Fast estimation of approximate matrix ranks using spectral densities

Shashanka Ubaru*   Yousef Saad*   Abd-Krim Seghouane†

November 29, 2016

Abstract

Many machine learning and data related applications require the knowledge of approximate ranks of large data matrices at hand. This paper presents two *computationally inexpensive* techniques to estimate the approximate ranks of such matrices. These techniques exploit approximate *spectral densities*, popular in physics, which are probability density distributions that measure the likelihood of finding eigenvalues of the matrix at a given point on the real line. Integrating the spectral density over an interval gives the eigenvalue count of the matrix in that interval. Therefore the rank can be approximated by integrating the spectral density over a carefully selected interval. Two different approaches are discussed to estimate the approximate rank, one based on Chebyshev polynomials and the other based on the Lanczos algorithm. In order to obtain the appropriate interval, it is necessary to locate a gap between the eigenvalues that correspond to noise and the relevant eigenvalues that contribute to the matrix rank. A method for locating this gap and selecting the interval of integration is proposed based on the plot of the spectral density. Numerical experiments illustrate the performance of these techniques on matrices from typical applications.

1 Introduction

Many machine learning, data analysis, scientific computations, image processing and signal processing applications involve large dimensional matrices whose relevant information lies in a low dimensional subspace. In the most common situation, the dimension of this lower dimensional subspace is unknown, but it is known that the input matrix is (or can be modeled as) the result of adding small perturbations (e.g., noise or floating point errors) to a low rank matrix. Thus, this matrix generally has full rank, but it can be well approximated by a low rank matrix. Many known techniques such as Principal Component Analysis (PCA) [36, 34], randomized low rank approximations [29, 46], low rank subspace estimations [14, 44, 69] exploit this low dimensional nature of the input matrices. An important requirement of these techniques is the knowledge of the dimension of the smaller subspace, which can be viewed as an approximate rank of the input matrix.

Low rank approximation and dimensionality reduction techniques are common in applications such as machine learning, facial and object recognition, information retrieval, signal processing, etc [68, 45, 62]. Here, the data consist of a $d \times n$ matrix X (i.e., n features and d instances), which either has low rank or can be well approximated by a low rank matrix. Basic tools used in these applications consist of finding low rank approximations of these matrices [29, 68, 5, 21]. Popular among these tools are the sampling-based methods, e.g., the randomized algorithms [29, 63]. These tools require the knowledge

---

*Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN, USA. `[ubaru001,saad]@umn.edu`

†Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, Victoria, Australia. `abd-krim.seghouane@unimelb.edu.au`

of the approximate rank of the matrix, i.e., the number of columns to be sampled. In the applications where algorithms such as online PCA [17] and stochastic approximation algorithms for PCA [2] are used, the rank estimation problem is aggravated since the dimension of the subspace of interest changes frequently. This problem also arises in the common problem of *subspace tracking* in signal and array processing [14, 20].

A whole class of useful methods in fields such as machine learning, among many others, consist of replacing the data matrix $X \in \mathbb{R}^{d \times n}$ with a factorization of the form $UV^\top$, where U and V each have k columns: $U \in \mathbb{R}^{d \times k}$ and $V \in \mathbb{R}^{n \times k}$. This strategy is used in a number of methods where the original problem is solved by fixing the rank to a preset value k [28]. Other examples where a similar rank estimation problem arises are when solving numerically rank deficient linear systems of equations [30], or least squares systems [31], in reduced rank regression [52], and in numerical methods for eigenvalue problems such as subspace iteration [58]. In the signal processing context, the rank estimation approach can be used as an alternative to model selection criteria to estimate the number of signals in noisy data [66, 39, 50].

In most of the examples just mentioned, where the knowledge of rank k is required as input, this rank is typically selected in an ad-hoc way. This is mainly because most of the standard rank estimation methods in the existing literature rely on expensive matrix factorizations such as the QR [12], LDL^T or SVD [26].

The rank estimation problem that is addressed in the literature generally focuses on applications. Tests and methods have been proposed in econometrics and statistics to estimate the rank and the rank statistic of a matrix, see, e.g., [16, 53, 9]. In statistical signal processing, rank estimation methods have been proposed to detect the number of signals in noisy data, using model selection criteria and also exploit ideas from random matrix theory [66, 39, 50]. Rank estimation methods have also been proposed in several other contexts, for reduced rank regression models [7, 6], for sparse spiked covariance matrices [8] and for data matrices with missing entries [35], to name a few examples. Article [37] discusses a dimension selection method for a Principal and Independent Component Analysis (PC-IC) combined algorithm. The method is based on a bias-adjusted skewness and kurtosis analysis. However, most of these rank estimation methods require expensive matrix factorizations, and many make some assumptions on the asymptotic behavior of the input matrices. Hence, using these methods for general large matrices in the data applications mentioned earlier is typically not viable.

This paper presents two inexpensive methods to estimate the approximate rank of large matrices. These methods require no form of matrix factorizations, and make no specific statistical, or assumptions on the asymptotic behavior of the matrices. The only assumption is that the input matrix can be approximated in a low dimensional subspace. In this case, there must be a set of relevant eigenvalues in the spectrum whose corresponding eigenvectors span this low dimensional subspace, and these are well separated from the smaller, noise-related eigenvalues (for details see section 2).

The rank estimation methods proposed in this paper rely on approximate spectral densities, also known as density of states (DOS) in solid-state physics [43] (introduced in section 3). An approximate spectral density can be viewed as a probability distribution, specifically a function that gives the probability of finding an eigenvalue at a given point in $\mathbb{R}$. Integrating this DOS function over an appropriate interval yields the count of the relevant eigenvalues that contribute to the approximate rank of the matrix. Spectral densities can be computed in various ways, but this paper focuses on two techniques namely, the Kernel Polynomial Method (KPM) and the Lanczos approximation, discussed in sections 3.1 and 3.2, respectively.

Recently, we proposed fast methods for the estimation of the numerical ranks of large matrices in [64]. The methods in [64] rely on polynomial filtering to estimate the count of the relevant eigenvalues. The approach used there is based on approximately estimating the trace of step functions of the matrices. The step function of the matrix (which is not available without its eigen-decomposition) is approximated by a polynomial of the matrix (which are inexpensive to compute), and a stochastic
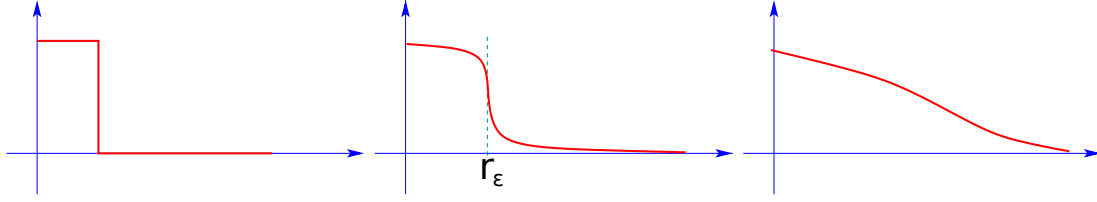
Figure 1: Three different scenarios for eigenvalues of PSD matrices

trace estimator [32] is used to approximate trace. The approach presented in this paper is totally different, and uses approximate spectral density of the matrix to estimate the relevant eigenvalue count. We also present several pertinent applications for these methods and give many additional details about approximate ranks and the methods employed.

The concept of spectral density is defined for real symmetric matrices. So, we consider estimating the approximate rank of a symmetric Positive Semi-Definite (PSD) matrix. If the input matrix X is rectangular or non-symmetric, we will seek the rank of the matrix $XX^\top$ or $X^\top X$. These two matrices need not be formed explicitly. The approximate rank of a PSD matrix typically corresponds to the number of eigenvalues (singular values) above a certain small threshold $\varepsilon > 0$. In order to find a good threshold (the appropriate interval of integration) to use, it is important to detect a gap in the spectrum that separates the smaller, noise related eigenvalues from the relevant eigenvalues which contribute to the rank. This is discussed in detail in section 4, where we introduce a method that exploits the plot of the spectral density function to locate this gap, and select a threshold that best separates the small eigenvalues from the relevant ones. We believe this proposed threshold selection method may be of independent interest in other applications. Section 6 discusses the performance of the two rank estimation techniques on matrices from various applications.

2 Numerical Rank

The *approximate rank* or *numerical rank* of a general $d \times n$ matrix X is often defined by using the closest matrix to X in the 2-norm sense. Specifically, the *numerical ε-rank r_ε* of X, with respect to a positive tolerance ε is

$$r_\varepsilon = \min\{rank(B) : B \in \mathbb{R}^{d \times n}, \|X - B\|_2 \le \varepsilon\}, \tag{1}$$

where $\| \cdot \|_2$ refers to the 2-norm or spectral norm. This standard definition has been often used in the literature, see, e.g., [27, §2.5.5 and 5.5.8], [24, §2], [30, §3.1], [31, §5.1]. The ε-rank of a matrix X is equal to the number of columns of X that are linearly independent for any perturbation of X with norm at most the tolerance ε. The definition is pertinent when the matrix X was originally of rank $r_\varepsilon < \min\{d, n\}$, but its elements have been perturbed by some small error or when the relevant information of the matrix lies in a lower dimensional subspace. The input (perturbed) matrix is likely to have full rank but it can be well approximated by a rank-r_ε matrix.

The singular values of X, with the numerical rank r_ε must satisfy

$$\sigma_{r_\varepsilon} > \varepsilon \ge \sigma_{r_\varepsilon+1}. \tag{2}$$

It should be stressed that the numerical rank r_ε is practically significant only when there is a *well-defined gap* between σ_{r_ε} and $\sigma_{r_\varepsilon+1}$ or in the corresponding eigenvalues of the matrix $X^\top X$ (also highlighted in [24, 30]). It is important that the rank r_ε estimated be robust to small variations of the threshold ε and the eigenvalues of X, and this can be ensured only if there is a good gap between the relevant eigenvalues and those close to zero associated with noise.

3

For illustration, consider the three situations shown in Figure 1. The curves shown in the figure are continuous renderings of simple plots showing eigenvalues λ_i (y-axis) of three different kinds of PSD matrices as a function of i (x-axis). The leftmost case is an idealized situation where the matrix is exactly low rank and the rank is simply the number of positive eigenvalues. The situation depicted in the middle plot is common in situations where the same original matrix of rank say r_ε is perturbed by some small noise or error. In this case, the approximate rank r_ε of the matrix can be estimated by counting the number of eigenvalues larger than a threshold value ε that separates the spectrum into two distinct well separated sets. The third curve shows a situation when the perturbation or the noise added overwhelms the small relevant eigenvalues of the matrix, so there is no clear separation between relevant eigenvalues and the others. In this case there is no clear-cut way of recovering the original rank.

The matrices related to the applications discussed earlier typically belong to the second situation and we shall consider only these cases in this paper. Unless there is an advance knowledge of the noise level or the size of the perturbation, we will need a way to estimate the gap between the small eigenvalues to be neglected and the others. This issue of selecting the gap, i.e., the parameter ε in the definition of ε-rank has been addressed by a few articles in the signal processing literature, e.g., see [50, 47, 38, 39]. Section 4 describes a method to determine this gap and to choose a value for the threshold ε based on the plot of spectral density of the matrix. Once the threshold ε is selected, the rank can be estimated by counting the number of eigenvalues of A that are larger than ε. This can be achieved by integrating the spectral density function of the matrix as described in the following sections.

3 Spectral Density and Rank Estimation

The concept of *spectral density* also known as the *Density of States* (DOS) is widely used in physics [43]. The Density of States of an $n \times n$ symmetric matrix A is formally defined as

$$\phi(t) = \frac{1}{n} \sum_{j=1}^{n} \delta(t - \lambda_j), \tag{3}$$

where δ is the Dirac δ-function and λ_j are the eigenvalues of A labeled decreasingly. Note that, formally, $\phi(t)$ is not a function but a *distribution*. It can be viewed as a probability distribution function which gives the probability of finding eigenvalues of A in a given infinitesimal interval near t. Several efficient algorithms have been developed in the literature to approximate the DOS of a matrix [61, 65, 43] without computing its eigenvalues. Two of these techniques are considered in this paper, from which the rank estimation methods are derived. The first approach is based on the Kernel Polynomial Method (KPM) [67, 65, 59], where the spectral density is estimated using Chebyshev polynomial expansions. The second approach is based on the Lanczos approximation [43], where the relation between the Lanczos procedure and Gaussian quadrature formulas is exploited to construct a good approximation to the spectral density.

As indicated earlier, the number of eigenvalues in an interval $[a, \ b]$ can be expressed in terms of the DOS as

$$\nu_{[a, \ b]} = \int_a^b \sum_j \delta(t - \lambda_j) dt \equiv \int_a^b n\phi(t) dt. \tag{4}$$

We assume the input matrix A to be large and of low numerical rank. This means that most of the eigenvalues of A are close to zero. The idea is to choose a small threshold $\varepsilon > 0$ such that the integration of the DOS above this threshold corresponds to counting those eigenvalues which are large (relevant) and contribute to the approximate rank. Thus, the approximate rank of the matrix A can be obtained by integrating the DOS $\phi(t)$ over the interval $[\varepsilon, \ \lambda_1]$:

$$r_\varepsilon = \nu_{[\varepsilon, \ \lambda_1]} = n \int_\varepsilon^{\lambda_1} \phi(t) dt. \tag{5}$$

4

A method for choosing a threshold value ε based on the spectral density curve is presented in section 4.

In the following sections, we describe the two different approaches for computing an approximate DOS (the Chebyshev polynomial approximation in sec. 3.1 and the Lanczos approximation in 3.2), and derive the expressions for the estimation of approximate matrix ranks based on these two DOS approaches.

3.1   The Kernel Polynomial Method

The Kernel Polynomial Method (KPM) proposed in [59, 65] determines an approximate DOS of a matrix by expanding the DOS in a finite set of Chebyshev polynomials of the first kind, see, e.g., [43] for a discussion. The following is a brief sketch of this technique. The rank expression based on KPM is then derived in the latter part of this section.

To begin with we assume that a change of variable has been made to map the initial interval $[\lambda_n, \lambda_1]$ containing the spectrum into the interval $[-1, 1]$. Since the Chebyshev polynomials are orthogonal with respect to the weight function $(1 - t^2)^{-1/2}$, we seek the expansion of : $\hat{\phi}(t) = \sqrt{1 - t^2}\phi(t) = \sqrt{1 - t^2} \times \frac{1}{n}\sum_{j=1}^{n} \delta(t - \lambda_j)$, instead of the original $\phi(t)$. Then, we write the partial expansion of $\hat{\phi}(t)$ as

$$\hat{\phi}(t) \approx \sum_{k=0}^{m} \mu_k T_k(t),$$

where $T_k(t)$ is the Chebyshev polynomial of degree k and the corresponding expansion coefficient μ_k is formally given by,

$$\mu_k = \frac{2 - \delta_{k0}}{n\pi} \sum_{j=1}^{n} T_k(\lambda_j). \tag{6}$$

Here δ_{ij} is the Kronecker symbol, so $2 - \delta_{k0}$ is 1 when $k = 0$ and 2 otherwise. Observe now that $\sum_{j=1}^{n} T_k(\lambda_j) = \text{Trace}[T_k(A)]$. This trace is approximated via a stochastic trace estimator [32, 54]. For this, we generate a set of random vectors $v_1, v_2, \ldots, v_{n_v}$ that obey a normal distribution with zero mean, and with each vector normalized so that $\|v_l\|_2 = 1, l = 1, \ldots, n_v$. The trace of $T_k(A)$ is then estimated as

$$\text{Trace}(T_k(A)) \approx \frac{n}{n_v} \sum_{l=1}^{n_v} (v_l)^\top T_k(A)v_l. \tag{7}$$

This will lead to the desired approximation of the expansion coefficients

$$\mu_k \approx \frac{2 - \delta_{k0}}{\pi n_v} \sum_{l=1}^{n_v} (v_l)^\top T_k(A)v_l. \tag{8}$$

Scaling back by the weight function, we obtain the approximation for the spectral density function in terms of Chebyshev polynomial of degree m as, $\tilde{\phi}(t) = \frac{1}{\sqrt{1-t^2}} \sum_{k=0}^{m} \mu_k T_k(t)$.

Since the Chebyshev polynomials are defined over the reference interval $[-1, 1]$, a linear transformation is necessary to map the eigenvalues of a general matrix A to this reference interval. The following transformation is used,

$$B = \frac{A - cI}{d} \qquad \text{with} \quad c = \frac{\lambda_1 + \lambda_n}{2}, \quad d = \frac{\lambda_1 - \lambda_n}{2}. \tag{9}$$

The maximum (λ_1) and the minimum (λ_n) eigenvalues of A are obtained from a small number of steps of the standard Lanczos iteration [49, §13.2].

Rank estimation by KPM

The approximate rank of a matrix A is estimated by integrating the DOS function $\phi(t)$ over the interval $[\varepsilon, \lambda_1]$. Since the approximate DOS $\tilde{\phi}(t)$ is defined in the interval $[-1, 1]$, a linear mapping is also required for the integration interval, i.e., $[\varepsilon, \lambda_1] \rightarrow [\hat{\varepsilon}, 1]$, where $\hat{\varepsilon} = \frac{\varepsilon - c}{d}$, with c and d same as in equation (9). The approximate rank can be estimated as

$$r_\varepsilon = \nu_{[\varepsilon, \lambda_1]} \;=\; n \left(\int_{\hat{\varepsilon}}^1 \tilde{\phi}(t)dt \right) \approx n \sum_{k=0}^m \mu_k \left(\int_{\hat{\varepsilon}}^1 \frac{T_k(t)}{\sqrt{1-t^2}} dt \right). \tag{10}$$

Let us consider the integration term in equation (10), which is a function of the summation variable k, and call it γ_k. Consider a general interval $[a, b]$ for integration. Then, by the definition of Chebyshev polynomials we have

$$\gamma_k \;=\; \frac{(2 - \delta_{k0})}{\pi} \int_a^b \frac{\cos(k\cos^{-1}(t))}{\sqrt{1-t^2}} dt.$$

Using the substitution $t = \cos\theta \Rightarrow dt = \sin\theta d\theta$, we get the following expressions for the coefficients γ_k,

$$\gamma_k = \begin{cases} \frac{1}{\pi}(\cos^{-1}(a) - \cos^{-1}(b)) & : k = 0, \\ \frac{2}{\pi}\left(\frac{\sin(k\cos^{-1}(a)) - \sin(k\cos^{-1}(b))}{k} \right) & : k > 0 \end{cases}. \tag{11}$$

Using the above expression and the expansion of the coefficients μ_k, the eigenvalue count over an interval $[a, b]$ by KPM becomes

$$\nu_{[a, b]} \approx \frac{n}{n_v} \sum_{l=1}^{n_v} \left[\sum_{k=0}^m \gamma_k (v_l)^T T_k(B) v_l \right]. \tag{12}$$

Setting the interval $[a, b]$ to $[\hat{\varepsilon}, 1]$, we get the Kernel (Chebyshev) polynomial method expression for the approximate rank r_ε of a symmetric PSD matrix.

Remark 1 (Step functions) *It is interesting to note that the coefficients γ_k derived in (11) are identical to the Chebyshev expansion coefficients for expanding a step function over the interval $[a, b]$. Approximating a step function using Chebyshev polynomials is a common technique used in many applications [60, 19]. Thus, we note that the above rank estimation method by KPM turns out to be equivalent to the rank estimation method based on Chebyshev polynomials presented in [64].*

Damping and other practicalities

Expanding discontinuous functions using Chebyshev polynomials results in oscillations known as *Gibbs Oscillations* near the boundaries (for details see [15]). To reduce or suppress these oscillations, damping multipliers are often added. That is, each γ_k in the expansion (12) is multiplied by a smoothing factor g_k^m which will tend to be quite small for the larger values of k that correspond to the highly oscillatory terms in the expansion. Jackson smoothing is a popular damping approach used in the whereby the coefficients g_k^m are given by the formula

$$g_k^m = \frac{\sin(k+1)\alpha_m}{(m+2)\sin(\alpha_m)} + \left(1 - \frac{k+1}{m+2} \right) \cos(k\alpha_m),$$

where $\alpha_m = \frac{\pi}{m+2}$. More details on this expression can be seen in [33, 19]. Not as well known is another form of smoothing proposed by Lanczos [40, Chap. 4] and referred to as σ-smoothing. It uses the following simpler damping coefficients, called σ factors by the author:

$$\sigma_0^m = 1; \; \sigma_k^m = \frac{\sin(k\theta_m)}{k\theta_m}, k = 1, \ldots, m \text{ with } \theta_m = \frac{\pi}{m+1}.$$
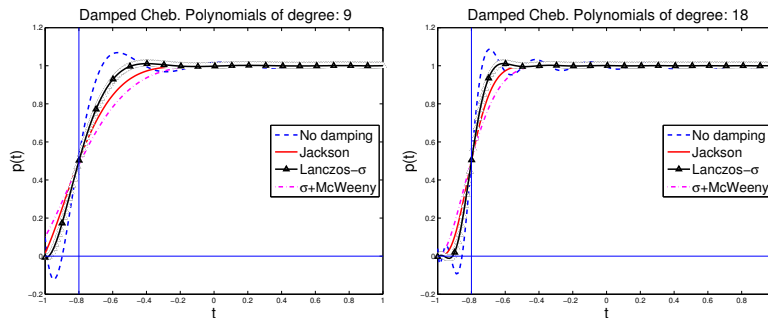
Figure 2: Four different ways of dampening Gibbs oscillations for Chebyshev approximation. All polynomials have the same degree (9 on the left and 18 on the right).

The damping factors are small for larger values of k and this has the effect of reducing the oscillations. The Jackson polynomials have a much stronger damping effect on these last terms than the Lanczos σ factors. For example the very last factors, and their approximate values for large m's, are in each case:

$$g_m^m = \frac{2\sin^2(\alpha_m)}{m+2} \approx \frac{2\pi^2}{(m+2)^3}; \qquad \sigma_m^m = \frac{\sin(\theta_m)}{m\theta_m} \approx \frac{1}{m}.$$

Jackson coefficients tend to over-damp the oscillations at the expense of sharpness of the approximation. The Lanczos smoothing can be viewed as an intermediate form of damping between no damping and Jackson damping. A comparison of the three forms of damping is shown in Figure 2. To the three forms of damping (no-damping, Jackson, σ-damping), we have added a fourth one which consists of compounding the degree 3 McWeeny filter with the Chebyshev polynomials. In the numerical experiments, we have used Lanczos σ damping.

An important practical consideration is that we can economically compute vectors of the form $T_k(B)v$, where B is as defined in (9) since the Chebyshev polynomials obey a three term recurrence. That is, we have $T_{k+1}(t) = 2tT_k(t) - T_{k-1}(t)$ with $T_0(t) = 1, T_1(t) = t$. As a result, the following iteration can be used to compute $w_k = T_k(B)v$, for $k = 0, \cdots, \cdots, m$

$$w_{k+1} = 2Bw_k - w_{k-1}, k = 1, 2, \ldots, m-1, \tag{13}$$

with $w_0 = v$; $w_1 = Bv$.

Remark 2 *Note that if the matrix B is of the form $B = Y^\top Y$, where Y is a linear transformation of the data matrix X using the mapping (9), then we need not compute the matrix B explicitly since the only operations that are required with the matrix B are matrix-vector products.*

3.2 The Lanczos approximation approach

The Lanczos approximation technique for estimating spectral densities discussed in [43] combines the Lanczos algorithm with multiple randomly generated starting vectors to approximate the DOS. We can develop an efficient method for the estimation of approximate rank based on this approach.

The Lanczos algorithm generates an orthonormal basis V_m for the *Krylov subspace:* Span$\{v_1, Av_1, \ldots, A^{m-1}v_1\}$ such that $V_m^\top AV_m = T_m$, where T_m is an $m \times m$ tridiagonal matrix, see [57] for details. The premise of the Lanczos approach to estimate the DOS (and the approximate rank) is the important observation that the Lanczos procedure builds a sequence of *orthogonal polynomials* with respect to the discrete (Stieljes) inner product,

$$\int p(t)q(t)dw(t) \equiv (p(A)v_1, q(A)v_1), \tag{14}$$

where p and q are orthogonal polynomials. The Gauss quadrature approximation for this integral (14) can be computed using the Lanczos procedure as,

$$\int p(t)dw(t) \approx \sum_{k=1}^{m} \tau_k^2 p(\theta_k) \quad \text{with} \quad \tau_k^2 = \left[e_1^\top y_k \right]^2, \tag{15}$$

where θ_k, $k = 1, \ldots, m$ are the eigenvalues of T_m and y_k, $k = 1, \ldots, m$ the associated eigenvectors, see [25] for details.

Assume now that the initial vector v_1 of the Lanczos sequence is expanded in the eigenbasis $\{u_i\}_{i=1}^n$ of A as $v_1 = \sum_{i=1}^n \beta_i u_i$ and consider the discrete (Stieljes) integral which we write formally as $\int p(t)dw(t) = (p(A)v_1, v_1) = \sum_{i=1}^n \beta_i^2 p(\lambda_i)$. We can view this as a certain distribution ϕ_{v_1} applied to p and write $(p(A)v_1, v_1) \equiv \langle \phi_{v_1}, p \rangle$. Assuming that $\beta_i^2 = 1/n$ for all i, this ϕ_{v_1} becomes exactly the DOS function. Indeed, in the sense of distributions,

$$\begin{aligned} \langle \phi_{v_1}, p \rangle \equiv (p(A)v_1, v_1) &= \sum_{i=1}^n \beta_i^2 p(\lambda_i) = \sum_{i=1}^n \beta_i^2 \langle \delta_{\lambda_i}, p \rangle \\ &= \frac{1}{n} \sum_{i=1}^n \langle \delta_{\lambda_i}, p \rangle \equiv \langle \phi_{v_1}, p \rangle, \end{aligned}$$

where δ_{λ_i} is a δ-function at λ_i. We now invoke the Gaussian quadrature rule (15) and write: $\langle \phi_{v_1}, p \rangle \approx \sum_{k=1}^m \tau_k^2 p(\theta_k) = \sum_{k=1}^m \tau_k^2 \langle \delta_{\theta_k}, p \rangle$ from which we obtain

$$\phi_{v_1} \approx \sum_{k=1}^{m} \tau_k^2 \delta_{\theta_k}.$$

So, in the ideal case when the vector v_1 has equal components $\beta_i = 1/\sqrt{n}$ we could use the above Gauss quadrature formula to approximate the DOS. Since the β_i's are not equal, we will use several vectors v and average the result of the above formula over them. For additional details, see [43].

If $(\theta_k^{(l)}, y_k^{(l)})$, $k = 1, 2, ..., m$ are eigenpairs of the tridiagonal matrix T_m corresponding to the starting vector $v_l, l = 1, \ldots, n_v$ and $\tau_k^{(l)}$ is the first entry of $y_k^{(l)}$, then the DOS function by Lanczos approximation is given by

$$\tilde{\phi}(t) = \frac{1}{n_v} \sum_{l=1}^{n_v} \left(\sum_{k=1}^{m} (\tau_k^{(l)})^2 \delta(t - \theta_k^{(l)}) \right). \tag{16}$$

The above function is a weighted spectral distribution of T_m, where τ_k^2 is the weight for the corresponding eigenvalue θ_k and it approximates the spectral density of A.

Rank estimation by the Lanczos approximation

Applying the distribution $\tilde{\phi}$ in (16) to the step function $h_{[a, b]}$ that has value one in the interval $[a, b]$ and zero elsewhere, we obtain the integral of the probability distribution of finding an eigenvalue in $[a, b]$. This must be multiplied by n to yield an approximate eigenvalue count in the interval $[a, b]$:

$$\nu_{[a, b]} \approx \frac{n}{n_v} \sum_{l=1}^{n_v} \left(\sum_k (\tau_k^{(l)})^2 \right) \quad \forall k : a \le \theta_k \le b. \tag{17}$$

The approximate rank of a matrix can be written in two forms $r_\varepsilon = \nu_{[\varepsilon, \lambda_1]}$ or $r_\varepsilon = n - \nu_{[\lambda_n, \varepsilon]}$. Since the Lanczos method gives better approximations for extreme eigenvalues, the second definition above is preferred. Therefore, the expression for the approximate rank of a symmetric PSD matrix by Lanczos approximation is given by

$$r_\varepsilon \approx \frac{n}{n_v} \sum_{l=1}^{n_v} \left(1 - \sum_k (\tau_k^{(l)})^2 \right) \quad \forall k : \lambda_n \le \theta_k \le \varepsilon. \tag{18}$$
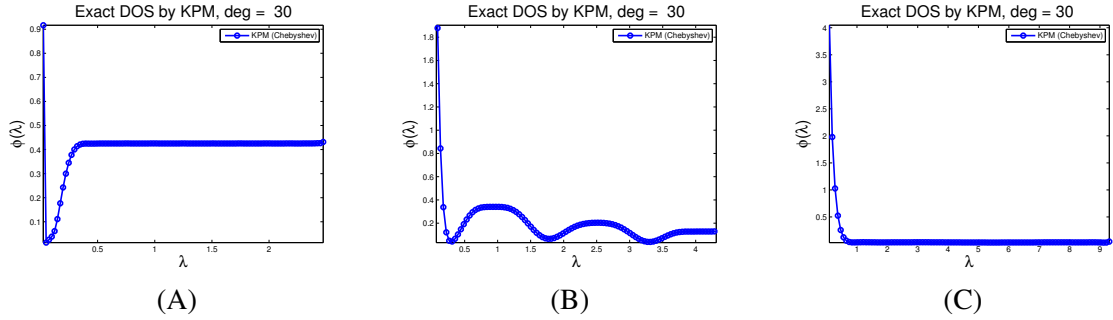
Figure 3: Exact DOS plots for three different types of matrices, namely (A) exactly low rank, (B) numerically low rank with large number of clustered relevant eigenvalues, and (C) numerically low rank with fewer of widely spread eigenvalues.

4 Threshold selection

An important requirement for the rank estimation techniques described in this paper is to define the interval of integration, i.e., $[\varepsilon, \lambda_1]$ in the case of KPM or $[\lambda_n, \varepsilon]$ in the case of the Lanczos approximation. As discussed in section 3.1, λ_1 and λ_n can be easily estimated with a few Lanczos iterations [49, 57]. The estimated rank is sensitive to the threshold ε selected. So, it is important to provide tools to estimate this tolerance parameter ε that separates the small eigenvalues which are assumed to be perturbations of a zero eigenvalue from the relevant eigenvalues, which correspond to nonzero eigenvalues in the original matrix. This is addressed in this section.

4.1 Existing Methods

A few methods to select the threshold ε have been developed for specific applications. In a typical application we only have access to the perturbed matrix $A + E$, where E is unknown and this perturbed matrix is typically of full rank. Even though E is unknown, we may have some knowledge of the source of E, for example in subspace tracking, we may know the expected noise power. In some cases, we may obtain an estimator of the norm of E using some known information. See [30, §3.1] for more details on estimating the norm of E in such cases.

The problem of estimating ε is similar to the problem of defining a threshold for hypothesis testing [39]. The articles mentioned in section 2 discuss a few methods to estimate the parameter ε using information theory criteria. Also, ideas from random matrix theory are incorporated, particularly when the dimension of signals is greater than the number of samples obtained. In the low rank approximation case, the maximum approximation error tolerance that is acceptable might be known. In this case, we may select this error tolerance as the threshold, since the best approximation error possible is equal to the first singular value that is below the threshold selected.The DOS plot analysis method introduced next requires no prior knowledge of noise level, or perturbation error or error tolerance, and makes no assumptions on the distribution of noise.

4.2 DOS plot analysis

For motivation, let us first consider a matrix that is exactly of low rank and observe the typical shape of its DOS function plot. As an example we take an $n \times n$ PSD matrix with rank $k < n$, that has k eigenvalues uniformly distributed between 0.2 and 2.5, and whose remaining $n - k$ eigenvalues are equal to zero. An approximate DOS function plot of this low rank matrix is shown in figure 3(A). The DOS is generated using KPM, with a degree $m = 30$ where the coefficients μ_k are estimated using the exact trace of the Chebyshev polynomial functions of the matrix. Jackson damping is used to eliminate oscillations in the

plot. The plot begins with a high value at zero indicating the presence of a zero eigenvalue with high multiplicity. Following this, it quickly drops to almost a zero value, indicating a region where there are no eigenvalues. This corresponds to the region just above zero and below 0.2. The DOS increases at 0.2 indicating the presence of new eigenvalues. Because of the uniformly distributed eigenvalues between 0.2 and 2.5, the DOS plot has a constant positive value in this interval.

To estimate the rank k of this matrix, we can count the number of eigenvalues in the interval $[\varepsilon, \lambda_1] \equiv [0.2, 2.5]$ by integrating the DOS function over the interval. The value $\lambda_1 = 2.5$ can be replaced by an estimate of the largest eigenvalue. The initial value $\varepsilon = 0.2$ can be estimated as *the point immediately following the initial sharp drop observed* or *the mid point of the valley.* For low rank matrices such as the one considered here, we should expect to see this sharp drop followed by a valley. The cutoff point between zero eigenvalues and relevant ones should be at the location where the curve ceases to decrease, i.e., the point where the derivative of the spectral density function becomes zero (local minimum) for the first time. Thus, the threshold ε can be selected as

$$\varepsilon = \min\{t : \phi'(t) = 0, \lambda_n \leq t \leq \lambda_1\}. \tag{19}$$

For more general numerically rank deficient matrices, the same idea based on the DOS plot can be employed to determine the approximate rank. Defining a cut-off value between the relevant singular values and insignificant ones in this way works when there is a gap in the matrix spectrum. This corresponds to matrices that have a cluster of eigenvalues close to zero, which are zero eigenvalues perturbed by noise/errors, followed by an interval with few or no eigenvalues, a gap, and then clusters of relevant eigenvalues, which contribute to the approximate rank. Two types of DOS plots are often encountered depending on the number of relevant eigenvalues and whether they are in clusters or spread out wide.

Figures 3(B) and (C) show two sample DOS plots which belong to these two categories, respectively. Both plots were estimated using KPM and the exact trace of the matrices, as in the previous low rank matrix case. The middle plot (figure 3(B)) is a typical DOS curve for a matrix which has a large number of eigenvalues related to noise which are close to zero and a number of larger relevant eigenvalues which are in a few clusters. The spectral density curve displays a fast decrease after a high value near zero eigenvalues due to the gap in the spectrum and the curve increases again due the appearance of large eigenvalue clusters. In this case, we can use equation (19) to estimate the threshold ε.

In the last DOS plot on figure 3(C), the matrix has again a large number of eigenvalues related to noise which are close to zero, but the number of relevant eigenvalues is smaller and these eigenvalues are spread farther and farther apart from each other as their values increase, (as for example when $\lambda_i = K(n-i)^2$.) The DOS curve has a similar high value near zero eigenvalues and displays a sharp drop, but it does not increase again and tends to hover near zero. In this case, there is no valley or local minimum, so the derivative of the DOS function may not reach the value zero. The best we can do here is detect a point at which the derivative exceeds a certain negative value, for the first time, indicating a significant slow-down of the initial fast decrease. In summary, the threshold ε for all three cases can be selected as

$$\varepsilon = \min\{t : \phi'(t) \geq tol, \lambda_n \leq t \leq \lambda_1\}. \tag{20}$$

Our sample codes use $tol = -0.01$ which seems to work well in practice.

When the input matrix does not have a large gap between the relevant and noisy eigenvalues (when numerical rank is not well-defined), the corresponding DOS plot of that matrix will display similar behavior as the plot in figure 3(C), except the plot does not go to zero. That is, the DOS curve will have a similar knee as in figure 3(C). In such cases, one can again use the above equation (20) for threshold selection, or use certain knee (of a curve) detection methods.

Threshold selection for the Lanczos method

For the Lanczos approximation method, the threshold can be estimated in a few additional ways. In the Lanczos DOS equation (16), we see that the τ_i^2s are the weights of the DOS function at θ_is, respectively. So, when the plot initially drops to zero (at the gap region), the corresponding value of τ_i^2 must be zero or close to zero, and then increase. Therefore, one of the methods to select the threshold for the Lanczos approximation is to choose the θ_i for which the corresponding difference in τ_i^2s becomes positive for the first time. That is,

$$\varepsilon = \min\{\theta_i : \Delta\tau_i^2 \geq 0, 1 \leq i \leq m-1\}, \tag{21}$$

where $\Delta\tau_i^2 = \tau_{i+1}^2 - \tau_i^2$.

The distribution of θ_i, the eigenvalues of the tridiagonal matrix T_m must be similar to the eigenvalue distribution of the input matrix. This is true at least for eigenvalues located at either end of the spectrum. It is possible to use this information to select the threshold based on a gap for the θ_is, i.e., a gap in the spectrum of T_m. However, this does not work as well as the class of methods based on the DOS plot.

Other applications

It is inexpensive to compute an approximate DOS of a matrix using the methods described in section 3. Hence, the above threshold selection methods using the DOS plots could be of independent interest in other applications where such thresholds need to be selected. For example, in the estimation of the trace (or diagonal) of matrix functions such as exponential function where the smaller eigenvalues of the matrix do not contribute much to the trace, and inverse functions where the larger eigenvalues of the matrix do not contribute much, etc., and also in the interior eigenvalue problems.

5 Algorithms

In this section we present the proposed algorithms to estimate the approximate rank of a large matrix, and discuss their computational costs. We then discuss the pros and cons of the two proposed methods. The convergence analysis for the methods is also briefly discussed at the end of this section. Algorithms 1 and 2 give the algorithms to estimate the approximate rank r_ε by the Kernel Polynomial method and by the Lanczos approximation method, respectively.

Algorithm 1 Rank estimation by KPM

Input: An $n \times n$ symmetric PSD matrix A, λ_1 and λ_n of A, degree of polynomial m, and the number sample vectors $\mathrm{n_v}$ to be used.
Output: The approximate rank r_ε of A.
1. Generate the random starting vectors $v_l : l = 1, \ldots \mathrm{n_v}$, such that $\|v_l\|_2 = 1$.
2. Transform the matrix A using (9) to B and form an $m \times \mathrm{n_v}$ matrix Y (using recurrence eq. (13)) with

$$Y(k,l) = (v_l)^\top T_k(B)v_l : l = 1, \ldots, \mathrm{n_v}, k = 0, \ldots, m$$

3. Estimate the coefficients μ_ks from Y and obtain the DOS $\tilde{\phi}(t)$ using eq. (8).
4. Estimate the threshold ε from $\tilde{\phi}(t)$ using the eq. (20).
5. Estimate the coefficients γ_k for the interval $[\hat{\varepsilon}, 1]$ and compute the approximate rank r_ε using eq. (12) and Y.

Computational Cost: The expensive step in estimating the rank by KPM (Algorithm 1) is in forming the matrix Y, i.e., computing the scalars $(v_l)^\top T_k(A)v_l$ for $l = 1, \ldots, \mathrm{n_v}, k = 0, \ldots, m$ (step 2). Hence,

---

Algorithm 2 Rank estimation by the Lanczos method

Input: An $n \times n$ symmetric PSD matrix A, the number of Lanczos steps (degree) m, and the number sample vectors n_v to be used.

Output: The approximate rank r_ε of A.

1. Generate the random starting vectors $v_l : l = 1, \ldots n_v$, such that $\|v_l\|_2 = 1$.

2. Apply m steps of Lanczos to A with these different starting vectors. Save the eigenvalues and the square of the first entries of the eigenvectors of the tridiagonal matrices. I.e., matrices W and V with

$$W(k, l) = \theta_k^{(l)}, V(k, l) = (\tau_k^{(l)})^2 : l = 1, .., n_v, k = 0, .., m$$

3. Estimate the threshold ε using matrices W and V and eq. (20) or (21).

Note: Eq. (20) requires obtaining the DOS $\tilde{\phi}(t)$ by eq. (16).

4. Compute the approximate rank r_ε for the threshold ε selected, using eq. (18) and matrices W and V.

---

the computational cost for the rank estimation by the Kernel polynomial method will be $O(\mathrm{nnz}(A)mn_v)$ for sparse matrices, where $\mathrm{nnz}(A)$ is the number of nonzero entries of A , $O(n^2 mn_v)$ for general $n \times n$ dense symmetric PSD matrices. Similarly, the expensive step in Algorithm 2 is computing the m Lanczos steps with different starting vectors. Hence, the computational cost for rank estimation by the Lanczos approximation method will be $O((\mathrm{nnz}(A)m + nm^2)n_v)$ for sparse matrices, $O((n^2 m + nm^2)n_v)$ for general $n \times n$ dense symmetric PSD matrices. Therefore, these algorithms are inexpensive (almost linear in terms of the number of entries in A for large matrices), compared to the methods that require matrix factorizations such as the QR or SVD.

Remark 3 *In some rank estimation applications, we may wish to estimate the corresponding spectral information (the eigenpairs or the singular triplets), after the approximate rank is estimated. This can be easily done with a Rayleigh-Ritz projection type methods, exploiting the vectors $T_k(A)v_l$ generated for rank estimation in KPM or the tridiagonal matrices T_m from the Lanczos approximation.*

Comparison: Here, we discuss the pros and cons of the two methods: the Lanczos approach and the polynomial approximations. For the Lanczos method, a practical disadvantage is that one needs to perform orthogonalization and store the intermediate Lanczos vectors. In terms of storage, polynomial approximations are more economical. Polynomial approximations are also independent of the matrix. Hence, we can easily obtain the posterior error estimates for a given degree of the polynomial by sampling the function. Matrix related computations are not required for such posterior error estimations unlike the Lanczos approach, where such error estimation is also not straightforward. We can obtain estimates by extending the analysis given in [55] for the Lanczos approach, however, these are not sharp. The disadvantage of polynomial approximation methods is that one generally needs to estimate the spectrum interval, which might not be easy for matrices with very small eigenvalues. The Lanczos approach does not require spectrum interval estimation. In addition, results in section 6 show that the convergence obtained by the Lanczos method will be better than those obtained by Chebyshev polynomial approximations for a given degree m.

Convergence and error estimation: Here we discuss how theoretical error bounds can be obtained (and improved) for the two methods. In both rank estimation methods described, the error in the rank estimated will depend on the two parameters we set, namely, the degree of the polynomial / number of Lanczos steps m, and the number of sample vectors n_v. The parameter n_v comes from the stochastic trace estimator (7) employed in both methods to estimate the trace. In the Lanczos approach, the

stochastic trace estimator is used in disguise, see Theorem 3.1 in [43]. That is, the idea of using multiple starting vectors for Lanczos to achieve β_i's equal on an average comes from the stochastic trace estimation. For this stochastic trace estimator, convergence analysis was developed in [3], and improved in [54] for sample vectors with different probability distributions. The best known convergence rate for (7) is $O(1/\sqrt{n_v})$. The error due to the trace estimator is independent of the size n of the input matrix, and to achieve $(1 + \epsilon)$ relative error approximation for the trace, we need $n_v = \Theta(1/\epsilon^2)$.

In section 3.1, we saw that the rank expression obtained by KPM is similar to the expression for approximating a step function of the matrix. It is not straightforward to develop the theoretical analysis for approximating a step function as in (12), since the function we are approximating is discontinuous. Article [1] gives the convergence analysis on approximating a step function. Any polynomial approximation can achieve a convergence rate of $O(1/m)$ [1]. However, this rate is obtained for point by point analysis (at the vicinity of discontinuity points), and uniform convergence cannot be achieved due to the Gibbs phenomenon.

We can obtain an improved theoretical result, if we first replace the step function by a piecewise linear approximation, and then employ polynomial approximation. Article [56] shows that uniform convergence can be achieved using polynomial approximation when the step function is constructed as a spline (piecewise linear) function. For example,

$$
\tilde{h}(t) = \begin{cases} 0 & : for\ t \in [-1, \varepsilon_0) \\ \frac{t - \varepsilon_0}{\varepsilon_1 - \varepsilon_0} & : for\ t \in [\varepsilon_0, \varepsilon_1) \\ 1 & : for\ t \in [\varepsilon_1, 1] \end{cases} .
\tag{22}
$$

For Chebyshev approximations, uniform convergence can be achieved if the function approximated is continuous and differentiable, see [60, Thm. 7.2].

For the Lanczos approximations, the error estimation for the quadrature rule is given in [25], but it applies only for continuous and highly differentiable functions . We can show that the quadrature rule converges twice as fast as any polynomial approximation, using Theorem 19.3 in [60]. However, since the step function is discontinuous, we cannot use these analyses directly. Although in practice, we see that accurate ranks can be estimated with fairly small number of Lanczos steps, and it converges faster than Chebyshev, see section 6.3.

A possible alternative to obtain stronger theoretical results for both methods is to replace the step function by an analytic function, for example, use a shifted version of $\tanh(\alpha t)$ function. In this case, both the Lanczos and Chebyshev approximations converge exponentially, see Thm. 8.2 and 19.3 in [60] for respective results. Convergence rate of Lanczos will be twice that of Chebyshev. An analysis with a similar idea (of using surrogate function) was recently presented in [22], where the step function was replace by a ridge regression type function.

The bounds given above for both the trace estimator and the approximation of step functions are general worst case results and are too pessimistic, since in practice we can get accurate rank estimation by the proposed methods. Therefore, such complicated implementations with surrogate functions (as in eq. (22) or $\tanh(\alpha t)$) are unnecessary in practice.

Using the following ideas, we can obtain error estimates for the methods and decide on the choice of the degree m and the number of vectors n_v. As mentioned earlier, we can compute the posterior error estimates for a given degree m for the Chebyshev method by sampling the obtained polynomial and the function inside the spectrum and computing the pointwise errors. If this error is satisfactory, we can retain the current degree m, else we can increase it. For the Lanczos method, such error estimation is not straightforward. We can obtain rough estimates by extending the analysis given in [55]. For the stochastic trace estimator in both methods, we know that the trace is obtained by averaging $v_l^\top f(A) v_l$ over different random vectors v_l. Along with the averages, we can also compute the standard deviations and compute standard deviation error bars. The width of the error bars (which reduces as n_v increases)

gives us a rough idea of how close we are to the exact trace, see section 6.3 for an empirical example. Thus, based on these error estimates and error bars, we can choose m and n_v appropriately.

6   Numerical Experiments

In this section, we illustrate the performance of the threshold selection method and the two rank estimation methods proposed in this paper on matrices from various applications. First, we consider three example matrices whose spectra belong to the three cases discussed in section 2.

6.1   Sample Matrices

In section 2, we discussed the three categories of eigenvalue distributions that are encountered in application matrices. Here, we illustrate how the DOS plot threshold selection method can be used to determine the cut off point ε for three example matrices which belong to these three categories, and use KPM to estimate their approximate ranks (the counts of eigenvalues above ε). Recall that in the first two cases, there exists a gap in the spectrum since the noise added is small compared with the magnitude of the relevant eigenvalues. So, the threshold selection method is expected to select an ε value in this gap. In the third case, the noise added overwhelms the relevant eigenvalues and the matrix is no longer numerically low rank. In this case, the approximate rank estimated cannot be accurate (in fact the approximate rank definition does not hold) as illustrated in this example.

Let us consider a simple example of a low rank matrix formed by a subset of columns of a Hadamard[1] matrix. Consider a Hadamard matrix of size 2048 (scaled by $1/\sqrt{2048}$ to make all eigenvalues equal to 1), and select the first 128 columns to form a subset Hadamard matrix H. Then, a 2048×2048 low rank matrix is formed as $A = HH^\top$, which has rank 128 (has 128 eigenvalues equal to one and the remaining equal to zero). Such low rank matrices can also be obtained as $A = X_1 X_2$, where $X_1 \in \mathbb{R}^{n \times r}$ and $X_2 \in \mathbb{R}^{r \times n}$ are some matrices. Depending on the magnitude of noise added to this matrix, three situations arise as discussed in section 2. The three spectra depicted in the first row of figure 4 belong to these three situations, respectively.

Case 1: Negligible Noise   This is an ideal case where noise added is negligible ($\sigma = 0.001$, SNR $= 20 \log_{10}\left(\|A\|_F/\|N\|_F\right) = 38.72$) and the shape of the original spectrum is not affected much, as depicted by the first plot in the figure. The matrix is no longer low rank since the zero eigenvalues are perturbed by noise, but the perturbations are very small. Clearly, there is a big gap between the relevant eigenvalues and the noisy eigenvalues. The second plot in the first column (left middle) of figure 4 shows the DOS plot of this matrix obtained by KPM with a Chebyshev polynomial of degree $m = 50$ and a number of samples $n_v = 30$. The threshold ε (the gap) estimated by the method described in section 4 using this DOS plot was equal to $\varepsilon = 0.52$ (taking mid point of the valley). The third plot in the same column (left bottom) plots the approximate ranks estimated by integrating the DOS above over the interval $[\varepsilon, \lambda_1] = [0.52, 1.001]$, using equation (10). In the plot, the circles indicate the approximate ranks estimated with the ℓth sample vector, and the dark line is the cumulative (running) average of these estimated approximate rank values. The average approximate rank estimated over 30 sample vectors is approximately 128.34. This is quite close to the exact number of eigenvalues in the interval which is 128, and is indicated by the dash line in the plot.

Case 2: Low Noise Level   In this case, the added noise ($\sigma = 0.004$, SNR $= 14.63$) causes (significant) perturbations in the eigenvalues, as shown in the first plot of the second column in figure 4. However,

---

[1]A Hadamard matrix is chosen since its eigenvalues are known apriori and is easy to generate. Reproducing the experiment will be easier.
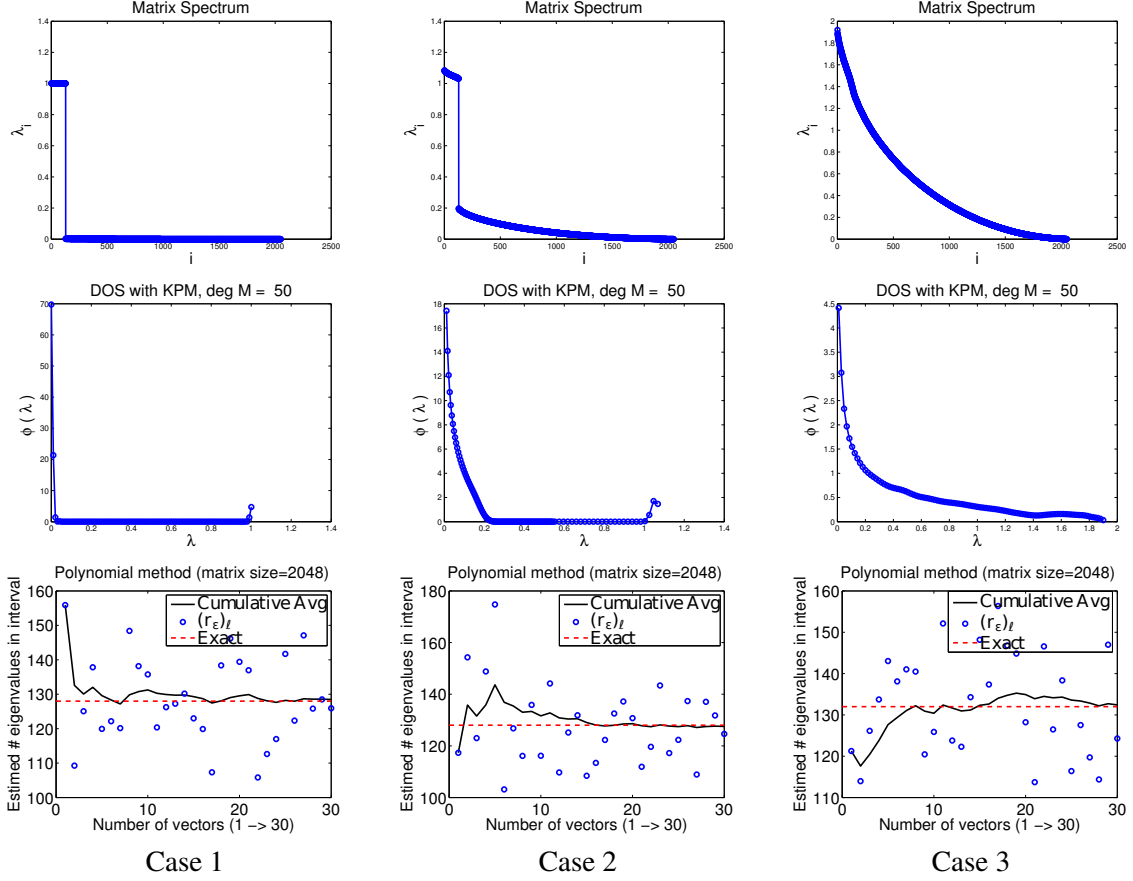
Figure 4: The matrix spectra, the corresponding DOS found by KPM and the approximate ranks estimated for the three cases.

there is still a gap between the relevant eigenvalues and the noise related eigenvalues, which can be exploited by the proposed threshold selection method. The middle plot in the figure shows the DOS plot for this case obtained using KPM with same parameters as before. The threshold ε selected from this plot was $\varepsilon = 0.61$ (mid point of the valley). The approximate ranks estimated are plotted in the last plot of the second column (middle bottom) of the figure. The average approximate rank estimated over 30 sample vectors is equal to 127.53. The approximate rank estimated by the method is still close to the actual relevant eigenvalue count.

Case 3: Overwhelming Noise In this case, the magnitude of the noise added ($\sigma = 0.014$, SNR $= -7.12$) is high and the perturbations in the zero eigenvalues overwhelms the relevant eigenvalues. As we can see in the spectrum of this matrix shown in the top right plot of figure 4, there is no gap in the spectrum and the matrix is not numerically low rank. The corresponding DOS plot (shown in the right middle plot) also does not show a sharp drop, and rather decreases gradually. The threshold ε estimated by the threshold selection method was $\varepsilon = 1.39$. We can see a small inflexion in the DOS plot near 1.4 and the method selected this point as the threshold since the derivative of the DOS function goes above the tolerance for the first time after this point. The approximate ranks estimated are plotted in the bottom right plot of the figure. The average approximate rank estimated over 30 sample vectors is equal to 132.42. There are 132 eigenvalues above the threshold. Interestingly, even though the approximate rank definition does not hold in this case, the rank estimated by the method is not too far from the original rank.
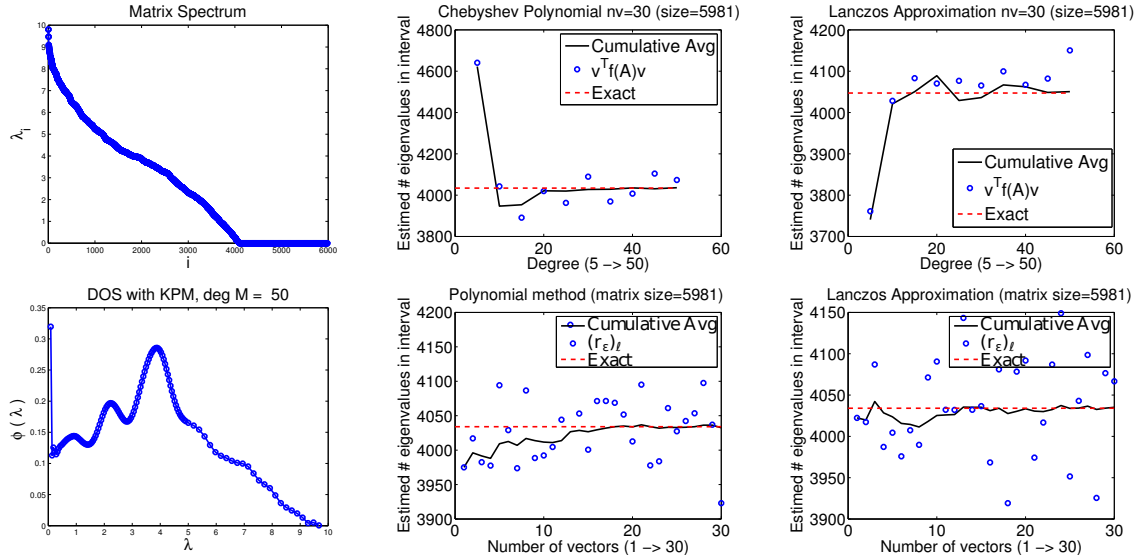
15

Figure 5: The spectrum, the DOS found by KPM, and the approximate ranks estimation by Kernel Polynomial method and by Lanczos approximation for the example `ukerbe1`.

6.2 Matrices from Applications

The following experiments will illustrate the performance of the two rank estimation techniques on matrices from various applications. The first example is with a $5,981 \times 5,981$ matrix named `ukerbe1` from the AG-Monien group made available in the University of Florida (UFL) sparse matrix collection [18] (the matrix is a Laplacian of an undirected graph from a 2D finite element problem). The performance of the Kernel Polynomial method and the Lanczos approximation method for estimating the approximate rank of this matrix is illustrated in figure 5.

The top left plot in figure 5 shows the matrix spectrum and the bottom left plot shows the DOS plot obtained using KPM with degree $m = 50$ and a number of samples $n_v = 30$. The threshold ε (the gap) estimated using this DOS plot was $\varepsilon = 0.16$. The top middle figure plots the estimated approximate ranks for different degrees of the polynomials used by KPM, with $n_v = 30$ (black solid line). The blue circles are the value of $v^\top f(A)v$ (one vector) when f is a step function approximated by k degree Chebyshev polynomials. The bottom middle plot in the figure plots the approximate ranks estimated by KPM for degree $m = 50$ for different number of starting vectors. The average approximate rank estimated over 30 sample vectors is equal to 4033.49. The exact number of eigenvalues in the interval is 4034, (indicated by the dash line in the plot).

Similarly, the top right plot in figure 5 shows the estimated approximate ranks by the Lanczos approximation method using different degrees (or the size of the tridiagonal matrix) and the number of sample vectors $n_v = 30$. The bottom right plot in the figure shows the estimated approximate ranks by the Lanczos approximation method using a degree (or the size of the tridiagonal matrix) $m = 50$ and different number of sample vectors. The average approximate rank estimated over 30 sample vectors is equal to 4035.19. The plots illustrate how the parameters m and n_v affect the ranks estimated. In this case ($m = 50, n_v = 30$), the number of matrix-vector multiplications required for both rank estimator techniques is 1500. A typical degree of the polynomial or the size of the tridiagonal matrix required by these methods is around $40 - 100$.

Timing Experiment In this experiment, we illustrate with an example how fast these methods can be. A sparse matrix of size 1.25×10^5 called `Internet` from the UFL database is considered, with $nnz(A) = 1.5 \times 10^6$. It took only 7.18 secs on average (over 10 trials) to estimate the approximate rank
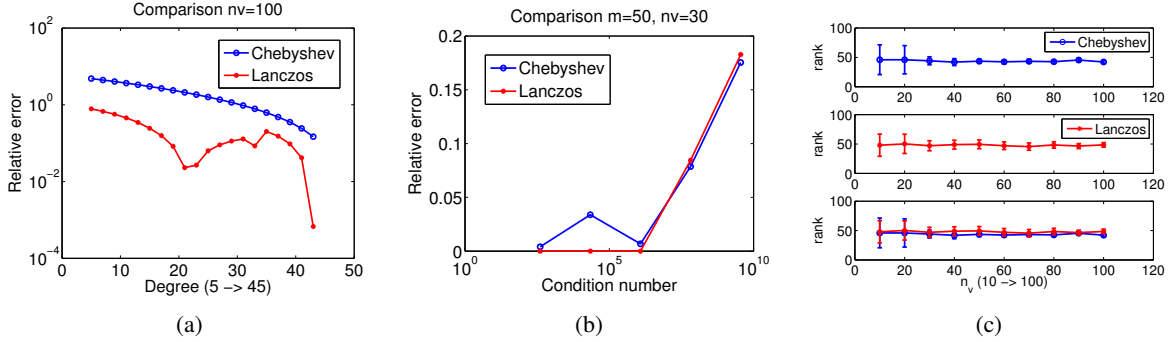
Figure 6: Performance comparison between Chebyshev and Lanczos methods: (a) Relative error vs. degree m, (b) Relative error vs. condition number of the matrices, and (c) estimation and standard error vs. number of starting vectors.

of this matrix using the Chebyshev polynomial method on a standard $3.3GHz$ Intel-i5 machine. It will be extremely expensive to compute its rank using an approximate SVD, for example using the `svds` or `eigs` function in Matlab which rely on ARPACK. It took around 2 hours to compute 4000 singular values of the matrix on the same machine. Rank estimations based on rank-revealing QR factorizations or the standard SVD are not possible for such large matrices on a standard workstation such as the one we used.

6.3 Comparisons

In this section, we compare the performances of the two rank estimation methods with respect to different parameters. First, in Figure 6(a), we compare the relative errors obtained by the two methods for different degrees m chosen. We consider the sparse matrix `california` (a graph Laplacian matrix) of size 9664×9664, nnz $\approx 10^5$ and $\kappa \approx 5 \times 10^4$ from the University of Florida (UFL)database. The number of starting vectors $n_v = 100$ for both cases, and the same threshold ε was chosen. The figure shows that the Lanczos method is superior in accuracy compared to the Chebyshev method. We get roughly 3 digits of accuracy with just a degree of around 40. The plots are averages over 5 trails. In the second experiment, we evaluate the performances of the methods with respect to the condition number of the matrix. We consider a Hadamard matrix H of size 8192 and form the test matrix as $HDH^{\top}$, where D is a diagonal matrix with entries such that the desired condition number is obtained. Figure 6(b) plots the relative errors obtained by the two methods for the rank estimations of the matrices with different condition numbers. The degree and the number of starting vectors used in both cases were $m = 50$ and $n_v = 30$. We observe that both methods have very similar performances. The errors increase as the condition number increases in both cases, in a similar fashion.

For very large matrices ($\sim 10^6$ and above), it is impractical to compute the exact numerical rank for verification. To gauge the approximation quality, we approximate the estimator variance by using sample variance and show the standard errors. That is, since the stochastic trace estimator computes trace as an average over different random vectors, we can also compute the standard deviation of these estimates for different number of vectors. Figure 6(c) plots the ranks estimated and the error bars obtained for different number of starting vectors for the matrix `webbase-1M` (Web connectivity matrix) of size $10^6 \times 10^6$ obtained from the UFL database. For Lanczos, degree $m = 80$ and for Chebyshev, $m = 120$ was chosen. The width of the error bars (which reduces as n_v increases) gives us a rough idea of how close we might be to the exact trace.

Table 1 lists the ranks estimated by KPM and the Lanczos methods for a set of sparse matrices from various applications. All matrices were obtained from the UFL Sparse Matrix Collection [18], and are sparse. The matrices, their applications and sizes are listed in the first two columns of the table. The threshold ε chosen from the DOS plot and the actual number of eigenvalues above the threshold for each

Table 1: Approximate Rank Estimation of various matrices

Matrices (Applications)	Size	Threshold	Eigencount	KPM		Lanczos		SVD
		ε	above ε	r_ε	Time (sec)	r_ε	Time (sec)	time
lpi_ceria3d (linear programming)	3576	28.19	78	74.69	10.8	78.74	8.7	449.3 secs
Erdos992 (collaboration network)	6100	3.39	750	747.68	4.47	750.84	11.3	876.2 secs
deter3 (linear programming)	7047	10.01	591	592.59	5.78	590.72	27.8	1.3 hrs
dw4096 (electromagnetics prblm.)	8192	79.13	512	515.21	4.93	512.46	13.9	21.3 mins
California (web search)	9664	11.48	350	352.05	3.72	350.97	16.8	18.7 mins
FA (Pajek network graph)	10617	0.51	469	468.34	12.56	473.4	28.4	1.5 hrs
qpband (optimization)	20000	0.76	15000	14989.9	9.34	15003.8	40.6	2.9 hrs

matrices are listed in the table. The corresponding approximate ranks estimated by the KPM and the Lanczos methods respectively using $m = 100$ and $n_v = 30$ are listed in the table, along with the time taken by the algorithms (the ranks and the timings are averaged over 10 trials). In addition, we also list the time taken to compute only the top 2000 singular values of each matrices (computed using `svds` function matlab which relies on ARPACK) in order to illustrate the computational gain of the proposed algorithms versus a good implementation of a partial SVD algorithm.

We observe that the Lanczos approximation method is slightly more expensive than the Chebyshev polynomial method. This is due to the additional orthogonalization cost of the Lanczos algorithm (see the computational cost section). However, the Lanczos method gives more accurate and stable results compared to KPM for the same degree and number of vectors.

In the following sections, we demonstrate the performance of the proposed rank estimation methods in different applications.

6.4 Matérn covariance matrices for grids

Matérn covariance functions are widely used in statistical analysis, for example in Machine Learning [51]. Covariance matrices of grids formed using Matérn covariance functions are popular in applications [13]. The objective of this experiment is to show how the rank estimation methods can be used to check whether such matrices are numerically low rank, and estimate their approximate ranks. For this, we shall consider such Matérn covariance matrices for 1D and 2D grids[2] to illustrate the performance of our rank estimation techniques.

The first covariance kernel matrix corresponds to a 1D regular grid with dimension 2048 [13]. The matrix is a 2048×2048 PSD matrix. We employ the two rank estimators on this matrix, which use the spectral density approach and hence we get to know whether the matrix is numerically low rank. The approximate rank estimated by KPM using a Chebyshev polynomial of degree 50 and 30 samples was 16.75. The actual count, i.e., actual number of eigenvalues above the threshold found by KPM, is 16. The approximate rank estimated by Lanczos approximation with degree 50 and 30 samples was 15.80.

Next, we consider a second covariance kernel matrix, that is from a 2D regular grid with dimension 64×64, resulting in a covariance matrix of size 4096×4096. The approximate rank estimated by KPM using a Chebyshev polynomial of degree 50 and 30 samples was 72.71, and by Lanczos approximation with same m and n_v was 72.90. The exact eigen-count above the threshold is 72. Hence, these covariance matrices are indeed numerically low rank, and a low rank approximations of the matrices can be used in applications.

In the next section, we shall consider two interesting applications for the rank estimation methods, particularly for the threshold selection method. We illustrate how the rank estimation methods can be used in these applications.

---

[2]The codes to generate these matrices were obtained from `http://press3.mcs.anl.gov/scala-gauss/`.

6.5 Eigenfaces and and Video Foreground Detection

It is well known that face images and video images lie in a low-dimensional linear subspace and the low rank approximation methods are widely used in applications such as face recognition and video foreground detection. *Eigenfaces* [62], a method essentially based on Principal Component Analysis (PCA), is a popular method used for face recognition. A similar approach based on PCA is used for background subtraction in surveillance videos [42]. To apply these PCA based techniques, we need to have an idea of the dimension of the smaller subspace. Here, we demonstrate how the threshold selection method and the rank estimation techniques introduced in the main paper can be used in face recognition and video foreground detection. We use the randomized algorithm discussed in [29] to compute the principle components in both experiments.

Eigenfaces It has been shown that the face images from the same person lie in a low-dimensional subspace of dimension at most 9 [4], based on the available degrees of freedom. However, these images are distorted due to cast shadows, specular reflections and saturations [71] and the image matrices are numerically low rank. Hence, it is not readily known how to selected the threshold for the singular values and choose the rank. Here, we demonstrate how our proposed method automatically selects an appropriate threshold for the the singular values and computes the approximate rank.

For this experiment, we used the images of the first two persons in the extended Yale face database B [23, 41]. There are 65 images of size 192×168 per person taken under different illumination conditions. Therefore, the input matrix is of size 65×32256 (by vectorizing the images). Article [71] discusses a method to estimate the unknown dimension of lower subspace based on the eigenvalue spectrum of the matrix and compares the performance of various robust PCA techniques for face recovery. Here, we illustrate how the Chebyshev polynomial KPM can be used to estimate the dimension of the smaller subspace in order to exploit an algorithm based on randomization [29] to recover the faces by the eigenfaces method. We can also use the algorithm presented in [22]. The randomized algorithms require the rank k as input, i.e., the size of the random matrix to be chosen to multiply the matrix which can be computed by the proposed technique. For the method in [22], we need the threshold ε as input, which can be computed by the proposed threshold selection technique. By using either of these algorithms, we can avoid computing the full SVD.

Figures 7(A) and (B) give the DOS and the approximate rank for the image matrix of the first person, respectively, estimated by KPM using a degree $m = 40$, and a number of sample vectors $n_v = 30$. The approximate rank estimated over 30 sample vectors was equal to 4.34. There are 4 eigenvalues above the threshold estimated using the DOS plot. The bottom image in figure 7(E) is the recovery of the top image, which is the first image of the first person, obtained by the eigenfaces method using randomized algorithm with 6 (rank 4+2 oversampling) random samples. Similarly, figure 7(C) and (D) give the DOS and the approximate rank for the image matrix of the second person, respectively. The approximate rank estimated was 2.07 while the exact number of eigenvalues is 2. The bottom image in figure 7(F) is the recovery of the top image, which is the first image of the second person, obtained using randomized algorithm with 4 (rank 2+2 oversampling) random samples. We can see that the recovered images are fairly good, indicating that the ranks estimated are fairly accurate. Better recovery can be achieved by robust PCA techniques [71, 10].

Video Foreground Detection Next, we consider the background subtraction problem in surveillance videos, where PCA is used to separate the foreground information from the background noise. We consider the two videos: "Lobby in an office building with switching on/off lights" and "Shopping center" available from `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`. These videos are used in some of the articles that discuss robust PCA methods [10, 71]. Here we illustrate how the rank estimation method can be used to obtain an appropriate value for the number of
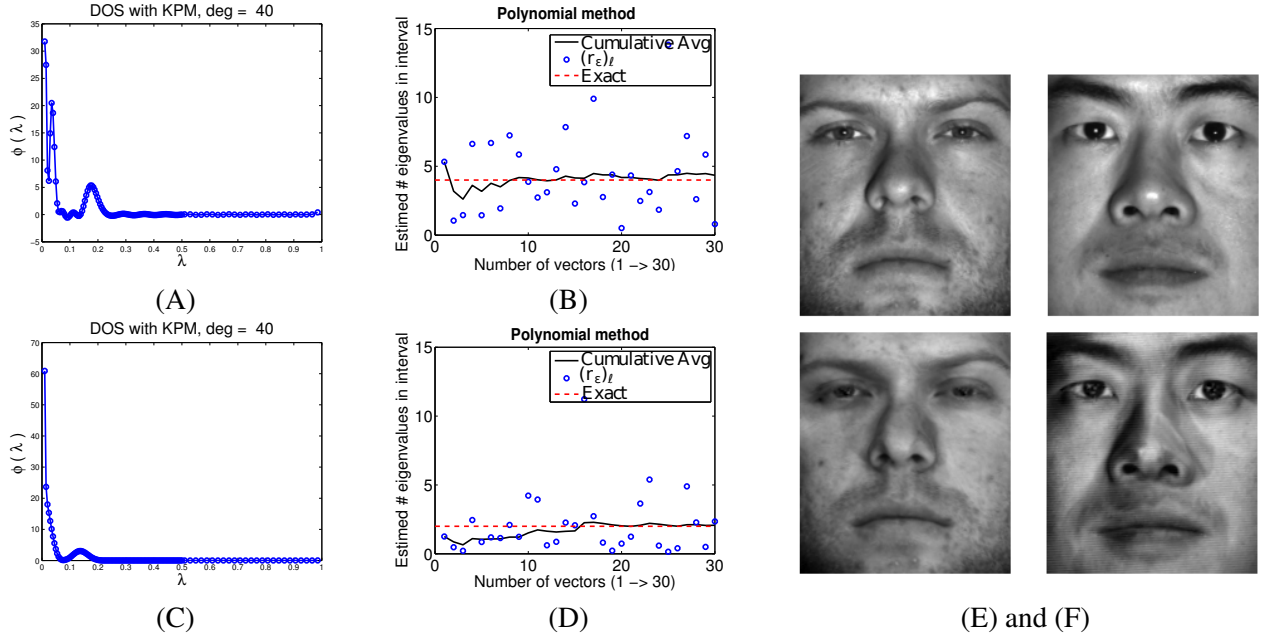
Figure 7: (A) DOS and (B) the approximate rank estimated for the image matrix of the first person. (C) DOS and (D) the approximate rank estimated for the image matrix of the second person. (E) Recovery of the first image of the first person by randomized algorithm with 6 random samples and (F) recovery of the first image of the second person by randomized algorithm with 4 random samples.
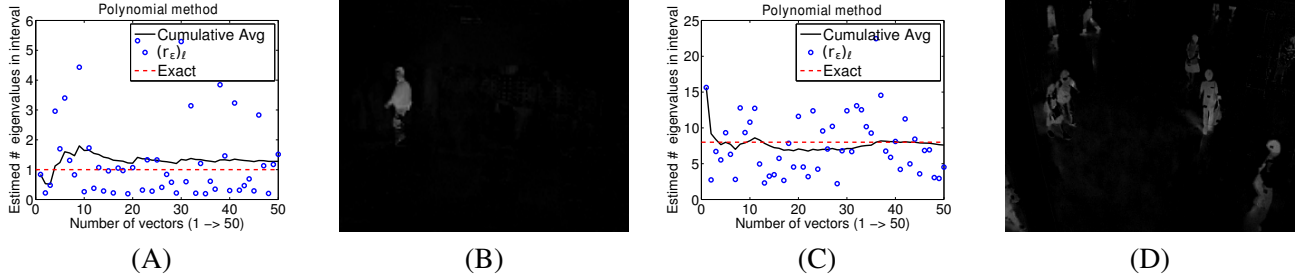

Figure 8: The approximate ranks estimation by KPM for the two video datasets and background subtraction by the randomized algorithm for two sample images.

principle components to be used for background subtraction. We use the randomized algorithm [29] to perform background subtraction.

The first video contains 1546 frames from 'SwitchLight1000.bmp' to 'SwitchLight2545.bmp' each of size 160×128. So the size of the data matrix is 1546×20480. The performance of KPM for estimating the number of principle components to be used for background subtraction of this video data is shown in figure 8. Figure 8(A) gives the approximate rank estimated by KPM for the office building video data matrix. The average approximate ranks estimated was equal to 1.28. The video has very little activities, with one or two persons moving in and out in a few of the frames. Therefore, there is only one eigenvalue which is very high compared to the rest and an approximate rank of one was estimated. The image in figure 8(B) is the foreground detected by subtracting the background estimated by using the randomized algorithm with 3 random samples, i.e, rank 1+2 oversampling.

The second video is from a shopping mall and contains more activities with many people moving

in and out of the frames, constantly. The video contains 1286 frames from 'ShoppingMall1001.bmp' to 'ShoppingMall2286.bmp' with resolution is 320×256. Therefore, the data matrix is of size 1286×81920. Figure 8(C) gives the approximate rank estimated by the Kernel polynomial method. The average approximate rank estimated was equal to 7.60. This estimated rank is higher than in the previous example and this can be attributed to the larger level of activity in the video. The image in figure 8(D) is the foreground detected by subtracting the background estimated by using randomized algorithm with 10 random samples, i.e., rank 8+2 oversampling. We observe that, we can achieve a reasonable foreground detection using the number of principle components estimated by the rank estimation method. Similar foreground detection has been achieved using robust PCA techniques, see [71, 10] for details. Thus, these two examples illustrate how the proposed methods can be used to select an appropriate dimension (the number of principle components) for PCA and robust PCA applications, particularly where the threshold selection for the singular values is non-trivial.

7   Conclusion

We presented two inexpensive techniques to estimate the approximate ranks of large matrices, that are based on approximate spectral densities of the matrices. These techniques exploit the spectral densities in two ways. First, the spectral density curve is used to locate the gap between the relevant eigenvalues which contribute to the rank and the eigenvalues related to noise, and select a threshold ε to separate between these two sets of eigenvalues (i.e., the interval of integration). Second, the spectral density function is integrated over this appropriate interval to get the approximate rank.

The ranks estimated are fairly accurate for practical purposes, especially when there is a gap separating small eigenvalues from the others. The methods require only matrix-vector products and hence are inexpensive compared to traditional methods such as QR factorization or the SVD. The lower computational cost becomes even more significant when the input matrices are sparse and/or distributively stored. Also, the proposed threshold selection method based on the DOS plots could be of independent interest in other applications such as the estimation of trace of matrix functions and the interior eigenvalue problems.

Acknowledgements

References

[1] S. V. Alyukov. Approximation of step functions in problems of mathematical modeling. *Mathematical Models and Computer Simulations*, 3(5):661–669, 2011.

[2] R. Arora, A. Cotter, K. Livescu, and N. Srebro. Stochastic optimization for PCA and PLS. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 861–868. IEEE, 2012.

[3] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8, 2011.

[4] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[6] F. Bunea, Y. She, M. H. Wegkamp, et al. Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics*, 39(2):1282–1309, 2011.

[7] E. Bura and R. D. Cook. Rank estimation in reduced-rank regression. *Journal of Multivariate Analysis*, 87(1):159–176, 2003.

[8] T. Cai, Z. Ma, and Y. Wu. Optimal estimation and rank detection for sparse spiked covariance matrices. *Probability Theory and Related Fields*, pages 1–35, 2013.

[9] G. Camba-Méndez and G. Kapetanios. Statistical tests and estimators of the rank of a matrix and their applications in econometric modelling. 2008.

[10] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[11] B. Cantrell, J. De Graaf, L. Leibowitz, F. Willwerth, G. Meurer, C. Parris, and R. Stapleton. Development of a digital array radar (DAR). In *Radar Conference, 2001. Proceedings of the 2001 IEEE*, pages 157–162. IEEE, 2001.

[12] T. F. Chan. Rank revealing QR factorizations. *Linear algebra and its applications*, 88:67–82, 1987.

[13] J. Chen, L. Wang, and M. Anitescu. A parallel tree code for computing matrix-vector products with the Matérn kernel. Technical report, Tech. report ANL/MCS-P5015-0913, Argonne National Laboratory, 2013.

[14] P. Comon and G. H. Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.

[15] R. Courant and D. Hilbert. *Methods of mathematical physics*, volume 1. CUP Archive, 1966.

[16] J. G. Cragg and S. G. Donald. On the asymptotic properties of ldu-based tests of the rank of a matrix. *Journal of the American Statistical Association*, 91(435):1301–1309, 1996.

[17] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.

[18] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

[19] E. Di Napoli, E. Polizzi, and Y. Saad. Efficient estimation of eigenvalue counts in an interval. *ArXiv preprint ArXiv:1308.4275*, 2013.

[20] X. G. Doukopoulos and G. V. Moustakides. Fast and stable subspace tracking. *Signal Processing, IEEE Transactions on*, 56(4):1452–1465, 2008.

[21] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[22] R. Frostig, C. Musco, C. Musco, and A. Sidford. Principal component projection without principal component analysis. *arXiv preprint arXiv:1602.06872*, 2016.

[23] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[24] G. Golub, V. Klema, and G. W. Stewart. Rank degeneracy and least squares problems. Technical report, DTIC Document, 1976.

[25] G. H. Golub and G. Meurant. Matrices, moments, and quadrature. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1993*, volume 303, pages 105–1–6. Pitman, Research Notes in Mathematics, 1994.

[26] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations, 4th edition*. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013.

[28] J. P. Haldar and D. Hernando. Rank-constrained solutions to linear matrix equations using power factorization. *Signal Processing Letters, IEEE*, 16(7):584–587, 2009.

[29] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[30] P. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Society for Industrial and Applied Mathematics, 1998.

[31] P. C. Hansen, V. Pereyra, and G. Scherer. *Least squares data fitting with applications*. JHU Press, 2012.

[32] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

[33] L. O. Jay, H. Kim, Y. Saad, and J. R. Chelikowsky. Electronic structure calculations for plane-wave codes without diagonalization. *Computer physics communications*, 118(1):21–30, 1999.

[34] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[35] C. Julià, A. D. Sappa, F. Lumbreras, J. Serrat, and A. López. Rank estimation in missing data matrix problems. *Journal of Mathematical Imaging and Vision*, 39(2):140–160, 2011.

[36] N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, 1997.

[37] I. Koch and K. Naito. Dimension selection for feature selection and dimension reduction with principal and independent component analysis. *Neural computation*, 19(2):513–545, 2007.

[38] S. Kritchman and B. Nadler. Determining the number of components in a factor model from limited noisy data. *Chemometrics and Intelligent Laboratory Systems*, 94(1):19 – 32, 2008.

[39] S. Kritchman and B. Nadler. Non-parametric detection of the number of signals: Hypothesis testing and random matrix theory. *IEEE Transactions on Signal Processing*, 57(10):3930–3941, Oct 2009.

[40] C. Lanczos. Applied analysis. 1956, 1956.

[41] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005.

[42] L. Li, W. Huang, I.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *Image Processing, IEEE Transactions on*, 13(11):1459–1472, 2004.

[43] L. Lin, Y. Saad, and C. Yang. Approximating Spectral Densities of Large Matrices. *SIAM Review*, 58(1):34–65, 2016.

[44] G. Liu and S. Yan. Active subspace: Toward scalable low-rank learning. *Neural computation*, 24(12):3371–3394, 2012.

[45] I. Markovsky. *Low rank approximation: algorithms, implementation, applications*. Springer Science & Business Media, 2011.

[46] P.-G. Martinsson, V. Rockhlin, and M. Tygert. A randomized algorithm for the approximation of matrices. Technical report, DTIC Document, 2006.

[47] A. B. Owen and P. O. Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3(2):564–594, 2009.

[48] P. Parker, P. J. Wolfe, and V. Tarokh. A signal processing application of randomized low-rank approximations. In *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, pages 345–350. IEEE, 2005.

[49] B. N. Parlett. *The symmetric eigenvalue problem*, volume 7. SIAM, 1980.

[50] P. Perry and P. Wolfe. Minimax rank estimation for subspace tracking. *Selected Topics in Signal Processing, IEEE Journal of*, 4(3):504–513, June 2010.

[51] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning series, MIT Press, 2006.

[52] G. C. Reinsel and R. P. Velu. *Multivariate reduced-rank regression*. Springer, 1998.

[53] J.-M. Robin and R. J. Smith. Tests of rank. *Econometric Theory*, 16(02):151–175, 2000.

[54] F. Roosta-Khorasani and U. Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, pages 1–26, 2014.

[55] Y. Saad. Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

[56] Y. Saad. Filtered conjugate residual-type algorithms with applications. *SIAM Journal on Matrix Analysis and Applications*, 28(3):845–870, 2006.

[57] Y. Saad. *Numerical Methods for Large Eigenvalue Problems- classics edition*. SIAM, Philadelpha, PA, 2011.

[58] Y. Saad. Analysis of subspace iteration for eigenvalue problems with evolving matrices. *SIAM Journal on Matrix Analysis and Applications*, 37(1):103–122, 2016.

[59] R. Silver and H. Röder. Densities of states of mega-dimensional Hamiltonian matrices. *International Journal of Modern Physics C*, 5(04):735–753, 1994.

[60] L. N. Trefethen. *Approximation theory and approximation practice*. Siam, 2013.

[61] I. Turek. A maximum-entropy approach to the density of states within the recursion method. *Journal of Physics C: Solid State Physics*, 21(17):3251, 1988.

[62] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[63] S. Ubaru, A. Mazumdar, and Y. Saad. Low rank approximation using error correcting coding matrices. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 702–710, 2015.

[64] S. Ubaru and Y. Saad. Fast methods for estimating the numerical rank of large matrices. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 468–477, 2016.

[65] L.-W. Wang. Calculating the density of states and optical-absorption spectra of large quantum systems by the plane-wave moments method. *Physical Review B*, 49(15):10154, 1994.

[66] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(2):387–392, 1985.

[67] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske. The kernel polynomial method. *Reviews of modern physics*, 78(1):275, 2006.

[68] J. Ye. Generalized low rank approximations of matrices. *Machine Learning*, 61(1-3):167–191, 2005.

[69] H. Zhang, Z. Lin, C. Zhang, and J. Gao. Relations among some low-rank subspace recovery models. *Neural Computation*, 27(9):1915–1950, 2015.

[70] Q. Zhang and K. M. Wong. Information theoretic criteria for the determination of the number of signals in spatially correlated noise. *Signal Processing, IEEE Transactions on*, 41(4):1652–1663, 1993.

[71] T. Zhang and G. Lerman. A novel m-estimator for robust PCA. *The Journal of Machine Learning Research*, 15(1):749–808, 2014.

A Additional Details and Experiments

In the main paper, we presented several numerical experiments and applications to illustrate the performances of the two rank estimation methods proposed. In this supplementary material, we give some additional details of the Lanczos approximation method and numerical experimental results. We also discuss some additional applications where our rank estimation methods can be useful.

A.1 Cumulative spectral density

The Lanczos approximation concept presented can be understood from another point of view. A Lanczos based approach for estimating the cumulative density of states (CDOS) is given in [43]. The approximate CDOS without applying regularization can be computed from the Lanczos approximation method as

$$\tilde{\psi}(t) = \sum_{l=1}^{n_v} \left(\sum_{k=1}^{m} (\rho_k^{(l)})^2 \delta(t - \theta_k^{(l)}) \right), \tag{23}$$

where $(\rho_k^{(l)})^2 = \sum_{j=1}^{k} (\tau_j^{(l)})^2$ and $\theta_k^{(l)}$ and $\tau_k^{(l)}$ are the eigenvalues and the first components of the eigenvectors of the tridiagonal matrix T_m for a starting vector v_l as defined in the main paper. The cumulative spectral density is a cumulative sum of the probability distribution of the eigenvalues (or spectral density) and is equivalent to a cumulative density function (CDF) which corresponds to integrating the probability density function (PDF) up to a point. The coefficient ρ_k^2 is the weighted sum of the τ_j^2s, up to the eigenvalue θ_k and it corresponds to integrating the spectral density in the interval $[\lambda_n, \theta_k]$. So, ρ_k^2 must be equal to the eigenvalue count in the interval $[\lambda_n, \theta_k]$, i.e.,

$$\rho_k^2 = \nu_{[\lambda_n, \theta_k]}.$$

Then, the approximate rank can be written as

$$r_\varepsilon \approx \frac{n}{n_v} \sum_{l=1}^{n_v} \left(1 - (\rho_k^{(l)})^2 \right) \quad k : \max\{\theta_k\} \leq \varepsilon. \tag{24}$$

Clearly, the approximate rank expressions given here in (24) and that obtained by the Lanczos approximation method in the main paper (eq. (19)) are equivalent.

A.2 Threshold ε and the gap

In the first experiment, we examine whether the threshold ε selected by the spectral density plot method discussed in the main paper is indeed located in the gap of the matrix spectrum. We consider two
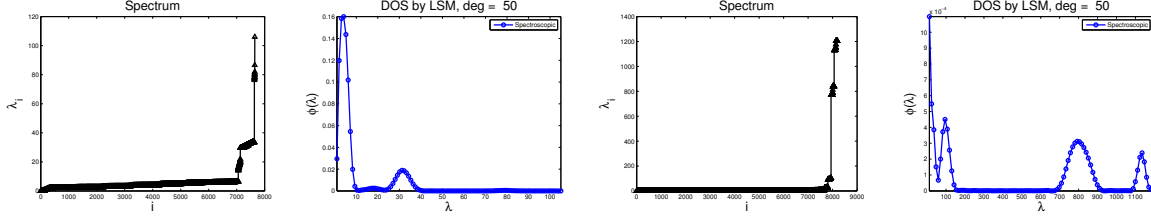
Figure 9: The spectra and the corresponding spectral densities obtained by the Lanczos method.

Table 2: Approximate Rank Estimation of Matérn covariance matrices

Type of Grid (dimension)	Matrix Size	Threshold ε	Eigencount above ε	m=50, n_v=30	
				r_ε by KPM	r_ε by Lanczos
1D regular Grid (2048×1)	2048×2048	17.60	16	16.75	15.80
1D no structure Grid (2048×1)	2048×2048	14.62	20	20.10	20.46
2D regular Grid (64×64)	4096×4096	11.30	72	72.71	72.90
2D no structure Grid (64×64)	4096×4096	11.97	70	69.20	71.23
2D deformed Grid (64×64)	4096×4096	11.48	69	68.11	69.45

matrices namely `deter3` and `dw4096` from UFL database, the matrices considered in rows 3 and 4 of Table I in the main paper, respectively. Figure 9 plots their spectra and the corresponding spectral densities obtained by the Lanczos method (LSM) using degree or the number of Lanczos steps $m = 50$. In the first spectrum (of deter3 matrix), there are around 7056 eigenvalues between 0 to 8 followed by a gap in the region 8 to 20. Ideally, the threshold ε should be in this gap. The DOS plot shows a high value between 0 to 8 indicating the presents of the large number of noise related eigenvalues and drops to zero near 10 depicting the gap. The threshold selected by the spectral density plot method was 10.01 (see Table I in the main paper) and clearly this value is in the gap.

Similarly, in the second spectrum (third plot of figure 9, of matrix dw4096), after several smaller eigenvalues, there is a gap in the spectrum from 20 to 100. The threshold selected by the spectral density plot method was 79.13 (see Table I, main paper). Thus, these two examples show us that the thresholds selected by the proposed method are indeed in the gaps of the matrix spectra. They also helps us visualize the connection between the actual matrix spectra and the corresponding spectral density plots.

A.3 Matérn covariance matrices

In the main paper, we saw two example matrices of Matérn covariance functions and how the rank estimation methods were used to show that these matrices were numerically low rank. Here, we consider some additional Matérn covariance matrices from a variety of grids.

Table 2 lists the approximate ranks estimated by KPM and the Lanczos methods for the Matérn covariance matrices of different types of 1D and 2D grids. The type of grids and their size are listed in the Table, along with the thresholds selected. The methods take only seconds to estimate the rank of these matrices (around 70-80 seconds to estimate the threshold and rank of a full matrix, of size 4096). These experiments confirm that the Matérn covariance matrices considered here are numerically low rank, and give their approximate ranks. These matrices can be well approximated by low rank matrices in applications.

In the next section, we shall consider two additional applications for the rank estimation methods, particularly for the threshold selection method. We illustrate how the rank estimation methods can be used in these applications.
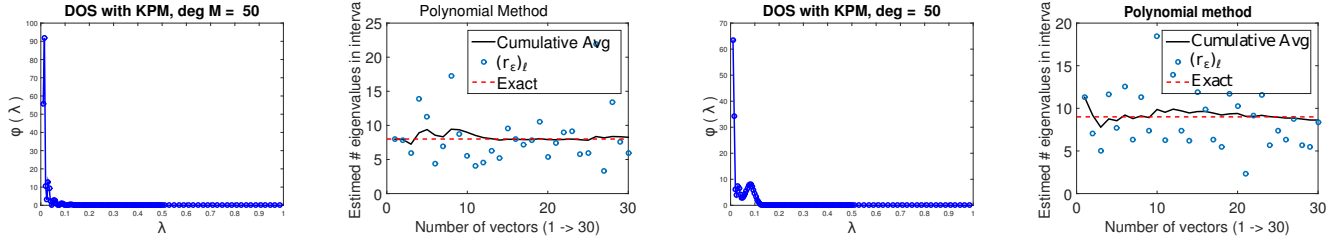
Figure 10: The DOS and the approximate ranks estimation by KPM for the two signal processing examples.

A.4 Detection of the number of signals

In this final experiment, we look at an application that comes from signal processing. Here we estimate the number of signals embedded in noisy signal data received by a collection of sensors using the rank estimator techniques presented in the paper. Such a problem arises in adaptive beamforming problem where there are many adaptive arrays, for example in large radars [11]. The goal is to detect the number of signals r in the data received by n sensors. This is equivalent to estimating the number of transmitting antennas. This estimation helps in performance evaluation metrics in adaptive beamforming [48] and in subspace tracking [50].

First, we consider $n = 1000$ element sensor array which is receiving $r = 8$ interference signals incident at angles $[-90^0, 90^0, -45^0, 45^0, 60^0, -30^0, 30^0, 0^0]$. Let $y(t)$ be the output signal from the sensors or the n-element array at time t. Then,

$$y(t) = \sum_{i=1}^{r} s_i(t)a_i + \eta(t) = As(t) + \eta(t), \tag{25}$$

where $A = [a_1(\theta_1), a_2(\theta_2), \ldots, a_r(\theta_r)]$ is an $n \times r$ matrix of interference steering vectors or simply a mixing matrix, $s(t) = [s_1(t), s_2(t), \ldots, s_r(t)]$ an $r \times 1$ signal vector and $\eta(t)$ is a white noise vector that is independent of the source and has variance σ^2. We consider the covariance matrix $C = \mathbb{E}[y(t)y(t)^\top]$ and apply the rank estimator methods to the covariance matrix to estimate the number of signals r in the received signal. The covariance matrix is generated using MATLAB R2014B function `sensorcov` available in the Phased Array System toolbox. We set the number of samples (to form the sample covariance matrix) to be 1000 which is equal to the dimension of the signals. The noise level was set at $-10DB$. Figures 10(A) and (B) show, respectively, the DOS obtained and the rank estimation using KPM with a Chebyshev polynomials of degree $m = 50$, and and the number of samples was $n_v = 30$. The approximate rank estimated over 30 sample vectors was equal to 8.26. We see that the rank estimator method has accurately estimated the number of interference signals in the received signal.

Next, we use the methods based on information theoretic criteria proposed in the literature to estimate the number of signals [66, 70]. Two such rank estimation techniques based on Minimum Description Length (MDL) and Akaike Information Criterion (AIC) were also used to estimate the rank of the covariance matrix, see [66, 70] for details. The number of signals estimate by these two methods were also equal to 8. However, these methods are expensive for large covariance matrices, since they require the estimation of all the singular values of the matrix.

In the second experiment, we have $n = 1000$ element sensor array which is receiving $r = 10$ interference signals incident at angles $[-90^0, 90^0, -45^0, 45^0, 60^0, -60^0, 15^0, -15^0, 30^0, 0^0]$. The noise level was set at $-10DB$. We use 750 samples to form the covariance matrix, which is less than the dimension of the signal. Figure 10(C) and (D) show the DOS and the approximate rank estimated for this example. The average rank estimated using KPM with Chebyshev polynomial of degree $m = 50$ and $n_v = 30$ was equal to 8.89. The actual eigen-count in the interval was 9. However, we know that the

exact number of interference signals in the received signal is $r = 10$. Interestingly, the methods based on information criteria also estimate the number of signals of this example to be 8 or 9 (value varies in different trials). This is because, there are only 9 eigenvalues of the covariance matrix which are large.

Experiments show that our rank estimation technique estimates the number of signals accurately, when the interference signal strength is high and noise power is low so that there is a gap between signal related eigenvalues and eigenvalues due to noise. This depends on factors such as the angle of incidence, the number of arrays, the number of samples in the covariance matrix, the surrounding noise, etc.