronger $(\epsilon, 0)$-differential privacy guarantee under a more general unit spectral norm change. On the other hand, in this paper we make the assumption that the input matrices are symmetric (which is justified for applications such as the Netflix dataset, where A is the covariance matrix). Thus, our results are incomparable to those of [10].

Comparison to [3]. The recent work of [3] gives (ϵ, δ)-differentially private algorithms for two related problems. The first is an algorithm for releasing the Laplacian matrix of a graph that satisifies (ϵ, δ)-differential privacy with respect to *edge changes* (i.e changing one 0-1 entry of the matrix). This setting is different from ours in two respects. First, we consider a significantly broader class of inputs, i.e. symmetric matrices, and give algorithms that are private with respect to *spectral norm changes*. Second, our notion of privacy is the more stringent ϵ-differential privacy. The authors of [3] also give an algorithm for answering *directional variance queries* for a covariance matrix, again under the (ϵ, δ)-differential privacy requirements. As the authors of [3] point out, this estimation procedure is somewhat weaker than low-rank approximation.

Thus, our results are the first to improve upon input perturbation for low rank approximation under the more stringent notion of $(\epsilon, 0)$-differential privacy. Furthermore, we use a stronger notion of perturbation, namely unit spectral norm perturbations.

Independent of our work, Chaudhuri et al. [6] propose and evaluate the exponential mechanism for PCA. They use a Markov Chain Monte Carlo method to sample approximately from the distribution, and show that it empirically outperforms input perturbation on real datasets. However, lacking bounds on convergence time, the authors use heuristic tests to check convergence of the chain. They leave open the question of how this impacts the privacy guarantee.

Techniques. A possible approach to try to sample from the exponential mechanism's distribution is the following. The set of points in $\mathbb{R}^d$ that have Rayleigh coefficient at most r is an ellipse. If we considered the distribution defined by $\exp(x^T Ax)$ in the unit ball, it can be checked that this gives good utility as well as privacy. Since the set of point $\{x \in \mathbb{R}^d : \|x\|_2 \leq 1, \exp(x^T Ax) \leq r\}$ is the intersection of an two ellipses, it is convex and standard techniques may suffice to sample from it and estimate its volume. However,

to be able to sample from the distribution, we need to estimate well the volume of the complement sets $L(r) = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1, \exp(x^T Ax) \geq r\}$. Since the volume of the unit ball $B_d(1)$ is easy to compute, we can estimate the volume of $L(r)$ as $\mathrm{Vol}(B_d(1)) - \mathrm{Vol}(B_d(1) \setminus L(r))$, the latter being convex. However, given a good approximation to the latter volume, we get a good approximation to the volume of $L(r)$ only when this volume is large enough. One the other hand, if we have one eigenvalue much larger than the others, a constant fraction of the probability mass of the exponential mechanism comes from a set of exponentially small Euclidean volume. Thus a naive approach would require the volume estimation to be accurate up to an exponentially small error, which is infeasible. We do not know if such an approach can lead to an efficient algorithm.

How do we get around this hurdle? We use a more direct approach to estimating the volume of the set $L(r)$. We observe that $L(r)$ is an integral of similar volumes in one lower dimension. We show that these volumes grow smoothly except possibly in the neighborhood of eigenvalues. In fact we can control the growth rate and argue that the integral is well approximated by weighted sum over a polynomial sized (but very non-uniform) net. The volumes in this lower dimensional space can be similarly recursively computed until we are left with a one dimensional integral that can be computed. This recursive approach is turned into a dynamic program to get a polynomial time algorithm.

Open problems. In this paper we present essentially optimal algorithms for the top eigenvector, and near optimal ones for the top k ones. Several open questions suggest themselves. The implementation of the exponential mechanism, while polynomial time, is fairly complicated and it is natural to look for simpler and more efficient algorithms for the problem. Analyzing smooth sensitivity based, or sample-and-aggregate based mechanisms may be one approach. For rank k approximation, our analysis is lossy and it would be natural to get better dependency on k.

Organization. Preliminaries and basic definitions are presented in section 2. We present our algorithm for differentially private rank-1 approximation in section 3. We then show how to use these rank-1 approximation primitives to obtain differentially private rank-k approximation in section 4. In section 5 we prove lower bounds on the utility/privacy tradeoff for rank-k approximation. Finally, in section 6 we give a polynomial time implementation of the exponential mechanism from section 3.

2 Preliminaries

In this paper we consider symmetric positive definite matrices $A \in \mathbb{R}^{d \times d}$. We will use the standard notation $\mathbb{S}^{d-1} = \{v \in \mathbb{R}^d : ||v||_2 = 1\}$ for the $(d-1)$-dimensional sphere in $\mathbb{R}^d$. For a vector $x \in \mathbb{R}^d$ we use the notation $||x||_2 = \sqrt{\sum_{i=1}^d x_i^2}$ for the 2-norm of x. Let $||A||_2$ denote the spectral norm of A, i.e. $||A||_2 = \max_{v \in \mathbb{S}^{d-1}} ||Av||_2$.

Definition 2.1. *An algorithm $M : \mathbb{R}^{d \times d} \to R$ (where R is the range of M) is ϵ-differentially private if for all $A, B \in \mathbb{R}^{d \times d}$ such that $||A - B|| \leq 1$ and all $S \subseteq R$ one has $\mathbf{Pr}[M(A) \in S] \leq e^\epsilon \cdot \mathbf{Pr}[M(B) \in S]$.*

A very useful property of differentially private algorithms is

Lemma 2.1. ([7], Composability) *The sequential application of algorithms $\{M_i\}$, each giving $\{\epsilon_i\}$-differential privacy, gives $(\sum_i \epsilon_i)$-differential privacy.*

We also need the following well-known result. A function is said to have sensitivity α if unit perturbations of the input result in the value of the function changing by at most α.

Lemma 2.2. ([8], Laplacian mechanism) *The Laplacian mechanism with noise of magnitude α/ϵ gives ϵ-differential privacy for queries of sensitivity at most α.*

For our purposes an important example of a function with sensitivity 1 is the function that maps a $d \times d$ symmetric matrix A to its principal eigenvalue $\lambda_1(A)$. For two matrices $A, B \in \mathbb{R}^{d \times d}$ one has $|\lambda_1(A) - \lambda_1(B)| \leq 1$, i.e. $\lambda_1(A)$ has unit sensitivity. Thus, one can always release the value of the principal eigenvalue of a matrix after adding Laplacian noise. This will be useful in our rank-k approximation algorithm.

Input perturbation. We now specify the bounds that can be achieved in our setting via input perturbation. It can be readily verified that adding Laplacian noise of magnitude $\Theta(d\sqrt{d}/\epsilon)$ to each entry of the matrix is necessary and sufficient to ensure ϵ-differential privacy with respect to spectral norm. The largest eigenvector of such a noise matrix is $\Theta(d^2/\epsilon)$, i.e. nontrivial approximation to the principal eigenvector of A will be achieved whenever $||A||$ is larger than $\Theta(d^2/\epsilon)$. On the other hand, our techniques will yield nontrivial approximation when $||A||$ is larger than only about $\Theta(d/\epsilon)$.

(ϵ, δ)-differential privacy. In order to point out the differences between our work and the recent papers of [10, 3], we give the definition of (ϵ, δ)-differential privacy.

DEFINITION 2.2. *An algorithm $M : \mathbb{R}^{d\times d} \to R$ (where R is the range of M) is (ϵ, δ)-differentially private if for any pair $A, B \in \mathbb{R}^{d\times d}$, $||A - B|| \le 1$ and all $S \subseteq R$ one has $\mathbf{Pr}[M(A) \in S] \le e^{\epsilon} \cdot \mathbf{Pr}[M(B) \in S] + \delta$.*

Note that this guarantee states that for any pair A, B the distributions $M(A)$ and $M(B)$ are multiplicatively close except on a small portion of the space *that may depend on* A, B.

3 Rank-1 approximation

In this section we analyze the utility of the well-known exponential mechanism of [13] applied to the problem of obtaining differentially private rank-1 approximations. We will show later in section 5 that this performance is nearly optimal. We instantiate the exponential mechanism on the unit sphere in d dimensions with the scoring function at $z \in \mathbb{S}^{d-1}$ equal to the value of the quadratic form $z^T Az$:

$$f_A(z) = \frac{e^{\epsilon z^T Az}}{\int_{\mathbb{S}^{d-1}} e^{\epsilon z^T Az} dS} \tag{3.2}$$

for $z \in \mathbb{S}^{d-1}$. We formalize this mechanism for reference as

Algorithm 1 SAMPLE-1D-EXP(A, ϵ, δ)

1: Sample $x \in \mathbb{S}^{d-1}$ from the distribution (3.2)

Note that is not obvious how to sample efficiently from the distribution given by (3.2), and obtaining a polynomial time algorithm for this problem is the main goal of section 6 below. For now, we concentrate on the utility/privacy tradeoff offered by (3.2). Privacy of the exponential mechanism was proved in [13], and hence we concentrate on utility. We will use

FACT 3.1. *[2] For $0 < r < 2$, a cap of radius r on $\mathbb{S}^{n-1}$ has measure at least $\frac{1}{2}(r/2)^{n-1}$*

We prove

LEMMA 3.1. *The exponential mechanism, given a symmetric positive definite $d\times d$ matrix A with largest eigenvalue λ, outputs a vector $v \in \mathbb{S}^{d-1}$ such that*

$$\mathbf{E}[v^T Av] \ge (1 - \delta)\lambda_1$$

as long as $\lambda_1 \ge Cd/(\epsilon\delta)\log(1/\delta)$ for an absolute constant $C > 0$.

Proof. Let $B = \{x \in \mathbb{S}^{d-1} : x^T Ax \le \lambda_1(1-2\delta)\}$ denote the set of *bad* points and $G = \{x \in \mathbb{S}^{d-1} : x^T Ax \ge \lambda_1(1-\delta)\}$ denote the set of *good* points. Let X denote the output of Algorithm 1.

Let u denote the principal eigenvector of A. Then

$$\mathbf{Pr}[x \in G] \ge \mathbf{Pr}[|x^T u| \ge 1 - \delta].$$

Thus, Fact 3.1 implies that for the base measure on the sphere

$$\mu(G) \ge (1/2)(\cos^{-1}(1-\delta)/2)^{d-1} \ge e^{-cd\log(1/\delta)}$$

for some absolute constant $c > 0$, when δ is small. Suppose that $\lambda_1 \ge 2cd\log(1/\delta)/(\epsilon\delta)$.

Then since $\mu(B) \le 1$,

$$\begin{aligned}\frac{\mathbf{Pr}[X \in B]}{\mathbf{Pr}[X \in G]} &\le \frac{\max_{x\in B} f_A(x)}{\mu(G)\min_{x\in G} f_A(x)} \\ &\le \frac{\frac{1}{Z}e^{-4cd\log(1/\delta)}}{\frac{1}{Z}e^{-2cd\log(1/\delta)} \cdot e^{-cd\log(1/\delta)}} \\ &= e^{-cd\log(1/\delta)}\end{aligned}$$

Hence,

$$\mathbf{Pr}[X \in B^c] \ge 1 - e^{-\Omega(d\log(1/\delta))}.$$

Theorem 1.1 now follows from Lemma 3.1 and the fact that the exponential mechanism is ϵ-differentially private[13].

4 Rank-k approximation

In this section we analyze an 'approximate SVD' procedure for obtaining a rank-k approximation of an $d \times d$ matrix A. The intuition behind our algorithm is quite simple. We use the algorithm for obtaining a differentially private rank-1 approximation given in section 3 to repeatedly find a (differentially private) vector $v \in \mathbb{R}^d$ that nearly maximizes $v^T Av$ and subtract $(v^T Av)vv^T$ from A (note that in fact we would subtract $r \cdot vv^T$, where r is a noisy version of $v^T Av$ to ensure that this step is differentially private). If v was the exact maximizer of $v^T Av$ among unit norm vectors, this would be exactly SVD, and we would have that the norm of the residual matrix after k steps of this operation is exactly λ_{k+1}.

Since our vector v is only an approximate maximizer, it becomes harder to control the decrease of the spectral norm after we subtract v. Nevertheless, we show that the spectral norm decreases to $\lambda_{k+1} + \delta\lambda_1$ after k iterations, as long as λ_1 is sufficiently large. While this may seem weak, we show in the lower bounds section below that such behavior is unavoidable.

Aspect ratio. The lower bound on λ_k of matrices for which our algorithm obtains a good rank-k approximation depends on the dimension of the matrix, the utility and privacy desired, and additionally on a parameter that we refer to *aspect ratio*. For a matrix A

with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{k+1} \geq \ldots \geq \lambda_d$ the aspect ration γ equals λ_k/λ_1. Thus, computing rank-k approximations to matrices with substantially different top eigenvalues requires more data with our algorithm. It would be very interesting to determine if this is necessary for all differentially private algorithms or is an artifact of our approach.

Algorithm 2 Rank-k approximation (ϵ– differential privacy parameter, δ-accuracy, γ-aspect ratio)

1: $A_0 \leftarrow A$
2: $B_0 \leftarrow \mathbf{0}$
3: **for** $i = 1$ to k **do**
4: $\quad \hat{\lambda}_{max,i} \leftarrow \lambda_1(A_{i-1}) + \mathrm{Lap}\left(\frac{k}{\epsilon}\right)$
5: $\quad \hat{\lambda}_{min,i} \leftarrow \lambda_d(A_{i-1}) + \mathrm{Lap}\left(\frac{k}{\epsilon}\right)$
6: $\quad \hat{\lambda}_{min,i} \leftarrow \max\{0, -\hat{\lambda}_{min,i}\}$
7: $\quad$ **if** $\hat{\lambda}_{max,i} > \hat{\lambda}_{min,i}$ **then**
8: $\quad\quad v_i \leftarrow$ SAMPLE-1D-EXP$(A_{i-1}, \frac{\epsilon}{k}, \frac{(\delta\gamma)^2}{k^2})$
9: $\quad$ **else**
10: $\quad\quad$ **return FAIL**
11: $\quad$ **end if**
12: $\quad r_i \leftarrow v_i^T A_{i-1} v_i + \mathrm{Lap}\left(\frac{k}{\epsilon}\right)$.
13: $\quad B_i \leftarrow B_{i-1} + r_i \cdot v_i v_i^T$
14: $\quad A_i \leftarrow A_{i-1} - r_i \cdot v_i v_i^T$.
15: **end for**
16: **return** B_k

In order to analyze the approximate SVD process implemented in Algorithm 2, we will use the following expression for the determinant of a rank-1 update of a matrix A in terms of the determinant of A:

$$\det(A + uv^T) = (1 + v^T A^{-1} u)\det(A). \tag{4.3}$$

In particular, we will use the following form of (4.3)

$$\det(A - tvv^T) = (1 - tv^T A^{-1} v)\det(A), \tag{4.4}$$

where we set t equal to the Rayleigh quotient $v^T A v$ with Laplacian noise (in line 11 of Algorithm 2).

Denote the eigenvalues of A by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. Denote the corresponding eigenvectors by $u_1, \ldots, u_d$. It follows from (4.3) that the eigenvalues of $A' := A - tvv^T$ are given by the roots of

$$p_{A'}(x) = p_A(x)\left(1 - t\sum_{i=1}^{d} \frac{\langle v, u_i\rangle^2}{x - \lambda_i}\right), \tag{4.5}$$

where $p_A(x)$ is the characteristic polynomial of A. Define $f_A(x) = 1 - t\sum_{i=1}^{d} \frac{\langle v,u_i\rangle^2}{x-\lambda_i}$. Our main technical lemma here is

LEMMA 4.1. *Let A be a symmetric matrix. There exists a constant $C > 0$ such that the following holds. Let v be a vector of unit l_2 norm such that $v^T A v \geq \lambda_1(1 - \delta/C)$, where $\delta \in (0, 1)$. Let $A' = A - tvv^T$, where $t \in (1 \pm \delta/C) \cdot v^T A v$. Denote the eigenvalues of A' by $\lambda'_1 \geq \lambda'_2 \geq \ldots \geq \lambda'_d$. Then*

1. *$\lambda_k \leq \lambda'_{k-1} \leq \min\{\lambda_{k-1}, \lambda_k + \delta\lambda_1\}$ for each $k = 1, \ldots, d$;*
2. *$\lambda'_d \geq \lambda_d - \sqrt{\delta}\lambda_1$.*

The proof of the lemma relies crucially on the characterization in (4.5), is mostly technical and is deferred to the full version of the paper due to space constraints.

REMARK 4.1. *The bounds in Lemma 4.1 are essentially best possible. For part (1), it is sufficient to consider a matrix A with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ such that $\lambda_2 \geq (1 - \delta)\lambda_1$, and note that under the assumptions of the lemma the vector v could coincide with the eigenvector corresponding to λ_2, leading to spectral norm $\lambda_1 \leq \lambda_2 + \delta\lambda_1$, as predicted.*

For part (2), this can be easily seen as follows. Let $A = e_1 e_1^T$. Let $v = (\sqrt{1 - x^2}, x)^T$. Then $v^T A v = 1 - x^2$, so v satisfies the conditions of Lemma 4.1 with $\delta = x^2$. On the other hand, the eigenvalues of $A - vv^T$ are $\pm x$. Thus, the smallest eigenvalue the matrix obtained by subtracting vv^T from A is $-\sqrt{\delta}$.

We now prove performance guarantees for Algorithm 2. First, we prove guarantees that depend on the aspect ratio of the input matrix.

LEMMA 4.2. *Let A be a symmetric positive semidefinite matrix with $\lambda_k \geq Cdk^3/(\epsilon\gamma^2\delta^3)$, where C is the constant from Lemma 4.1. Then Algorithm 2 outputs a rank-k matrix B_k such that $||A - B_k|| \leq \lambda_{k+1} + \delta\lambda_k$. Furthermore, Algorithm 2 is 4ϵ-differentially private with respect to the spectral norm.*

Proof. Denote the eigenvalues of A_i by $\lambda_1^i, \ldots, \lambda_d^i$, $i = 0, \ldots, k$. By choosing constants appropriately by Lemma 3.1 we have for each $i = 1, \ldots, k$

$$\mathbf{Pr}\left[v_i^T A_{i-1} v_i \leq \lambda_1^i - \frac{dk^3 \log(k/\delta)}{4\epsilon\delta^2}\right] < e^{-\Omega(d\log(k/\delta))}, \tag{4.6}$$

where $v \in \mathbb{S}^d$ is the vector sampled in lines 6-10 of Algorithm 2.

We prove the following claims for all $i = 0, \ldots, k$.

(1) For all $j \leq k + 1 - i$ one has $\lambda_{j+i}^0 \leq \lambda_j^i \leq \lambda_{j+i}^0 + \frac{(\delta\gamma)^2 i}{k^2}\lambda_1$;

(2) $\lambda_d^i \geq -i \cdot \frac{\delta\gamma}{k} \cdot \lambda_1$;

(3) $v_i^T A_{i-1} v_i \geq (1 - \delta/(2C))\lambda_1^{i-1}$ for $i \geq 1$;

(4) $r_i \in (1 \pm \delta/C)\lambda_1^{i-1}$ for $i \geq 1$.

The proof is by induction on i.

Base: $i = 0$ (1) and (2) are trivially true and (3) and (4) are vacuous.

Inductive step: $i \to i+1$:

(3) and (4) By the inductive hypothesis we have **(a)** $\lambda_1^i \geq \lambda_{i+1}^0 \geq Cdk^3/(\epsilon\delta^3)$, so (4.6) implies $v_{i+1}^T A_i v_{i+1} \geq (1 - \delta/(2C))\lambda_1^i$. Let $Y \sim \text{Lap}(k/\epsilon)$ denote the Laplacian random variable such that $r_{i+1} = v_{i+1}^T A_i v_{i+1} + Y$. Since

$$\mathbf{Pr}[|Y| > Cdk^3/(\epsilon\delta)] < e^{-\Omega(dk^2/\delta)},$$

we get that whp $r_{i+1} \in (1 \pm \delta/C)v_{i+1}^T A_i v_{i+1}$.

(1) and (2) By the inductive hypothesis we have for all $j \leq k+1-i$ that **(a)** $\lambda_j^i \geq \lambda_{j+i}^0 \geq Cdk^3/(\epsilon\delta^3)$. By (3) and (4) that we just proved for $i+1$ we have **(b)** $v_{i+1}^T A_i v_{i+1} \geq (1 - \delta/(2C))\lambda_1^i$ and **(c)** $r_{i+1} \in (1 \pm \delta/C)\lambda_1^i$. This together with Lemma 4.1 implies

$$\lambda_{j+1}^i \leq \lambda_j^{i+1} \leq \lambda_{j+1}^i + \frac{(\delta\gamma)^2}{k^2}\lambda_1$$

for $j+1 \leq d$. Furthermore, since by the inductive hypothesis $\lambda_{j+i}^0 \leq \lambda_j^i \leq \lambda_{j+i}^0 + \frac{(\delta\gamma)^2 i}{k^2}\lambda_1$ for all $j \leq k+1-i$, we get that

$$\lambda_{j+i+1}^0 \leq \lambda_j^{i+1} \leq \lambda_{j+i+1}^0 + (i+1)\cdot\frac{(\delta\gamma)^2}{k^2}\lambda_1$$

for all $j+1 \leq k+1-i$, i.e. $j \leq k+1-(i+1)$, proving the inductive step for (1).

Similarly, by the inductive hypothesis we have $\lambda_d^i \geq -i \cdot \frac{\delta\gamma}{k} \cdot \lambda_1$, so Lemma 4.1, (2) implies that $\lambda_d^{i+1} \geq \lambda_d^i - \frac{\delta\gamma}{k} \cdot \lambda_1^i \geq -(i+1) \cdot \frac{\delta\gamma}{k} \cdot \lambda_1$.

Thus, we have proved that $\lambda_1^k \leq \lambda_{k+1}^0 + (\delta\gamma)^2\lambda_1$ and $|\lambda_d^k| \leq (\delta\gamma)\lambda_1 \leq \delta\lambda_k$. First, this implies that the procedure succeeds whp, i.e. line 7 always returns **true**. Second, this implies that $||A^k|| \leq \max\{\lambda_{k+1}^0 + (\delta\gamma)^2\lambda_1, \delta\gamma\lambda_1\} \leq \lambda_{k+1}^0 + \delta\lambda_k$ as required.

Finally, ϵ-differential privacy with respect to the spectral norm follows by observing that SAMPLE-1D-EXP is run with privacy parameter ϵ/k, so releasing the output from k such invocations gives ϵ-differential privacy. Furthermore, the addition of Laplacian noise of magnitude ϵ/k in lines 4, 5, 11 ensures that (a) $\hat{\lambda}_{max,i}, \hat{\lambda}_{min,i}$ can be released and compared publicly, yielding $2\epsilon/k$-differential privacy for each step, and (b) r_i can be released so that releasing $r_i v_i v_i^T$ is ϵ/k-differentially private. Summing up the privacy parameters for all steps, we conclude that the procedure is 4ϵ-differentially private. ∎

We now use Lemma 4.2 to obtain guarantees that only rely on an assumption on λ_1. Let $A \in \mathbb{R}^{d\times d}$ be a symmetric positive semidefinite matrix and let $\lambda_1 \geq \ldots \geq \lambda_k \geq \ldots \geq \lambda_d$ denote the eigenvalues of A. Furthermore, suppose that $\lambda_1 \geq Cdk^3/(\epsilon\delta^6)$. Let $\gamma = \delta$ and let $\lambda_{k'}$ be the smallest such that $\lambda_{k'} \geq \max\{\gamma\lambda_1, Cdk^3/(\epsilon\gamma^2\delta^3)\} = \delta\lambda_1$. If $k' \geq k$, then we are done by running Algorithm 2. Otherwise by Lemma 4.2 running the algorithm for k' steps yields spectral norm at most $\lambda_{k'+1} + \delta\lambda_{k'} \leq \lambda_{k'+1} + \delta\lambda_1 \leq \lambda_{k+1} + 2\delta\lambda_1$.

Finally, it is easy to see that running the algorithm for k' steps, where k' is as defined above, can be done in a differentially private manner. Indeed, one can calculate noisy versions of at most k top eigenvalues of A as it is done in Algorithm 2 until one finds k'. By our assumptions on k' the addition of Laplacian noise will not introduce significant inaccuracies by the same argument as in the proof of correctness of Algorithm 2. We have proved Theorem 1.2, which we also state here for convenience:

Theorem 18. *Let $A \in \mathbb{R}^{d\times d}$ be a symmetric psd matrix and let $\lambda_1 \geq \ldots \geq \lambda_d$ denote the eigenvalues of A. There exists an ϵ-differentially private polynomial time algorithm for computing a matrix A_k of rank at most k so that $||A - A_k||_2 \leq \lambda_{k+1} + \delta\lambda_1$ as long as $\lambda_1 \geq C_1 dk^3/(\epsilon\delta^6)$ for a constant $C_1 > 0$.*

5 Lower bounds

In this section we present lower bounds for the tradeoff between privacy and approximation quality for rank-1 and rank-k approximation. Since the bounds for rank-1 approximation are somewhat simpler, we present them first, showing that the rank-1 approximation algorithm from section 3 is essentially optimal for constant factor multiplicative approximation. Both bounds follow by packing arguments on the $(d-1)$-dimensional sphere for rank-1 approximation and on the Grassmannian manifold for general k.

5.1 Rank-1 approximation

It is convenient to introduce the following notation for spherical caps. For a vector $u \in \mathbb{S}^{d-1}$ and $\theta \in (0,1)$ let

$$C_\theta(u) := \{v \in \mathbb{S}^{d-1} : \langle v, u\rangle \geq 1 - \theta\}. \tag{5.7}$$

For the lower bound, we will use

FACT 5.1. *For any $\theta \in (0,1)$ there exists a family of vectors $u_1, \ldots, u_N$ of unit norm with $N = e^{\Omega(d\log(1/\theta))}$ such that $C_\theta(u_i) \cap C_\theta(u_j) = \emptyset$ for $i \neq j$.*

Proof. Such a family can be obtained by a simple greedy packing on the sphere. ∎

THEOREM 5.1. *Let $\epsilon > 0$ be a fixed privacy parameter. Suppose that there exists an ϵ-differentially private algorithm that, given any a symmetric positive semidefinite $d \times d$ matrix A with maximum eigenvalue λ_1, outputs a vector $v \in \mathbb{R}^d$ of unit l_2 norm such that*

$$v^T A v \geq (1-\delta)\lambda_1,$$

as long as $\lambda_1 \geq \lambda_1^*(d,\epsilon,\delta)$. Then $\lambda_1^*(d,\epsilon,\delta) = \Omega(d\log(1/\delta)/\epsilon)$.

Proof. Let $\mathcal{F} = \{u_1,\ldots,u_N\}$, $N = e^{\Omega(d\log(1/\delta))}$ denote a family of vectors such that for $u_i, u_j \in \mathcal{F}, i \neq j$ one has $C_{1-\sqrt{1-2\delta}}(u_i) \cap C_{1-\sqrt{1-2\delta}}(u_j) = \emptyset$. The existence of such a family is guaranteed by Fact 5.1 together with the fact that $1-\sqrt{1-2\delta} \geq \delta$ for all $\delta \in [0,1/2]$.

Now let $A_i = \gamma(d/\epsilon)\log(1/\delta)\cdot u_i u_i^T$, where $\gamma > 0$ is a constant that we will choose to be sufficiently small. Suppose that the algorithm, when given $A = A_i$ as input, outputs a vector such that

$$\mathbf{E}_{A_i}[\lambda_1 - x^T A x] \leq \delta\lambda_1,$$

where the subscript A_i means 'when the algorithm is given A_i as input'. Then by Markov's inequality

$$\mathbf{Pr}_{A_i}[\lambda_1 - x^T A x \leq 2\delta\lambda_1] \geq 1/2. \tag{5.8}$$

However, the set of vectors x that satisfy (5.8) is exactly the set of vectors x that satisfy $\langle x, u_i\rangle \geq \sqrt{1-2\delta}$. Setting $\theta = 1-\sqrt{1-2\delta} \leq 2\delta$, we get that at least half of the probability mass on input A_i should go into its spherical cap of radius $\cos^{-1}(1-2\delta)$ around u_i.

On the other hand, since $||A_i - A_j||_2 \leq 2\gamma(d/\epsilon)\log(1/\delta)$, we have by the differential privacy guarantee that each A_i should put at least $e^{-2\gamma(d\log(1/\delta))}$ mass into $C_{1-\sqrt{1-2\delta}}(u_i)$ for each $u_i \in \mathcal{F}$. Hence, one necessarily has

$$e^{-2\gamma d\log(1/\delta)}\cdot e^{\Omega(d\log(1/\delta))} \leq 1,$$

a contradiction for sufficiently small constant $\gamma > 0$. ∎

5.2 Rank-k approximation In this section we give lower bounds on the privacy/accuracy tradeoff for rank-k approximation. The bounds follow by a careful packing argument on the Grassmannian manifold, which we now outline. As before, we assume that an ϵ-differentially private algorithm outputs a rank-k matrix $\hat{A}$ such that

$$||A - \hat{A}||_2 \leq \lambda_{k+1} + \delta\lambda_1$$

for some $\delta > 0$ as long as $\lambda_1 \geq \lambda^*(d,k,\epsilon,\delta)$, and derive a lower bound on $\lambda^*(d,k,\epsilon,\delta)$.

The lower bound will follow by considering matrices $A = \alpha YY^T$, where Y is a $d \times k$ matrix with orthogonal columns of unit l_2 norm and α is a scaling to be chosen later. Our first step is to show that the approximating matrix $\hat{A}$ can essentially be assumed to be a projection matrix onto a k-dimensional subspace.

LEMMA 5.1. *[Projection matrices suffice] Let $A = YY^T$ and let $\hat{A} = Z\Sigma Z^T$, where Y and Z are $d \times k$ matrices with orthonormal columns such that $||A - \hat{A}|| \leq \delta$. Then $||A - ZZ^T|| \leq 2\delta$.*

Proof. We first show that $\Sigma_{ii} \geq 1-\delta$ for $i = 1,\ldots,k$. Consider

$$Y^T(YY^T - Z\Sigma Z^T)Y = I - (Z^TY)^T\Sigma(Z^TY) \tag{5.9}$$

Suppose that $\Sigma_{ii} < 1-\delta$. We consider two cases.

(1) Suppose that Z^TY is singular, i.e. there exists $v \in \mathbb{R}^k$ such that $Z^TYv = 0$. Multiplying (5.9) by v and v^T then yields

$$v^TY^T(YY^T - Z\Sigma Z^T)Yv = I > \delta, \tag{5.10}$$

a contradiction since $||Y^T(YY^T - Z\Sigma Z^T)Y||_2 \leq ||Y^T||\cdot||YY^T - Z\Sigma Z^T||\cdot||Y||_2 \leq \delta$.

(2) Now suppose that Z^TY is non-singular. Let v be a unit norm vector such that $Z^TYv = \lambda Z_i$ for a constant i, where i is the i-th column of Z, where $|\lambda| \leq ||Z^TY|| \leq 1$. Plugging this into (5.9), we get

$$v^TY^T(YY^T - Z\Sigma Z^T)Yv > 1-(1-\delta) = \delta, \tag{5.11}$$

a contradiction as before.

We now show that $\Sigma_{ii} < 1+\delta$ for all i. Indeed, otherwise letting v_i denote the i-th columns of Z, we have

$$v_i^T(A - \hat{A})v_i < v_i^TAv_i - (1+\delta) < -\delta.$$

Thus, we have that $1-\delta \leq \Sigma_{ii} \leq 1+\delta$, which now implies that $||YY^T - ZZ^T|| \leq 2\delta$. Indeed, suppose that there exists a vector $x \in \mathbb{R}^d$ such that $||x||_2 = 1$ and $|(YY^T - ZZ^T)x| > 2\delta$. Then

$$|(YY^T - Z\Sigma Z^T)x| > |(YY^T - Z\Sigma Z^T)x| - \delta = \delta,$$

contradicting our assumption. ∎

Thus, a rank-k matrix $\hat{A} = Z\Sigma Z^T$ is a good approximation to $A = YY^T$, then ZZ^T is also a good approximation to A. We use this fact now to derive utility/privacy tradeoffs for our problem.

Packing on the Grassmannian manifold. Recall that the Grassmannian manifold $\mathbf{G}_{k,d}$ is the set of

k-dimensional subspaces in $\mathbb{R}^d$. We will represent points in $\mathbf{G}_{k,d}$ by $d \times k$ matrices with orthonormal columns $Y \in \mathbb{R}^{d\times k}$, with the understanding that Y represents the k-dimensional space that it spans and thus is defined up to an element of the orthogonal group in the subspace. Since we will only use Y to construct the projection matrix YY^T, this will not matter. For $Y \in \mathbf{G}_{k,d}$ define

$$C_\delta^k(Y) = \{S \in \mathbf{G}_{k,d} : ||YY^T - SS^T||_2 \leq \delta\}. \tag{5.12}$$

We would like to find a large family $\mathcal{F} = \{Y^1, \ldots, Y^N\}, N = 2^{\Omega(k(d-k)\log(1/\delta))}$ such that $C_\delta^k(Y^i) \cap C_\delta^k(Y^j) = \emptyset$ for $i \neq j$. To prove the existence of such a family we need upper bounds on the volume of $C_\delta^k(Y)$ for $Y \in \mathbf{G}_{k,d}$.

Volume of $C_\delta^k(Y)$. Let $Y, S \in \mathbf{G}_{k,d}$ be two uniformly random subspaces. The quantity $||YY^T - SS^T||_2$ is the cosine of the largest canonical angle between Y and S, and is given by the largest singular value of $Y^T S$. The canonical angle between two subspace is a metric [17]. The distribution of the canonical angle between two uniformly random subspaces is given by [1]

$$\begin{aligned}\mathbf{Pr}(\theta_k < \hat{\theta}_k) &= \gamma_{n,k} \cdot (\sin\hat{\theta}_k)^{k(d-k)-1} \\ &\quad \cdot {}_2F_1\left(\frac{d-k}{2}, \frac{1}{2}; \frac{d+1}{2}; \sin^2\hat{\theta}_k I_{k-1}\right),\end{aligned} \tag{5.13}$$

where ([14])

$$\begin{aligned}{}_2F_1(a,b;c;X) &= \frac{\Gamma_k(c)}{\Gamma_k(a)\Gamma_k(c-a)} \\ &\quad \int_{0<Y<I_k} \det(I-XY)^{-b}(\det Y)^{a-(k+1)/2} \\ &\quad \det(I-Y)^{c-a-(k+1)/2}dY,\end{aligned} \tag{5.14}$$

$\Gamma_k(a) = \pi^{k(k-1)/4}\prod_{i=1}^k \Gamma[a - \frac{1}{2}(i-1)]$ and $c_{n,k} = k(n-k)\frac{\Gamma(\frac{k+1}{2})\Gamma(\frac{d-k+1}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{d+1}{2})}$. We will use the following crude estimate on (5.13):

CLAIM 5.2. *Let $\sin\hat{\theta}_k = \delta$, so that $\hat{\theta}_k = \Theta(\delta)$. Then $\mathbf{Pr}[\theta_k < \hat{\theta}_k] = e^{-O(k(d-k)\log(1/\delta))}$.*

Proof. By inspection of (5.14) we have that ${}_2F_1\left(\frac{d-k}{2}, \frac{1}{2}; \frac{d+1}{2}; z \cdot I_{k-1}\right) \leq {}_2F_1\left(\frac{d-k}{2}, \frac{1}{2}; \frac{d+1}{2}; I_{k-1}\right)$. Thus, it follows from (5.13) that

$$\begin{aligned}\mathbf{Pr}(\theta_k < \hat{\theta}_k) &\leq (\sin\hat{\theta}_k)^{k(d-k)-1}\mathbf{Pr}(\theta_k < \pi/2) \\ &= e^{-O(k(d-k)\log(1/\delta))}.\end{aligned}$$

■

It now follows that

COROLLARY 5.1. *For each $\delta > 0$ there exists family $\mathcal{F} = \{Y^1, \ldots, Y^N\}, N = 2^{\Omega(k(d-k)\log(1/\delta))}$, where $Y^i \in \mathbf{G}_{k,d}$, such that $C_\delta^k(Y^i) \cap C_\delta^k(Y^j) = \emptyset$ for $i \neq j$.*

Proof. We obtain such a family by a simple greedy packing. Let Y^1 be an arbitrary element of $\mathbf{G}_{k,d}$. For each $i = 1, \ldots, N$ we have by Claim 5.2 that

$$\mathbf{G}_{k,d} \setminus \bigcup_{j=1}^{i} C_{2\delta}^k(Y^j) \neq \emptyset$$

unless $N \geq 2^{ck(d-k)\log(1/\delta)}$ for a constant $c > 0$. Thus, we have found N elements $Y^i \in \mathbf{G}_{k,d}$ such that $||Y^i(Y^i)^T - Y^j(Y^j)^T||_2 > 2\delta$ for $i \neq j$. It now follows by triangle inequality that $C_\delta(Y^i) \cap C_\delta(Y^j) = \emptyset$, as required. ■

We now obtain

THEOREM 5.3. *Any ϵ-differentially private algorithm that given a symmetric positive definite matrix A with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ outputs a rank-k approximation $\hat{A}_k$ to A such that*

$$||A - \hat{A}_k||_2 \geq \lambda_{k+1} + \delta\lambda_1$$

for some $\delta > 0$, as long as $\lambda_1 \geq \lambda^*(d,k,\epsilon,\delta)$. Then $\lambda^*(d,k,\epsilon,\delta) = \Omega\left(\frac{1}{\epsilon}k(d-k)\log(1/\delta)\right)$.

Proof. Let $\mathcal{F} = \{Y_1, \ldots, Y_N\}, N = 2^{\Omega(k(d-k)\log(1/\delta)}$ denote the family of subspaces guaranteed by Corollary 5.1, so that for each $i \neq j$ one has

$$C_{4\delta}^k(Y^i) \cap C_{4\delta}^k(Y^j) = \emptyset.$$

Consider matrices $A^i = \gamma Y_i Y_i^T, i = 1, \ldots, N$, so that $||A^i|| = \gamma$. By assumption of the lemma the algorithm outputs a matrix $\hat{A}_k$ such that

$$\mathbf{E}_{A^i}\left[||A^i - \hat{A}_k^i||_2\right] \leq \delta\gamma,$$

where as before the subscript A^i refers to the fact the expectation is over the distribution of the algorithm when the input is A^i.

Let $\hat{A}_k^i = \hat{Y}^i\hat{\Sigma}(\hat{Y}^i)^T$, where Y is a $d \times k$ matrix with orthonormal columns. Let $\tilde{A}_k^i := \hat{Y}^i(\hat{Y}^i)^T$. By Lemma 5.1 we have

$$\mathbf{E}_{A^i}\left[||A^i - \tilde{A}_k^i||_2\right] \leq 2\delta\gamma.$$

Now by Markov's inequality one has

$$\mathbf{Pr}_{A^i}\left[||A^i - \tilde{A}_k^i||_2 \leq 4\delta\gamma\right] > 1/2,$$

i.e. the algorithm necessarily places at least half the probability mass into $C^k_{4\delta}(Y^i)$. However, since the algorithm ϵ-differential privacy, we also get that the algorithm necessarily puts at least $e^{-2\epsilon||A^i||} = e^{-2\epsilon\gamma}$ mass into each of $C^k_{4\delta}(Y^i)$, implying that

$$e^{-2\epsilon\gamma} \cdot 2^{\Omega(k(d-k)\log(1/\delta))} \leq 1.$$

This means that $\gamma = \Omega\left(\frac{1}{\epsilon}k(d-k)\log(1/\delta)\right)$, a contradiction with our assumption on γ. ∎

6 Implementing the exponential mechanism

In this section we give a polynomial time implementation of the exponential mechanism for outputting a differentially private rank-1 approximation to a symmetric matrix from section 3. Recall that the exponential mechanism simply samples $x \in \mathbb{S}^{d-1}$ from the distribution

$$f_A(x) = \frac{e^{\epsilon x^T A x}}{\int_{\mathbb{S}^{d-1}} e^{\epsilon s^T A s} dS}. \tag{6.15}$$

Standard approaches to sampling. The main difficulty in sampling from (6.15) stems from the fact that the function $x^T A x$ is in general *non-convex* on $\mathbb{S}^{d-1}$, making standard techniques for sampling from exponential size spaces inapplicable. It is also easy to see that acceptance/rejection from the Gaussian distribution would take exponential time. Another natural approach would be to map the set $S_A(r)$ via stereographic projection to $\mathbb{R}^{d-1}$ and use techniques for sampling from log-concave distributions[11]. However, the density obtain via this transformation is unfortunately not log-concave. We now present our approach to sampling from (6.15) in polynomial time. In the some parts of this section we abuse notation somewhat and use the variable ϵ with a different meaning (in particular, in the definition of an (ϵ, δ)-mesh).

In what follows we will be interested in estimating volumes of sets of the form

$$S_A(r) = \left\{x \in \mathbb{S}^{d-1} : x^T(\lambda_1 I - A)x \leq r^2\right\}.$$

for $r > 0$, where A is a $d \times d$ symmetric matrix. We will use the notation $V_A(r) = \mu_{d-1}(S_A(r))$, where μ_{d-1} is the $(d-1)$-dimensional uniform measure on $\mathbb{S}^{d-1}$. When the matrix A is fixed, we will drop the subscript for brevity and simply write $V(r)$. Note that $V(r)$ is increasing in r and $V(0) = 0, V(\sqrt{\lambda_1 - \lambda_d}) = 1$. We first define a modified version of the Γ distribution:

$$\Gamma_A(t, \epsilon) = Z_A^{-1} \cdot e^{-\epsilon t} V_A(\sqrt{t}), \tag{6.16}$$

where $Z_A = \int_{t=0}^{\infty} e^{-\epsilon t} V_A(\sqrt{t}) dt$ is a normalizing constant. Note that $\Gamma_A(t, \epsilon)$ is indeed a version of the Γ distribution, where the volume of a ball of radius t is replaced by the volume of the set $S_A(\sqrt{t})$.

If we have access to estimates of $V_A(r)$ as well as a procedure for sampling uniformly from $S_A(r)$, then sampling from (6.15) can be done as follows:

Algorithm 3 SAMPLE-1D-EXP(A, ϵ, δ)

1: Sample $t \sim \Gamma_A(\epsilon)$
2: Sample X uniformly from $S_A(\sqrt{t})$.

We now verify that Algorithm 3 indeed gives (6.15). Fix $x \in \mathbb{S}^{d-1}$. Then

$$\begin{aligned}\mathbf{Pr}[X = x] &= Z_A^{-1} \cdot \int_{x^T(\lambda_1 I - A)x}^{\infty} \epsilon e^{-\epsilon s} ds \\ &= Z_A^{-1} \cdot e^{-\epsilon x^T(\lambda_1 I - A)x} \sim e^{\epsilon x^T A x}.\end{aligned}$$

Note that while it appears that (6.16) requires using arithmetic with at least a polynomial in d number of bits, we will show below how to obtain estimates of $V_A(t)$ and sample from this distribution in polynomial time with only $O(\log d)$ bit arithmetic. We will then give a polynomial-time procedure for sampling from an approximately uniform distribution on $S_A(\sqrt{t})$.

Eigenvalue separation. In the rest of the paper we will assume *eigenvalue separation* for the matrix A. This is without loss of generality for the following reason. Let $A = \sum_{i=1}^d \lambda_i u_i u_i^T$. Let $E := \sum_{i=1}^d (i/d) u_i u_i^T$, so that $||E||_2 \leq 1$. Then one has $A + E = \sum_{i=1}^d (\lambda_i + i/d) u_i u_i^T$, i.e. $\lambda_i - \lambda_{i+1} \geq 1/d$. On the other hand, since our mechanism is ϵ-differentially private with respect to the *spectral* norm, privacy is preserved since $||E||_2 \leq 1$. We will let $\Delta = 1/d$ denote a lower bound on eigenvalue separation of A.

Overview of techniques. Before going into the details of our implementation, we give a brief overview of our techniques. Our main goal in this section is to be able to approximate volumes of sets $S_A(r)$ for $r \in [0, \sqrt{\lambda_1 - \lambda_d}]$. The main complication that arises is that these sets are in general not convex, making standard techniques inapplicable. In order to overcome this, we design a scheme that approximates volumes of these sets recursively by evaluating appropriate integrals.

The main technical contribution here consists of estimates of the growth rate of the volume of the sets $S_A(r)$ as a function of r. In particular, we show that one can construct a polynomial size mesh over the set of possible values of r such that the *multiplicative change* in the volume of $S_A(r)$ is bounded by $e^{O(\epsilon)}$ over each interval in the mesh. We show that a mesh with $\tilde{O}(d/(\Delta\epsilon))$ points is sufficient for approximating volumes to precision that is sufficient for our purposes.

Here d is the dimension of the input matrix and Δ is a lower bound on eigenvalue separation.

Another important issue that we need to overcome is ensuring that the recursive estimation of volumes produces small error even though we are estimating exponentially small volumes using $O(\log d)$ bit arithmetic. This is nontrivial since the estimation procedure will in general blow up estimation errors made in the recursive calls. However, we will show that a certain *multiplicative-additive* error remains properly bounded. Our main tool here will be estimating the error in terms of the volume of the intersection of $S_A(r)$ with *thin slices* of $\mathbb{S}^{d-1}$. Analysis of this intersection will also be helpful in proving growth bounds on the volume of $S_A(r)$.

Thin slices of $\mathbb{S}^{d-1}$. Here and below we work in the eigenbasis of the matrix A. Furthermore, wlog we restrict our attention to $x \in \mathbb{S}^{d-1}$ with all nonnegative coordinates since the quadratic form of interest is insensitive to flipping the sign of any coordinate. For $z > 0$ let

$$H(z) = \left\{x \in \mathbb{S}^{d-1} : \exists i, |x_i| < z\right\},$$

i.e. the union of slices of $\mathbb{S}^{d-1}$ sandwiched between coordinate hyperplanes $\pm z$ away from the origin. Also, let $\bar{H}(z) := \mathbb{S}^{d-1} \setminus H(z)$. We will chose a parameter $z^* = 1/\text{poly}(d, \lambda_1 - \lambda_d)$ and use $H(z^*)$ in our estimates throughout this section. Our procedure for estimating the volume of $S_A(r)$ will have error bounded by the volume of $S_A(r) \cap H(z^*)$ for all r, where z will be related to the precision of our arithmetic. This will allow us to get good estimates of sets $S_A(r)$ for value of r sufficiently larger than z^*. Note that the runtime of our algorithm will depend on $\log(1/z^*)$, which will be $O(\log \text{poly}(d, \lambda_1 - \lambda_d))$.

We now show that the volume of $S_A(r) \cap \bar{H}(z^*)$ is multiplicatively close to the volume of $S_A(r)$ except when r is tiny. Consider the mapping $f(x) = \frac{x+\Delta}{||x+\Delta||_2}$, where $\Delta = (Z \cdot \delta_1, \delta_1^2, \ldots, \delta_1^2)^T$. We will choose the parameter $Z = \text{poly}(d, \lambda_1 - \lambda_d)$ and $\delta_1 = \Theta\left(\frac{1}{(\lambda_1 - \lambda_d)\text{poly}(d)}\right)$ (we will have $Z \cdot \delta_1 = 1/\text{poly}(d, \lambda_1 - \lambda_d)$). Note that the mapping increases all coordinates by a small amount δ_1^2, moves an appropriate amount of mass to the first coordinate and normalizes the resulting vector.

LEMMA 6.1. *Fix $r^* = 1/poly(d, \lambda_1 - \lambda_d)$. Then for a sufficiently large $Z = poly(d, 1/\epsilon, \lambda_1 - \lambda_d)$ and sufficiently small $\delta_1 = 1/poly(d, 1/\epsilon, \lambda_1 - \lambda_d)$ The mapping $f(\cdot)$ satisfies*

(Inclusion) *$f(S_A(r)) \subseteq S_A(r)$ for all $r \geq r^*$;*

(Small compression) *for all $W \subset \mathbb{S}^{d-1}$ one has $vol\,(f(W)) \geq (1 - \epsilon/d^3) vol\,(W)$.*

The proof follows by lower bounding the Jacobian of the transformation f and is deferred to the full version of the paper due to space constraints. We can now show that one can restrict attention to $S_A(r) \cap \bar{H}(z)$ if z is sufficiently small in comparison to r, with only a small multiplicative loss in volume:

COROLLARY 6.1. *Fix $r^* = 1/poly(d, \lambda_1 - \lambda_d)$. There exists a choice of $z^* = 1/poly(d, \lambda_1 - \lambda_d)$ and $\delta_1 = 1/poly(d, \lambda_1 - \lambda_2)$ such that $f(\mathbb{S}^{d-1}) \subseteq \mathbb{S}^{d-1} \cap \bar{H}(z^*)$ and for any $r \geq r^*$ one has*

$$\mu(S_A(r) \cap \bar{H}(z^*)) \geq (1 - \epsilon/d^3)\mu(S_A(r)).$$

Proof. By the first property of Lemma 6.1, the function f maps $S_A(r)$ into $S_A(r) \cap \bar{H}(z^*)$. By the second property we have vol $(f(S_A(r))) \geq (1 - \epsilon/d^3)$vol $(S_A(r))$, which yields the result since $f(S_A(r)) \subseteq S_A(r) \cap H(z^*)$. ∎

From now on we consider the values of r^* and z^* fixed so that discarding thin slices $H(z^*)$ from the sphere yields a $1-\epsilon/d^3$ approximation to the volume of $S_A(r), r \geq r^*$.

A list of thresholds. Our estimates below will use several important thresholds for the radius r of the set $S_A(r)$, which dictate the precision of our arithmetic. We gather these thresholds here in the order in which they are set and state the relations between them:

A: **(r_{conv})** For all $r \leq r_{conv}$ the set $S_A(r)$ is convex on $\mathbb{S}^{d-1}$ **and** is essentially flat, i.e. sampling from $S_A(r)$ can be done easily by sampling from an appropriate ellipse (see section 6.5)

B: **(Δ_r)** Change in the radius that we introduce during sampling to avoid radii close to eigenvalues of A, chosen so that $V(r + d \cdot \Delta_r) \in (1 \pm 1/d^2)V(r)$ for all $r \geq r_{conv}$.

C: **(r^*)** We will use recursive estimates for the volume $V_A(r)$ for all $r \geq r^*$ in the main sampling loop in Algorithm 5. Furthermore, we need $r^* < \epsilon\Delta_r/(4d^2\sqrt{\lambda_1 - \lambda_d})$.

D: **(z^*)** Choose z^* small enough so that for all $r \geq r^*$ one has that $\mu(S_A(r) \cap \bar{H}(z^*)) \geq (1 - \epsilon/d^3)\mu(S_A(r))$.

E: **(r_{min})** We will compute recursive estimates of volumes for all $r \geq r_{min}$. Choose r_{min} so that the volume of any spherical cap around $(1, 0, \ldots, 0)$ that fits inside $H(z^*)$ is at most r_{min}^{k-1}.

F: **(Δ^*)** Choose Δ^* so that for all $r \geq r_{min}$ one has $V_A(r + \Delta^*) \leq e^\epsilon V_A(r)$. This will be used as a weak estimate on the growth of $V_A(r)$ close to eigenvalues of A to construct an (ϵ, δ)-mesh for $V_A(r)$ (defined below).

In general, we have $\Delta^* < r_{min} < z^* < r^* < \Delta_r < r_{conv}$. All quantities will be on the order of $1/\text{poly}(d, \lambda_1 - \lambda_d)$, contributing a $\log \text{poly}(d, \lambda_1 - \lambda_d)$ term to the runtime.

6.1 Estimating volume growth We now bound the growth of $V(r)$ as a function of r. We prove our bounds in two regimes. First, we prove bounds that depend on the distance between r and the closest eigenvalues of the matrix A (we prove two inequalities that depend on the distance to the closest eigenvalue on the left and right respectively). The third bound works independently of the distance to eigenvalues, but gives worse parameters than the first two.

We will need the following simple

LEMMA 6.2. *Fix an interval $I := [\lambda_i, \lambda_{i-1}]$ and assume that $\lambda_{i-1} - \lambda_i \geq \Delta$. For each $r^2 \in [\lambda_i, \lambda_{i-1}]$ and for all $\delta < \frac{3\Delta}{(\lambda_1 - \lambda_d)\epsilon d}$ one has*

1. *If $r^2 > (\lambda_i + \lambda_{i-1})/2$, then*
$$V(r) \leq e^{\epsilon} V(\sqrt{r^2 + \delta(\lambda_{i-1} - r^2)}).$$
2. *If $(r_{min})^2 < r^2 < (\lambda_i + \lambda_{i-1})/2$, then*
$$V(r) \leq e^{\epsilon} V(\sqrt{r^2 + \delta(r^2 - \lambda_i)}).$$

Furthermore, one has for all $r \in [r_{min}, \lambda_1 - \lambda_d]$

$$V(r) \leq e^{\epsilon} V(\sqrt{r^2 + \Delta^*}),$$

where $\Delta^* = 1/\text{poly}(d, \lambda_1 - \lambda_d)$.

The proof is deferred to the full version of the paper due to space constraints.

6.2 Approximating $V_A(r)$ In what follows we will approximate volumes $V(r)$ by expressing them as integrals of lower dimensional volumes. In order to estimate the integrals to a multiplicative factor in polynomial time, we will use the concept of an (ϵ, δ)-mesh. An (ϵ, δ)-mesh is simply a partitioning of the interval that we will need to integrate over into subintervals such that the variation of the integrand inside each subinterval is small in a multiplicative sense. The bounds from Lemma 6.2 will be helpful in constructing such meshes for the functions $V(r)$. Formally,

DEFINITION 6.1. *Let $I = [a, b] \subset \mathbb{R}^+$ be an interval. A subset $\mathbf{M} = \{m_1, m_2, \ldots, m_n\} \subset I$ with $a = m_0 \leq m_1 \leq m_2 \leq \ldots \leq m_n = b$ is an (ϵ, δ)-mesh for a function $f : I \to \mathbb{R}^+$ if*

1. *for all $1 < i+1 \in [n]$ one has for all $x \in [m_i, m_{i+1}]$*
$$f(x) \geq e^{-\epsilon} \min\{f(m_i), f(m_{i+1})\}$$
and
$$f(x) \leq e^{\epsilon} \max\{f(m_i), f(m_{i+1})\},$$
2. *$m_1 - m_0 \leq \delta$.*

We refer to the interval $[m_0, m_1]$ as the special *interval.*

It follows from volume estimates that

LEMMA 6.3. *For any $\epsilon > 0$ and $\delta > 0$ there exists an (ϵ, δ)-mesh for $V_k(\eta)$ with $|\mathbf{M}| = \tilde{O}(d(\lambda_1 - \lambda_d)/(\epsilon\Delta \log(1/\delta))$, where $\Delta > 0$ is a lower bound on eigenvalue separation.*

Proof. For each $i = 1, \ldots, d$ let $\eta_k^i = \sqrt{\lambda_i + (1 + \frac{3\epsilon\Delta}{(\lambda_1 - \lambda_d)\cdot d})^k}$, $k = 0, \ldots, O(\frac{(\lambda_1 - \lambda_d)\cdot d \log(1/\delta)}{\epsilon\Delta})$. Here Δ is the lower bound on eigenvalue separation. It follows from Lemma 6.2 that this set of points yields an (ϵ, δ)-mesh. ∎

Later we will use the concept of an (ϵ, δ)-mesh in the following situation. Suppose that we need to evaluate

$$\int_a^b f(g(x))h(x)dx$$

for a function $f(x)$ that is expensive to evaluate (this will be the volume of an appropriate set $S_A(r)$) an increasing function $g(x)$ and a 'well-behaved' function $h(x)$. In order to evaluate this integral to a $1 \pm \epsilon$ multiplicative factor, it is sufficient to have access to values of $f(x)$ for x belonging to an (ϵ, δ)-mesh of the range of $g(x)$, i.e. $g([a, b])$, as well as a sufficiently fine mesh for $h(x)$.

Refining a mesh. Let $\mathbf{M}_f$ denote an (ϵ, δ)-mesh for the function f on $[g(a), g(b)]$, and let $\mathbf{M}_h$ denote an (ϵ, δ)-mesh for h on $[a, b]$. We refer to the mesh $\mathbf{M} := g^{-1}(\mathbf{M}_f) \cup \mathbf{M}_h$ as the *refinement* of $g^{-1}(\mathbf{M}_f)$ by $\mathbf{M}_h$. It can be readily verified that $\mathbf{M}$ is a $(2\epsilon, \max\{g^{-1}(\delta), \delta\})$ mesh for $f(g(x))h(x)$.

6.3 Recursive expression for volumes We now show how to calculate a sufficiently good approximation to $V(r)$. For each $k = 1, \ldots, d$ let A_k denote the restriction of A to the k-dimensional subspace spanned by the first k eigenvectors. For $k = 1, \ldots, d$ let

$$S_k(r) = \left\{ x \in \mathbb{S}^{k-1} : \sum_{i=2}^{k} (\lambda_1 - \lambda_i) x_i^2 \leq r^2 \right\},$$
$$V_k(r) = \mu_{k-1}(S_k(r)).$$

Thus, the sets $S_k(r)$ are the level sets of the Rayleigh quotient for matrices A_k. For a radius $R > 0$ we will also use the notation

$$S_k(R, r) = \left\{ x \in R \cdot \mathbb{S}^{k-1} : \sum_{i=2}^{k} (\lambda_1 - \lambda_i) x_i^2 \leq r^2 \right\},$$
$$V_k(R, r) = \mu_{k-1}(S_k(R, r)),$$

i.e. the same sets on a sphere of radius R. Note that $V_k(R,r) = R^{k-1} \cdot V_k(1, r/R)$. Then for each k the value of $V_k(r)$ is given by

(6.17)
$$\begin{aligned}
&\int_{-A(r)}^{A(r)} V_{k-1}\left(\sqrt{1-x^2}, \sqrt{r^2-(\lambda_1-\lambda_k)x^2}\right) \frac{1}{\sqrt{1-x^2}} dx \\
&= \int_{-A(r)}^{A(r)} V_{k-1}\left(\sqrt{\frac{r^2-(\lambda_1-\lambda_k)x^2}{1-x^2}}\right) \left(1-x^2\right)^{k-5/2} dx \\
&= 2\int_0^{A(r)} V_{k-1}\left(g_{r,k}(x)\right) p_{k-1}(x) dx,
\end{aligned}$$

where $g_{r,k}(x) := \sqrt{\frac{r^2-(\lambda_1-\lambda_k)x^2}{1-x^2}}$, $p_k(x) := (1-x^2)^{k-3/2}$ and $A(r) = r/\sqrt{\lambda_1-\lambda_k}$. In what follows we will often omit the second subscript k in $g_{r,k}(x)$ when k is clear from context. We will be interested in the case $r^2 - (\lambda_1 - \lambda_k) < 0$ since otherwise $V_k(r) = \text{vol}\,(\mathbb{S}^{k-1})$.

We will need the following simple

CLAIM 6.2. *$g_r(x)$ is a decreasing function.*

Proof.

$$\begin{aligned}
\sqrt{\frac{r^2-(\lambda_1-\lambda_k)x^2}{1-x^2}} &= r\sqrt{\frac{1-(x\sqrt{\lambda_1-\lambda_k}/r)^2}{1-x^2}} \\
&= r\sqrt{\frac{1-z^2}{1-\gamma z^2}},
\end{aligned}$$

where $\gamma \in (0,1]$ and $z \in [0,1]$. Thus,

$$\begin{aligned}
\sqrt{\frac{1-z^2}{1-\gamma z^2}} &= \sqrt{\frac{1-\gamma^{-1}+\gamma^{-1}-z^2}{1-\gamma z^2}} \\
&= \sqrt{\gamma^{-1} + \frac{1-\gamma^{-1}}{1-\gamma z^2}},
\end{aligned}$$

which is a decreasing function of z. ∎

The proof of the following claim is immediate

CLAIM 6.3. *There exists an (ϵ,δ)-mesh for $p_k(x)$ on $O((k/\epsilon)\log(1/\delta))$ points.*

Proof. Let $m_i = (1-\epsilon/k)^i$ for $i = 0, \ldots, O\left(\frac{k}{\epsilon}\log(\frac{1}{\delta})\right)$. ∎

6.4 Estimating volumes We start by introducing notation. Let $r_{min} < r_{conv} = 1/\text{poly}(d, \lambda_1 - \lambda_d)$ denote a lower bound on the radius of the sets $S_A(r)$ that we will handle recursively. We will handle sets $S_A(r)$ for $r \geq r_{min}$ in a recursive fashion, and sample directly from the appropriate distribution if $r < r_{min}$. The latter will be possible since $S_A(r)$ will be convex for $r < r_{conv} = 1/\text{poly}(d, \lambda_1 - \lambda_d)$ (see Section 6.5 below).

For each k let $\mathbf{M}(k)$ denote an (ϵ,δ)-mesh for $V_k(r)$, where $r \in [0, \sqrt{\lambda_1 - \lambda_d}]$, whose existence is guaranteed by Lemma 6.3. We will calculate estimates $\hat{V}_k(r)$ for $r \in \mathbf{M}(k)$ recursively. The calculation in dimension k will rely on the calculation for the smaller dimension. We will use the relation (6.17)

$$V_k(r) = 2\int_0^{r/\sqrt{\lambda_1-\lambda_d}} V_{k-1}\left(g_r(x)\right) p_{k-1}(x) dx. \tag{6.18}$$

For each $k = 2, \ldots, d$ and each $r_j \in \mathbf{M}(k)$ we find (using binary search) $\hat{R}$ such that

$$\begin{aligned}
&e^{-\epsilon k} V_k(r_j) - 2k \cdot \text{vol}\,\left(S_k(r_j) \cap H_k(z^*)\right) \\
&\leq \hat{R}^{k-1} \\
&\leq e^{\epsilon k} V_k(r_j) + 2k \cdot \text{vol}\,\left(S_k(r_j) \cap H_k(z^*)\right)
\end{aligned} \tag{6.19}$$

if such $\hat{R} \geq r_{min}$ exists and will approximate $V_k(r_j)$ by 0 otherwise.

DEFINITION 6.4. *We write $\hat{R}(j,k)$ to denote the estimated radius $\hat{R}$ in (6.19) for point r_j in the mesh $\mathbf{M}(k)$. Also, we abuse notation somewhat by writing $\hat{R}(m,k)$ to denote $\hat{R}(j,k)$ for $m \in [m_{j-1}, m_j]$.*

Estimation procedure. We will approximate (6.18) as follows. For each $r \in \mathbf{M}(k)$ let $\mathbf{M}_r$ be a refinement of the mesh for $p_{k-1}(x)$ guaranteed by Claim 6.3 by $g_r^{-1}(\mathbf{M}(k))$. We then let R^* be the largest $R \geq r_{min}$ such that

$$\begin{aligned}
q(R) := \frac{2}{R} \cdot \sum_{i=1}^{|\mathbf{M}_r|} (m_i - m_{i-1}) \left(\frac{\hat{R}(g_r(m_i), k-1)}{R}\right)^{k-2} \\
\cdot\, p_{k-1}(m_i) \leq 1.
\end{aligned} \tag{6.20}$$

If $R^* > r_{min}$, we set $\hat{R}(j,k) := R^*$, Otherwise if $q(R^*) < 1/2$, we set $\hat{R}(j,k) := 0$, else $\hat{R}(j,k) := R^*(q(R^*))^{1/k}$.

Note that this estimation procedure can be used directly to obtain a polynomial time algorithm for approximating $V_k(r)$ via a dynamic program, i.e. by first computing and storing the value of $V_k(r)$ for small k and r belonging to the appropriate mesh.

Before giving the error analysis we prove some useful lemmas.

CLAIM 6.5. *For all $r \geq r_{min}$ one has*

$$\int_0^{2\Delta^*} V_{k-1}\left(g_r(x)\right) p_{k-1}(x) dx \geq V_{k-1}(r)/\textit{poly}(d, \lambda_1 - \lambda_d).$$

Proof. By Lemma 6.2 one has for all $r \geq r_{min}$ that $V(r+\Delta^*) \leq e^{\epsilon}V(r)$. Also, we have $g_r(x) = \sqrt{\frac{r^2-(\lambda_1-\lambda_k)x^2}{1-x^2}} \geq \sqrt{r^2-(\lambda_1-\lambda_k)x^2} = r\sqrt{1-(\lambda_1-\lambda_k)x^2/r^2} \approx r-(\lambda_1-\lambda_k)x^2/2r$.

Thus, for all $x \leq \sqrt{2r\Delta^*/(\lambda_1-\lambda_k)}$ and in particular for all $x \leq \Delta^*$ one has $V(g_r(x)) \geq e^{\epsilon}V(r)$, which yields the result. ∎

Choosing r_{min}. We now choose the value of r_{min}. For $\epsilon \in (0,1)$ and a vector $v \in \mathbb{S}^{d-1}$ we refer to the set $\{u \in \mathbb{S}^{d-1} : |\langle u,v\rangle| \geq \epsilon\}$ as the ϵ-cap around v, denoted by $C_\epsilon(v)$. By Fact 3.1 one has that vol $(C_{\sqrt{1-(z^*)^2}}(e_1)) \geq \frac{1}{2}(z^*/2)^{d-1}$, where $e_1 = (1,0,\ldots,0)$. We now let $r_{min} = z^*/4$ and obtain

CLAIM 6.6. *If $V_k(r) \leq (r_{min})^{k-1}$, then $S_k(r) \subset H(z^*)$.*

Finally, the following claim is immediate from Claim 6.2

CLAIM 6.7. *The function vol $(S_{k-1}(g_r(x)) \cap H(z^*))p_{k-1}(x)$ is non-increasing in x.*

Proof. By Claim 6.2 we have $S_{k-1}(g_r(x)) \cap H(z^*) \subseteq S_{k-1}(g_r(y)) \cap H(z^*)$ if $x \geq y$. Since $p_{k-1}(x)$ is non-increasing, this completes the proof. ∎

Our model of roundoff errors is fixed precision arithmetic with $O(\log \text{poly}(d, \lambda_1-\lambda_d))$ bits, i.e. lower order bits are lost in arithmetic operations. Since all arithmetic operations are performed on non-negative numbers, we only obtain *underestimates* of volumes due to loss of bits in the arithmetic[2]. We prove

LEMMA 6.4. *Suppose that the estimation procedure in (6.20) uses (ϵ,δ)-meshes with sufficiently small $\delta = 1/poly(d,\lambda_1-\lambda_d) < z^*$. Then the procedure outputs an approximation $\hat{V}_k(r)$ to $V_k(r)$ such that*

$$e^{-\epsilon k}V_k(r) - 2k \cdot vol\,(S_k(r) \cap H_k(z^*)) \leq \hat{V}_k(r) \leq e^{\epsilon k}V_k(r)$$

for all $r \geq 0$. In particular, we have $e^{-2\epsilon k}V(r) \leq \hat{V}(r) \leq e^{2\epsilon k}V(r)$ for $r \geq r_{min}$.

Proof. It will be convenient to use the notation $\Delta_i^m = m_i - m_{i-1}$. The proof is by induction on k.

Base:$k=2$ Since the values of $V_k(r)$ are calculated analytically, the conclusion of the lemma follows with ϵ approximately equal to machine precision.

[2]While we prefer to state Lemma 6.4 in this form, it can be easily seen that similar bounds with vol $(S_k(r) \cap H(z^*))$ appearing in both upper and lower bounds follow by the same argument when rounding up can also occur.

Inductive step:$(k-1) \to k$ Consider the approximation for $V_k(r)$ for some $r \geq 0$. First note that one has $(\hat{R}(i,k)/R)^{k-1} \leq \text{poly}(d, \lambda_1-\lambda_d)$ by Claim 6.5. Let $\mathbf{M}_r$ denote a mesh for $p_{k-1}(\cdot)$ refined by $g_r^{-1}(\mathbf{M}(k-1))$.

Recall that the algorithm uses the approximation (6.20)

(6.21)
$$\hat{q}(R) := \frac{2}{R}\sum_{i=1}^{|\mathbf{M}_r|}\Delta_i^m\left(\frac{\hat{R}(g_r(m_i),k-1)}{R}\right)^{k-2} \cdot p_{k-1}(m_i),$$

Fix $i = 1,\ldots,|\mathbf{M}_r|$ and let $\mathcal{I} := [\alpha,\beta]$ denote the interval that $[m_{i-1},m_i]$ belongs to in the mesh $\mathbf{M}(k-1)$. We consider two cases, depending on whether $\mathcal{I}$ is the special interval in $\mathbf{M}(k-1)$.

Case 1. Suppose that $\mathcal{I}$ is the special interval. Then $\hat{R}(g_r(m_i),k-1) = 0$. On the other hand,

(6.22)
$$\int_{\bigcup_{\mathcal{I} \text{ is special}}\mathcal{I}} V_{k-1}(g_r(s))p_{k-1}(s)ds \leq \text{vol}\,(S_k(r) \cap H(z^*)).$$

since $\delta < z^*$ by assumption of the lemma (recall that δ is the parameter of the meshes that we use).

Case 2. Suppose that $\mathcal{I}$ is not a special interval. By the inductive hypothesis

(6.23)
$$\begin{aligned}&(\hat{R}(g_r(m_i),k))^{k-2} \geq e^{-\epsilon(k-1)}V_{k-1}(g_r(m_i))\\ &-2(k-1)\cdot \text{vol}\,(S_{k-1}(g_r(m_i)) \cap H(z^*))\\ &\text{and}\\ &(\hat{R}(g_r(m_i),k))^{k-2} \leq e^{\epsilon(k-1)}V_{k-1}(g_r(m_i)).\end{aligned}$$

By definition of an (ϵ,δ)-mesh we have for all $s \in [m_{i-1},m_i]$

(6.24)
$$\begin{aligned}&e^{-\epsilon}V_{k-1}(g_r(m_{i-1}))p_{k-1}(m_{i-1})\\ &\leq V_{k-1}(g_r(s))p_{k-1}(s) \leq\\ &e^{\epsilon}V_{k-1}(g_r(m_i))p_{k-1}(m_i).\end{aligned}$$

Hence,

$$\begin{aligned}&e^{-\epsilon}\Delta_i^m V_{k-1}(g_r(m_{i-1}))p_{k-1}(m_{i-1})\\ &\leq \int_{m_{i-1}}^{m_i} V_{k-1}(g_r(s))p_{k-1}(s)ds\\ &\leq e^{\epsilon}\Delta_i^m V_{k-1}(g_r(m_{i-1}))p_{k-1}(m_{i-1}).\end{aligned}$$

We have

(6.25)
$$\begin{aligned}
&\sum_{i=1}^{|\mathbf{M}_r|} \Delta_i^m(\hat{R}(g_r(m_i),k-1))^{k-2}p_{k-1}(m_i)\\
&\geq \sum_{i=1}^{|\mathbf{M}_r|} \Delta_i^m(e^{-\epsilon k}V_{k-1}(g_r(m_i))\\
&-2(k-1)\text{vol }(S_{k-1}(g_r(m_i))\cap H(z^*)))p_{k-1}(m_i)\\
&\geq e^{-\epsilon k}\left(\int_0^{r/\sqrt{\lambda_1-\lambda_k}} V_{k-1}(g_r(x))p_{k-1}(x)dx\right) - L,
\end{aligned}$$

where the value of L is given by

$$2(k-1)\sum_{i=1}^{|\mathbf{M}_r|}\Delta_i^m\text{vol }(S_{k-1}(g_r(m_i))\cap H(z^*)))p_{k-1}(m_i).$$

We now bound L. By Claim 6.2 the function vol $(S_{k-1}(g_r(x))\cap H(z^*))p_{k-1}(x)$ in non-increasing in x, and hence

(6.26)
$$\begin{aligned}
&\sum_{i=1}^{|\mathbf{M}(k)|}\Delta_i^m\text{vol }(S_{k-1}(g_r(m_i))\cap H(z^*))p_{k-1}(m_i)\\
&\leq \int_0^{z/\sqrt{\lambda_1-\lambda_d}}\text{vol }(S_{k-1}(g_r(x))\cap H(z^*))p_{k-1}(x)dx\\
&\leq \text{vol }(S_k(r)\cap H(z^*))
\end{aligned}$$

since the lhs uses the value of the function at the right endpoint of each interval. Thus,

$$L\leq 2(k-1)\text{vol }(S_k(r)\cap H(z^*)),$$

showing that

$$\begin{aligned}
&\sum_{i=1}^{|\mathbf{M}_r|} \Delta_i^m(\hat{R}(g_r(m_i),k-1))^{k-2}p_{k-1}(m_i)\\
&\geq e^{-\epsilon k}\left(\int_0^{r/\sqrt{\lambda_1-\lambda_k}} V_{k-1}(g_r(x))p_{k-1}(x)dx\right)\\
&-2(k-1)\text{vol }(S_k(r)\cap H(z^*)).
\end{aligned}$$

A similar calculation shows that

$$\begin{aligned}
&\sum_{i=1}^{|\mathbf{M}_r|} \Delta_i^m(\hat{R}(g_r(m_i),k-1))^{k-2}p_{k-1}(m_i)\\
&\leq e^{\epsilon k}\left(\int_0^{r/\sqrt{\lambda_1-\lambda_k}} V_{k-1}(g_r(x))p_{k-1}(x)dx\right).
\end{aligned}$$

Finally, we bound roundoff errors that arise from a $O(\log \text{poly}(d,\lambda_1-\lambda_d))$-precision evaluation of (6.21). There are $|\mathbf{M}_r|$ intervals of total length 1, and the maximum value of the integrand in each interval is at most $V_k(r)$. Thus, $O(\delta|\mathbf{M}_r|)V_k(r)$ volume could have been lost to roundoff errors. Choosing $\delta = 1/\text{poly}(d,\lambda_1-\lambda_d)$ small enough ensures that this lost volume is at most vol $(S_k(r)\cap H(z^*))/2$ by Claim 6.5.

Let R^* denote the smallest $R\geq r_{min}$ such that $\hat{q}(R)\leq 1$. We now consider two cases:

(A) Suppose that $\hat{q}(R^*) < 1/2$, and in particular $R^*\leq r_{min}$. Putting (6.22) and (6.26) together with Claim 6.5, we get

(6.27)
$$\begin{aligned}
V_k(r) &\leq e^{\epsilon k}(\hat{q}(R^*)(R^*)^{k-1}\\
&\quad + 2(k-1)\text{vol }(S_{k-1}(r)\cap H(z^*))))\\
&\quad + \text{vol }(S_k(r)\cap H(z^*)) + V_k(r)\delta|\mathbf{M}(k)|\\
&< 2ke^{\epsilon k}\text{vol }(S_k(r)\cap H(z^*))),
\end{aligned}$$

which proves the inductive step since the estimation procedure approximates $V_k(r)$ by 0 in this case.

(B) Suppose that $\hat{q}(R^*)\geq 1/2$. In this case we have that

$$\delta|\mathbf{M}|\text{poly}(d,\lambda_1-\lambda_d)\leq \epsilon\hat{q}(R^*),$$

for a sufficiently small $\delta = O(1/\text{poly}(d,\lambda_1-\lambda_d))$, and hence setting $\hat{R}(i,k)$ to $R\cdot(\hat{q}(R^*))^{1/(k-1)}$ satisfies

$$\begin{aligned}
(\hat{R}(i,k))^k &\geq e^{-\epsilon k}V_k(r) - 2k\cdot\text{vol }(S_k(r)\cap H(z^*))\\
(\hat{R}(i,k))^k &\leq e^{\epsilon k}V_k(r)
\end{aligned}$$

as required.

■

We will later show how this recursive estimation procedure allows us to sample almost uniformly from the sets $S_A(r)$. However, before we do that, we need to remove the restriction of sufficiently large radius that is imposed in Lemma 6.4. This turns out to be quite simple, since for sufficiently small radius r the sets $S_A(r)$ are well approximated by ellipses under an appropriate projection, which we show in the next section.

6.5 Sampling for small volumes In this section we show how to sample nearly uniformly from the set $S_k(r)$ for sufficiently small $r < r_{conv} = 1/\text{poly}(d,\lambda_1-\lambda_d)$. We start by showing that the sets $S_k(r)$ are convex (with respect to geodesics on $\mathbb{S}^{d-1}$) for sufficiently small r. For all $r<\sqrt{\lambda_1-\lambda_2}$ define $f: S_A(r)\to\mathbb{R}^{d-1}$ by

$$f(x_1,\mathbf{x}) := \frac{\mathbf{x}}{x_1}.$$

Note that this is well-defined for $(x_1, \mathbf{x}) \in S_A(r), r < \sqrt{\lambda_1 - \lambda_2}$ since one necessarily has $x_1 > 0$. Let $\mathbf{z} = f((x_1, \mathbf{x}))$. One has

$$||\mathbf{z}|| = \frac{\sqrt{1 - x_1^2}}{x_1} = \sqrt{1/x_1^2 - 1},$$

so

$$x_1 = \frac{1}{\sqrt{1 + ||\mathbf{z}||^2}}.$$

Thus,

$$f^{-1}(\mathbf{z}) = \frac{1}{\sqrt{1 + ||\mathbf{z}||^2}}(1, \mathbf{z}).$$

Thus, $\mathbf{z} \in f(S_A(r))$ iff

$$\sum_{i=2}^{d} (\lambda_1 - \lambda_i) z_i^2 \leq r^2(1 + ||z||^2), \tag{6.28}$$

i.e.

$$\mathbf{z}^T \left(\text{diag}(\lambda_1 - \lambda_2, \ldots, \lambda_1 - \lambda_d) - I \cdot r^2\right) \mathbf{z} \leq r^2, \tag{6.29}$$

which is a convex set whenever $r^2 \leq \lambda_1 - \lambda_2$. Indeed, consider the 3-dimensional plane spanned by $0, \mathbf{z}_1, \mathbf{z}_2$. The line segment connecting $\mathbf{z}_1$ and $\mathbf{z}_2$ can be projected onto the sphere to form a geodesic, implying convexity of $f^{-1}(S_A(\eta))$. For future reference we let

(6.30)
$$E(r) = \{\mathbf{z} \in \mathbb{R}^{d-1} : \mathbf{z}^T \left(\text{diag}(\lambda_1 - \lambda_2, \ldots, \lambda_1 - \lambda_d) - I \cdot r^2\right) \mathbf{z} \leq r^2\}$$

We have proved

LEMMA 6.5. *For a symmetric matrix A for all $r \leq \sqrt{\lambda_1 - \lambda_2}$ the sets $S_A(r)$ are convex.*

It will be convenient to use

DEFINITION 6.8. *A distribution with pdf $p(\mathbf{x}), \mathbf{x} \in \mathbb{S}^k$ on a measurable subset $A \subseteq \mathbb{S}^k$ is e^ϵ-uniform if $p(\mathbf{x}) \in e^{\pm\epsilon}/vol\ (A)$ for all $\mathbf{x} \in A$.*

A more careful analysis of the mapping that we just defined reveals that the sets $S_A(r)$ are also quite flat, which yields a simple algorithm for sampling and volume estimation:

LEMMA 6.6. *For sufficiently small $\eta > 0$*

1. *A $e^{O(\eta)}$-uniform distribution on $S_A(r)$ for $r \leq r_{conv} = 1/poly(d, \lambda_1 - \lambda_d) := \sqrt{(\lambda_1 - \lambda_2)\eta/d}$ can be obtained by sampling uniformly from an ellipse.*
2. *for $r < r_{conv}$ one has $\frac{vol_{\ d-1} E(r)}{vol\ (S_A(r))} \in e^{\pm O(\eta)}$.*

Proof. We only give an outline of the proof, deferring the details to the full version. Let $r_{conv} = \sqrt{(\lambda_1 - \lambda_2)\eta/d}$, where $\eta > 0$ is a precision parameter. For each $x \in S_A(r)$ one has $\sum_{i=2}^{d} (\lambda_1 - \lambda_i) x_i^2 \leq (\lambda_1 - \lambda_2)\eta/d$. Thus, one has $\sum_{i=2}^{d} x_i^2 \leq \eta/d$, i.e. $x_1 \geq \sqrt{1 - \eta/d} \geq 1 - 2\eta/d$ for sufficiently small $\eta < 1$.

Parameterizing the surface of the sphere using coordinates $(x_2, x_3, \ldots, x_d)$, we get that the projection $(x_1, \mathbf{x}) \to \mathbf{x}/x_1$ changes volumes by at most a $e^{\pm O(\eta)}$ factor, implying that the obtained distribution is within this multiplicative factor of uniform. The approximation for volume follows immediately. ∎

6.6 Sampling from $V_{k-1}(g_r(x))p_{k-1}(x)$ for small x

In this section we show how to sample x_k from the distribution

$$q(x) \sim V_{k-1}(g_r(x))p_{k-1}(x), x_k \in [L, r/\sqrt{\lambda_1 - \lambda_k}]$$

when $r < \sqrt{\lambda_1 - \lambda_k} - \Delta_r$ and $g_r(L) \leq r^*$. Here $\Delta_r \geq 1/\text{poly}(d, \lambda_1 - \lambda_d)$ is the parameter using to shift the radius r away from eigenvalues in Algorithm 5 below. Intuitively, this is the regime where V_{k-1} behaves essentially like a $(k-2)$-dimensional ball, which we exploit to get a simple sampling algorithm. We will crucially use the assumption $r^* < \frac{\epsilon \Delta_r}{4d^2\sqrt{\lambda_1 - \lambda_d}}$.

Recall that $g_r(x) = \sqrt{\frac{r^2 - (\lambda_1 - \lambda_k)x^2}{1 - x^2}}$. We first get a convenient and sufficiently accurate approximation to the function $g_r(x)$. Let

$$x^* := r/\sqrt{\lambda_1 - \lambda_k} < 1 - \Delta_r/\sqrt{\lambda_1 - \lambda_d}. \tag{6.31}$$

Write $x = \sqrt{(x^*)^2 - \xi^2}$, so that

$$\begin{aligned} g_r(x) &= \sqrt{\frac{r^2 - (\lambda_1 - \lambda_k)x^2}{1 - x^2}} \\ &= (\lambda_1 - \lambda_k)^{1/2} \cdot \sqrt{\frac{\xi^2}{1 - (x^*)^2 + \xi^2}} \end{aligned} \tag{6.32}$$

Since $g_r(L) \leq r^*$, we have $\xi \leq r^*/\sqrt{\lambda_1 - \lambda_d} \leq \epsilon(1 - (x^*)^2)/d^2$ by (6.32) together with (6.31). Furthermore, we have by (6.32) that

$$\begin{aligned} g_r(x) &= (\lambda_1 - \lambda_k)^{1/2} \cdot \sqrt{\frac{\xi^2}{1 - (x^*)^2 + \xi^2}} \\ &\in \sqrt{(1 \pm \epsilon/d^2)\frac{\lambda_1 - \lambda_k}{1 - (x^*)^2}} \cdot \xi. \end{aligned}$$

On the other hand, since $g_r(x) \leq r \leq r_{conv}$, we have by Lemma 6.6

$$V_{k-1}(g_r(x)) \in (1 \pm \epsilon/d) \cdot c_k(g_r(x))^{k-2}, \tag{6.33}$$

where c_k is a constant that depends only on the dimension k.

Since $\xi = \sqrt{(x^*)^2 - x^2}$,

(6.34)
$$V_{k-1}(g_r(x))p_{k-1}(x) \sim \xi^{k-2} \cdot (1-(x^*)^2-\xi^2)^{(k-3)/2} \in (1 \pm \epsilon/d) \cdot \xi^{k-2}.$$

We need to sample ξ from distribution (6.34) subject to $\sqrt{(x^*)^2 - \xi^2} \geq L$, i.e. $\xi \leq \sqrt{L^2 - (x^*)^2}$. Sampling from the distribution with pdf

$$f(\xi) = \frac{\xi^{k-2}}{\int_0^{\sqrt{L^2-(x^*)^2}} s^{k-2} ds}$$

can be easily done in $O(k)$ time using acceptance/rejection from the uniform distribution. We refer to this sampling procedure as

Algorithm 4 SAMPLE-CORNER(k, r, L)

1: Sample $x_k \geq L$ from the distribution in (6.34).
2: Sample a uniformly random point $(x_1, \ldots, x_{k-1})$ from $E(g_r(x_k))$.

6.7 Uniform sampling We now turn to the problem of uniform sampling for general r. As before, let $\mathbf{M}(k)$ denote the ϵ-mesh for $V_k(r)$. As before, for fixed k let $\mathbf{M}_r$ denote a mesh for the function $V_{k-1}(g_r(x))p_{k-1}(x)$.

Outline of the sampling process. We now give the algorithm for sampling from $S_A(r)$ using our volume estimates. The procedure is very simple: in order to sample $\mathbf{x} = (x_1, \ldots, x_d)$ uniformly from $S_A(r)$ it is sufficient to sample the $x_d \sim V_{d-1}(g_r(x_d))p_{d-1}(x_d)$, and then recursively sample $\bar{x} = (x_1, \ldots, x_{d-1})$ from $S_{d-1}(g_r(x_d))$. Since we have obtained multiplicative approximations of the respective volumes above, the multiplicative error in the distribution that we obtain will be at most raised to power d, i.e. the number of steps in the recursion, which we can handle by choosing our precision appropriately. One issue that arises is that this procedure may require sampling from $S_k(r)$ with r smaller than r_{min}, but we do not have estimates for such small volumes. However, the convexity of the sets $S_k(r)$ for such small r enables us to use the procedure SAMPLE-CORNER described in the previous section.

Thus, our sampling procedure consists of two steps. First, we invoke a recursive sampling procedure that uses volume estimates (6.20) to produce (possibly all) coordinates of the output point. This procedure will terminate without producing all coordinates only if sampling from a set of very small radius is required. If that happens, we sample the remaining coordinates using the procedure SAMPLE-CORNER that we described in the previous section. The procedure SAMPLE-REC (Algorithm 5) does exactly that, and outputs two vectors $\mathbf{x}$ and $\mathbf{z}$, which correspond to the part of the input sampled by the first and last method respectively (note that $\mathbf{z}$ may be empty). We then convert $\mathbf{x}$ and $\mathbf{z}$ into a point in $\mathbb{S}^{d-1}$ in the procedure SAMPLE-OUTER (Algorithm 6).

This high level overview overlooks two problems. First, we would like to use arithmetic with $\log \text{poly}(d, \lambda_1 - \lambda_d)$ bits, which is not entirely straightforward since volumes are in general exponentially small in d. However, we show that this can be easily overcome by splitting the set of possible values for x into a nested collection of intervals so that each next intervals contains half of the probability mass of its enclosing interval. A more delicate problem is the problem of numerical stability. Note that roundoff errors in the radius r invariably either avoid certain points in the set $S_A(r)$ or output points from outside of $S_A(r)$ with positive probability, which is clearly not differentially private. However, we will show later that the distribution that we obtain, although not strictly uniform, is still sufficiently good for sampling.

Geometric decomposition. Suppose that we need to sample uniformly from $V_k(r)$. As before, let $\mathbf{M}_r$ is a refinement of the mesh for $p_{k-1}(x)$ guaranteed by Claim 6.3 by $g_r^{-1}(\mathbf{M}(k))$. For all $i = 1, \ldots, |\mathbf{M}_r|$ let

$$\Sigma_i(R) := \frac{2}{R} \sum_{s=i}^{|\mathbf{M}_r|} (m_s - m_{s-1}) \left(\frac{\hat{R}(g(m_s), k-1)}{R} \right)^{k-2} p_{k-1}(m_s)$$

for $R \geq r_{min}$.

We now partition intervals of the mesh $\mathbf{M}_r$ into groups containing a geometrically decaying fraction of mass. Let R_1 be the smallest such that $R_1 \geq r_{conv}$ and $\Sigma_1(R_1) \geq 1$, and let i_1 be the largest such that $\Sigma_{i_1}(R_1) \geq 1/2$. Similarly, for each $j = 2, \ldots, s$ let R_j be the smallest such that $R_j \geq r_{conv}$ and $\Sigma_{i_{j-1}}(R_j) \leq 1$, and let i_j be the largest such that $\Sigma_{i_j}(R_j) \geq 1/2$. If no $R_j \geq r_{conv}$ satisfies this condition, we stop the process and let $s = j - 1$.

Note that $s = O(d \log \text{poly}(d, \lambda_1 - \lambda_d))$ since only a constant fraction of probability mass remains at each step, and the volume of a sphere of radius r_{conv} is $r_{conv}^{\Omega(d)}$. For convenience we let $i_0 = |\mathbf{M}_r|$. For each $j = 1, \ldots, s$ let $I_j = [m_{i_{j-1}}, m_{i_j}]$. The algorithm is specified more formally as Algorithm 5.

Precision in Algorithm 6. Note that multiplication in lines 3 and 5 can be done with $O(\log(d))$ size arithmetic. Indeed, let $u_k, u_{k-1}, \ldots, u_j$ denote the co-

ordinates sampled in the first $k-j+1$ steps. Then

$$\sum_{i=j}^{k} x_i^2 = \sum_{i=j}^{k} u_i^2 \prod_{l=i+1}^{k} (1-u_l^2) \geq 1 - \prod_{l=j}^{k} (1-u_l^2).$$

Thus, for any $u_1, \ldots, u_{j-1}$ one has

$$\sum_{i=1}^{d} (\lambda_1 - \lambda_i) x_i^2 \geq \left(1 - \prod_{l=j}^{k} (1-u_l^2)\right) (\lambda_1 - \lambda_j).$$

Hence, if $\prod_{l=j}^{k}(1-u_l^2) < \Delta/(\lambda_1 - \lambda_j)$, where $\Delta = \Omega(1/\mathrm{poly}(d))$ is a lower bound on eigenvalue separation, then one necessarily has $r^2 \geq \lambda_1 - \lambda_{j-1}$, and hence the next step in Algorithm 5 samples uniformly from a sphere, i.e. j is the last index that belongs to $\mathbf{x}$.

REMARK 6.9. *Note that in Algorithm 5, we always ensure that the desired radius r is away from any eigenvalue by at least $\Delta_r = 1/poly(d, \lambda_1 - \lambda_d)$[3]. Thus, we can assume that $g_r(x) = 0$ for $x = x^* < 1 - 1/poly(d, \lambda_1 - \lambda_d)$, which we use in the procedure SAMPLE-CORNER.*

Note that at this point we do not prove that Algorithm 5 produces an ϵ-approximation to the uniform distribution over the set $S_A(r)$. In fact, this is not the case due to the change in the radius in line 2. However, we show next that using this algorithm as a subroutine in sampling from the exponential distribution yields a ϵ-approximation, as required.

Since Algorithm 5 does not output a point on a sphere, we now specify an outer sampling procedure that corrects that. This procedure is given by Algorithm 6.

6.8 Sampling from the modified Γ distribution Recall that in order to sample from the exponential distribution, it is sufficient to sample $t \propto e^{-\epsilon t} V_A(\sqrt{t})$, and then sample uniformly from $S_A(t)$.

Equipped with multiplicative approximations of $V_A(\sqrt{t})$, this sampling can be implemented similarly to the sampling of coordinates in Algorithm 5. One difference is that the function $e^{-\epsilon t} V_A(\sqrt{t})$ is no longer monotone in t, but this can be easily handled by bucketing intervals of the mesh by the values of the function. Alternatively, a simple calculation reveals that evaluating $V_A(\sqrt{t})$ over a regular grid with step size $O(\epsilon/d)$ will not introduce more than a $e^{O(\epsilon)}$ multiplicative distortion to the output probability distribution over $x \in \mathbb{S}^{d-1}$.

[3]Note that this perturbation not only makes the sampling non-uniform, but in fact makes the produced density function 0 at points where it should be positive under uniform sampling. While this seems to contradict ϵ-differential privacy, we will show that the resulting distribution is still ϵ-close to exponential in the numerical stability section below.

Algorithm 5 Uniform sampling from $V_k(r)$:SAMPLE-REC(k, r)

1: If $r < r_{conv}$ then $\mathbf{z} \leftarrow$ SAMPLE-CORNER(k, r, 0), return (z, x).
2: $i \leftarrow k+1$
3: **for** $k = d$ downto 2 **do**
4: If $|r - \sqrt{\lambda_1 - \lambda_j}| < \Delta_r$ for some j, set $r \leftarrow \sqrt{\lambda_1 - \lambda_j} + \Delta_r$.
5: If $r > \sqrt{\lambda_1 - \lambda_k}$ return a random vector in $\mathbb{S}^{k-1}$.
6: **for** $j = 1$ to s **do**
7: **if Ber**$(0/1, \Sigma_{i_j}(R_j)/\Sigma_{i_{j-1}}(R_j)) = 0$ **then**
8: Pick $x_k \sim \hat{V}_{k-1}(g_r(x)) p_{k-1}(x)$ from I_j.
9: $r \leftarrow g_r(x_k)$
10: $i \leftarrow i-1$
11: **break**
12: **else**
13: **if** $j = s$ **then**
14: $\mathbf{z} \leftarrow$ SAMPLE-CORNER(k, r, x_k).
15: **end if**
16: **break**
17: **end if**
18: **end for**
19: **end for**
20: **return** $(\mathbf{z}, \mathbf{x})$

6.9 Numerical stability In this section we show that our approximation to the exponential distribution produces a ϵ-differentially private distribution even in the presence of roundoff errors. As mentioned before, the procedure in Algorithm 5 does not produce the uniform distribution over the appropriate level set due to the shift in the value of r around eigenvalues of the matrix. We now quantify the distribution that is produced, and then show that this approximate distribution is sufficient to ensure ϵ-differential privacy. We assume that Δ_r is chosen to be small enough so that

$$V_k(r + d \cdot \Delta_r) \in (1 \pm \epsilon/d) V_k(r) \tag{6.35}$$

for all $r \geq r_{conv}$ and $k = 2, \ldots, d$, which is possible by the bounds in Lemma 6.2.

Algorithm 6 Outer sampling step: SAMPLE-OUTER(k, r)

1: $(\mathbf{z}, \mathbf{u}) \leftarrow$ SAMPLE-REC(k, r)
2: **for** $j = 1$ to **length**$(\mathbf{u})$ **do**
3: $x_j \leftarrow u_j \cdot \prod_{l=j+1}^{k} \sqrt{1-u_l^2}$.
4: **end for**
5: $\mathbf{z} \leftarrow \mathbf{z} \cdot \prod_{l=1}^{k} \sqrt{1-u_l^2}$. **return** $(\mathbf{z}, \mathbf{x})$

We will use the notation $\mathrm{d}\mu_{k-1}(r)$ to denote the uniform $k-1$ dimensional measure on $r \cdot \mathbb{S}^k$, a sphere

of radius r in $\mathbb{R}^k$. We will denote $\mu_k := \mu_k(1)$.

LEMMA 6.7. *Let $p(x,r), x \in \mathbb{S}^{d-1}$ denote the probability density function produced by Algorithm 5 when invoked to sample uniformly from $S_A(r)$. Then*

1. *$p(\mathbf{x},r) = 0$ for all $\mathbf{x} \notin S_A(r+\Delta_r)$;*
2. *$p(\mathbf{x},r) \geq e^{-O(\epsilon d^2 \log(1/r_{min}))}\, \mathrm{d}\mu_{k-1}/V_A(r)$ for all $\mathbf{x} \in S_A(r)$;*
3. *$p(\mathbf{x},r) \leq e^{O(\epsilon d^2 \log(1/r_{min}))}\, \mathrm{d}\mu_{k-1}/V_A(r)$ for all $\mathbf{x} \in \mathbb{S}^{d-1}$.*

Proof. The proof is by induction on the variable k in Algorithm 5. We prove that for any $k = 2, \ldots, d$ the k last iterations of the main loop in Algorithm 5 output a point $\mathbf{x} \in \mathbb{S}^{k-1}$ with distribution $p^k(\mathbf{x},r)$ such that

1. $p^k(\mathbf{x},r) = 0$ for all $\mathbf{x} \notin S_k(r + k \cdot \Delta_r)$;
2. $p^k(\mathbf{x},r) \geq e^{-O(\epsilon k d \log(1/r_{min}))}\, \mathrm{d}\mu_{k-1}/V_k(r)$ for all $\mathbf{x} \in S_k(r)$;
3. $p^k(\mathbf{x},r) \leq e^{O(\epsilon k d \log(1/r_{min}))}\, \mathrm{d}\mu_{k-1}/V_k(r)$ for all $\mathbf{x} \in \mathbb{S}^{d-1}$,

The first property follows immediately, since at most Δ_r is added to r at each iteration.

Base:$k = 2$ The conditions are satisfied with Δ_r equal to machine precision.

Inductive step: $k \to k+1$ Write a point $\mathbf{x} \in \mathbb{S}^k$ as $\mathbf{x} = (\mathbf{x}_0, x_k)$, where $\mathbf{x}_0 \in \sqrt{1-x_k^2} \cdot \mathbb{S}^{k-2}$. By the inductive hypothesis $p^{k-1}(x,r) = 0$ if $\mathbf{x}_0 \notin S_{k-1}(g_r(x_k)+(k-1)\Delta_r)$ and belongs to the interval

$$\left[\frac{e^{-O(\epsilon(k-1)d\log(1/r_{min}))}}{V_{k-1}(g_r(x_k))}, \frac{e^{O(\epsilon(k-1)d\log(1/r_{min}))}}{V_{k-1}(g_r(x_k))}\right]$$

otherwise.

Thus, the probability that $x_k \in [a, a+da]$ is output in Algorithm 5 is between $e^{-O(\epsilon d \log(1/r_{min}))} \cdot \hat{V}_{k-1}(g_r(a))p_{k-1}(a)da$ and $e^{O(\epsilon d \log(1/r_{min}))} \cdot \hat{V}_{k-1}(g_{r+\Delta_r}(a))p_{k-1}(a)da$. The bound of $e^{O(\epsilon d \log(1/r_{min}))}$ on the multiplicative error follows from the fact that the loop in line 6 can only be executed $O(d\log(1/r_{min}))$ times, each of which accumulates a multiplicative error of at most e^{ϵ}.

Let $r' \in (r, r+\Delta_r)$ denote the modified radius. By the inductive hypothesis we have that

1. $p_{k-1}(\mathbf{x}|x_k = a) = 0$ when $\mathbf{x}_0 \notin S_{k-1}(g_{r'}+(k-1)\Delta_r)$;
2. $$p_{k-1}(\mathbf{x}|x_k = a) \geq \frac{e^{-O(\epsilon(k-1)d\log(1/r_{min}))}}{\mathrm{vol}\,(\sqrt{1-a^2}\cdot S_{k-1}(g_{r'}))}$$ for $\mathbf{x}_0 \in S_{k-1}(g_{r'})$;
3. $$p_{k-1}(\mathbf{x}|x_k = a) \leq \frac{e^{O(\epsilon(k-1)d\log(1/r_{min}))}}{\mathrm{vol}\,(\sqrt{1-a^2}\cdot S_{k-1}(g_{r'}))}$$ for all $\mathbf{x}_0$.

Integrating over all a, we get that $p_k(\mathbf{x}) = \frac{1}{V_k(r')}\int_0^1 p_{k-1}(\mathbf{x}|x_k = a)\hat{V}_{k-1}(g_{r'}(a))p_{k-1}(a)da$ belongs to the interval

$$\left[e^{-O(\epsilon k d \log(1/r_{min}))}, e^{O(\epsilon k d \log(1/r_{min}))}\right] \cdot \frac{\mathrm{d}\mu_{k-2}(\sqrt{1-x_k^2})dx_k}{V_k(r')\sqrt{1-x_k^2}} = \mathrm{d}\mu_{k-1}/V_k(r)$$

as required.

■

It now follows easily that the distribution produced by our algorithm is ϵ-differentially private:

LEMMA 6.8. *Let $\hat{p}(\mathbf{x}), \mathbf{x} \in \mathbb{S}^{d-1}$ denote the distribution produced by using Algorithm 5 to sample from the exponential distribution. Then $\hat{p}(\mathbf{x})$ is within a $e^{\pm O(\epsilon d^2 \log(1/r_{min}))}$ factor of the distribution achieved by the exponential mechanism.*

Proof. By Lemma 6.7 we have that $\mathbf{x}$ is output with probability at least

$$\int_{\mathbf{x}^T(\lambda_1 I - A)\mathbf{x}}^{\infty} e^{-\epsilon s} e^{-O(\epsilon d^2 \log(1/r_{min}))} ds \geq e^{-O(\epsilon d^2 \log(1/r_{min}))} \cdot e^{-\epsilon \mathbf{x}^T(\lambda_1 I - A)\mathbf{x}}.$$

and at most

$$\int_{\mathbf{x}^T(\lambda_1 I - A)\mathbf{x} - \Delta^*}^{\infty} e^{-\epsilon s} e^{O(\epsilon d^2 \log(1/r_{min}))} ds = e^{-O(d\Delta^* + \epsilon d^2 \log(1/r_{min}))} e^{-\epsilon \mathbf{x}^T(\lambda_1 I - A)\mathbf{x}} = e^{-O(\epsilon d^2 \log(1/r_{min}))} e^{-\epsilon \mathbf{x}^T(\lambda_1 I - A)\mathbf{x}}.$$

■

We have proved

THEOREM 6.10. *There exists a polynomial time algorithm for sampling from the distribution given by the exponential mechanism. The algorithm takes time $\tilde{O}(d^6/\epsilon)$ when invoked on a $d \times d$ matrix.*

Proof. We set $\epsilon' = \Theta(\epsilon/(d^2 \log(1/r_{min})))$ to bring the multiplicative approximation error in Lemma 6.7 to at most e^ϵ. This yields a mesh of size $\tilde{O}((\lambda_1 - \lambda_d)d^5/\epsilon)$ by Lemma 6.3. Thus, the estimation procedure (6.20) and Algorithm 5 take $\tilde{O}((\lambda_1 - \lambda_d)d^6/\epsilon)$ time. ∎

REMARK 6.11. *It is interesting to note that due to the strong notions of privacy that we use in this work, namely spectral privacy, the following preprocessing step is feasible. Let A denote the input matrix and denote the eigenvalues by* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$. *Suppose further that most eigenvalues are close to zero, e.g.* $\lambda_j \leq \delta$ *for* $j > d_{eff}$, *where* d_{eff} *is the* effective dimension. *Then modifying A be making all eigenvalues* $\lambda_j, j > d_{eff}$ *zero only loses* δ *in the privacy guarantees and simplifies geometry of A, making Algorithm 5 run in time poly*(d_{eff}).

7 Acknowledgements

The authors would like to thank Frank McSherry for useful discussions.

References

[1] P.-A. Absil, A. Edelman, and P. Koev. On the largest principal angle between random subspaces. *Linear Algebra Appl*, 2005.

[2] K. Ball. An elementary introduction to modern convex geometry. *Flavors of Geometry, MSRI Publications*, 1997.

[3] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. *FOCS*, 2012.

[4] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '05, pages 128–138, New York, NY, USA, 2005. ACM.

[5] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "You Might Also Like:" Privacy Risks of Collaborative Filtering. In *IEEE Symposium on Security and Privacy*, pages 231–246, 2011.

[6] Kamalika Chaudhuri, Anand Sarwate, and Kaushik Sinha. Near-optimal algorithms for differentially private principal components. In *Neural Information Processing Systems (NIPS)*, 2012. To appear.

[7] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[9] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Proceedings of the 9th international conference on Theory of Cryptography*, TCC'12, pages 339–356, Berlin, Heidelberg, 2012. Springer-Verlag.

[10] Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. *STOC*, pages 1255–1268, 2012.

[11] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. *FOCS*, pages 57–68, 2006.

[12] Frank McSherry and Ilya Mironov. Differentially private recommender systems: building privacy into the net. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 627–636, New York, NY, USA, 2009. ACM.

[13] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. *FOCS*, pages 94–103, 2007.

[14] R. Muirhead. *Aspects of multivariate statistical theory.* Wiley Series in Probability and Mathematical Statistics, 1982.

[15] Arvind Narayanan and Vitaly Shmatikov. Robust Deanonymization of Large Sparse Datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125, 2008.

[16] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *ACM Symposium on Theory of Computing*, pages 75–84, 2007.

[17] Alan Weinstein. Almost invariant submanifolds for compact group actions. *J. Eur. Math. Soc. (JEMS)*, 2:53–86, 2000.