Fast methods for estimating the Numerical rank of large matrices

Shashanka Ubaru UBARU001@UMN.EDU
Yousef Saad SAAD@CS.UMN.EDU
Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN USA

Abstract

We present two *computationally inexpensive* techniques for estimating the numerical rank of a matrix, combining powerful tools from computational linear algebra. These techniques exploit three key ingredients. The first is to approximate the projector on the non-null invariant subspace of the matrix by using a polynomial filter. Two types of filters are discussed, one based on Hermite interpolation and the other based on Chebyshev expansions. The second ingredient employs stochastic trace estimators to compute the rank of this wanted eigen-projector, which yields the desired rank of the matrix. In order to obtain a good filter, it is necessary to detect a gap between the eigenvalues that correspond to noise and the relevant eigenvalues that correspond to the non-null invariant subspace. The third ingredient of the proposed approaches exploits the idea of spectral density, popular in physics, and the Lanczos spectroscopic method to locate this gap.

1. Introduction

In many machine learning, data analysis, scientific computations and signal processing applications, the high dimensional data encountered generally have intrinsically low dimensional representations. A widespread tool used in these applications to exploit this low dimensional nature of data is the Principal Component Analysis (PCA) (Jolliffe, 2002). PCA essentially takes the initial data matrix $X \in \mathbb{R}^{d\times n}$ and replaces it by a rank-k version (a lower dimensional matrix), which has the effect of capturing the intrinsic information of X. Other well known techniques such as randomized low rank approximations (Halko et al., 2011; Ubaru et al., 2015) and low rank subspace estimations (Comon & Golub, 1990; Doukopoulos & Moustakides, 2008) also exploit the ubiquitous low rank character of data. However, a difficulty with these approaches that is well-recognized in the literature is that, it is not known in advance how to select the reduced rank k. This problem is aggravated in the applications of algorithms such as online PCA (Crammer et al., 2006), stochastic approximation algorithms for PCA (Arora et al., 2012) and subspace tracking (Doukopoulos & Moustakides, 2008), where the dimension of the subspace of interest changes frequently.

The rank estimation problem also arises in many useful methods employed in fields such as machine learning for example, where the data matrix $X \in \mathbb{R}^{d\times n}$ is replaced with a factorization of the form $UV^\top$, where $U \in \mathbb{R}^{d\times k}$ and $V \in \mathbb{R}^{n\times k}$. In these methods, the original problem is solved by fixing the rank of the unknown matrix to a preselected value k (Haldar & Hernando, 2009). Similar rank estimation problems are encountered in reduced rank regression (Reinsel & Velu, 1998), when solving numerically rank deficient linear systems of equations (Hansen, 1998), and in numerical methods for eigenvalue problems that are used to compute the dominant subspace of a matrix, for e.g., subspace iteration (Saad, 2016).

In the most common situation, the rank k required as input in the above applications is typically selected in an ad-hoc way. This is because standard rank estimation methods in the existing literature rely on expensive matrix factorizations such as the QR (Chan, 1987), LDL^T or SVD (Golub & Van Loan, 2012). Other methods also assume certain asymptotic behavior such as normal responses, for the input matrices (Camba-Méndez & Kapetanios, 2008; Perry & Wolfe, 2010). Many of the rank estimation methods proposed in the literature focus on specific applications, e.g., in econometrics and statistics (Camba-Méndez & Kapetanios, 2008), statistical signal processing (Kritchman & Nadler, 2009; Perry & Wolfe, 2010), reduced-rank regression (Bura & Cook, 2003), estimating the dimension of linear systems (Hannan, 1981) and others.

Powerful and inexpensive tools from computational linear algebra can be developed to estimate the approximate ranks of large matrices. The goal of this paper is to present examples of such methods. These methods require only matrix-vector products ('matvecs') and are inexpensive compared

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

to traditional methods. In addition, they do not make any particular statistical, or asymptotic behavior assumptions on the input matrices. Since the data matrix can be approximated in a low dimensional subspace, the only assumption is that there is a set of relevant eigenvalues in the spectrum that correspond to the eigenvectors that span this low dimensional subspace, and that these are well separated from the smaller, noise-related eigenvalues.

The rank estimation techniques presented in this paper combine three key ingredients. First, a polynomial filter is used to approximate a spectral projector, the trace of which is exactly the desired rank (see sec. 3). Second, stochastic trace estimators (Hutchinson, 1990) are exploited to estimate the rank of this projector. Finally, in order to determine a good filter to use, we need to locate a gap in the spectrum and select a threshold that separates the smaller eigenvalues from the relevant ones that contribute to the rank. This paper discusses a simple method to estimate this threshold based on the spectral density function (Lin et al., 2016) of the matrix, see section 4 for details. Section 6 discusses the performance of the rank estimation techniques on matrices from various applications. First, the key concepts that are required to develop the rank estimators are discussed in the following section.

2. Key concepts

This paper aims at estimating the 'numerical' rank of a symmetric positive semi-definite (PSD) matrix A. This matrix may be a covariance matrix associated with some data X, or may just be of the form[1] $A = X^\top X$ or $XX^\top$ for the given data X, of which we seek the numerical rank.

2.1. Numerical rank

The *numerical rank* or *approximate rank* of a $d \times n$ matrix X, with respect to a positive tolerance ε is defined as

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

where $\|.\|_2$ refers to the 2-norm or spectral norm. This is a standard definition that can be found, for example, in (Golub & Van Loan, 2012; Golub et al., 1976; Hansen, 1998). Here, the matrix X is assumed to be a perturbed version of some original matrix of rank $r_\varepsilon < \{d, n\}$. Although the perturbed matrix is likely to have full rank, it can usually be well approximated by a rank-r_ε matrix. The singular values of a matrix X with approximate rank r_ε satisfy

$$\sigma_{r_\varepsilon} > \varepsilon \ge \sigma_{r_\varepsilon+1}. \quad (2)$$

It is important to note that the notion of numerical rank r_ε is useful only when there is a well-defined gap between σ_{r_ε} and $\sigma_{r_\varepsilon+1}$ (Hansen, 1998). The issue of determining this gap, i.e., selecting the parameter ε in the definition of ε-rank, is one of the key tasks for estimating the approximate rank. A few methods have been proposed in the signal processing literature to address this issue (Kritchman & Nadler, 2009; Perry & Wolfe, 2010). In section 4, we describe a different approach to locate the gap and choose a value for the tolerance or threshold ε based on the Lanczos spectroscopic approach.

Once the gap is identified and the threshold ε is set, the simplest idea for estimating the rank is to count the number of eigenvalues of A that are larger than ε. For this task, eigenvalue count methods can be invoked, see for e.g., (Di Napoli et al., 2013). Recently, article (Zhang et al., 2015) discussed the communication complexities for such numerical rank estimations (assuming ε is given) in the distributed settings using deterministic and randomized algorithms. The randomized algorithm discussed in (Zhang et al., 2015) is based on the same idea of counting the eigenvalues above the given threshold $\varepsilon \ge 0$, using a similar algorithm to the one in (Di Napoli et al., 2013). In this paper, we address both the issues of determining a proper threshold ε to use, and that of estimating the rank once ε is determined.

2.2. The dominant spectral projector

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric semi-positive definite matrix with eigenvalues $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$ and associated orthonormal eigenvectors $u_1, u_2, \ldots, u_n$, respectively. One of the main ideas explored in this paper is to compute the rank by estimating the trace of the eigen-projector:

$$P = \sum_{\lambda_i \,\in\, [a,\, b]} u_i u_i^\top, \quad (3)$$

where the interval $[a,\ b]$ is (implicitly or explicitly) selected so that it includes the relevant dominant eigenvalues that determine the rank. This idea of eigen-projectors is also used in the 'eigenvalue count' algorithm discussed in (Di Napoli et al., 2013). The eigenvalues of a projector are either zero or one and so the trace of P equals the number of terms in the sum (3), i.e., the number of eigenvalues $\eta_{[a,\ b]}$ in $[a,\ b]$,

$$\eta_{[a,b]} = \text{Trace}(P)\ .$$

Although P is typically not available, it can be inexpensively approximated in practice by a polynomial of A. First, we can interpret P as a step function of A given by

$$P = h(A), \text{ where } h(t) = \begin{cases} 1 & \text{if } t \in [a,\ b] \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

Next, this step function $h(t)$ can be approximated by a polynomial of degree m, say $\psi_m(t)$ and the projector P

[1]We will see that this matrix-matrix product need not be formed explicitly.

is expressed as $P \approx \psi_m(A)$. In this form, it becomes possible to estimate the trace of P by a stochastic estimator (Hutchinson, 1990). The issue of selecting a, b will be addressed in sections 3 and 4.

2.3. The trace estimator

Hutchinson's unbiased estimator (Hutchinson, 1990) uses only matrix-vector products to approximate the trace of a generic matrix D. The method estimates the trace $\mathtt{tr(D)}$ by first generating random vectors $v_l, l = 1, .., \mathrm{n_v}$ with equally probable entries ± 1, and then computing the average over the samples of $v_l^\top D v_l$,

$$\text{Trace}(D) \approx \frac{1}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} v_l^\top D v_l. \tag{5}$$

It is known that any random vectors v_l with mean of entries equal to zero and unit 2-norm can be used (Avron & Toledo, 2011). Thus, substituting D with $\psi_m(A)$ in (5), will yield the following estimate of the trace of P:

$$\text{Trace}(P) \approx \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} v_l^\top \psi_m(A) v_l. \tag{6}$$

Before we describe the types of polynomials $\psi_m(t)$ that we propose to use, it is important to note that the above expression does not require to form the matrix $\psi_m(A)$. All that is needed is to efficiently compute the vectors $\psi_m(A)v_l$ for any v_l, and this can be accomplished by a sequence of matrix-by-vector products with the matrix A (see supplementary material for additional details).

3. Polynomial filters

In our approach, the projector $P = h(A)$ in (4) is approximated by $\psi_m(A)$, where $\psi_m(t)$ is a 'filter' polynomial. In practice, we only need $\psi_m(t)$ to transform the larger relevant eigenvalues into a value close to one and the smaller eigenvalues to a value close to zero. We first consider a simple filter based on Hermite interpolation (sec. 3.1), which has a number of advantages relative to the more common Chebyshev filter, which is described in section. 3.2.

3.1. The McWeeny filter

The McWeeny transform (McWeeny, 1960) has been used in solid-state physics to develop 'linear-scaling' methods (Li et al., 1993). It starts by scaling and shifting the matrix so that its eigenvalues are in the interval $[0,\ 1]$. This can be achieved by simply defining $B = A/\lambda_1$, where the largest eigenvalue λ_1 can be inexpensively computed with a few steps of the Lanczos algorithm (Golub & Van Loan, 2012).

The McWeeny filter is a polynomial of cubic order whose goal is to push larger eigenvalues of B closer to one and smaller eigenvalues closer to zero. In fact it is simply a Hermite interpolation of a function that has the values $y_0 = 0, y_1 = 1$ at $x_0 = 0, x_1 = 1$ and derivatives equal to zero at both points. This leads to

$$\psi(t) = 3t^2 - 2t^3. \tag{7}$$

So a basic method for estimating the rank without using any parameter is to first calculate λ_1 and define $B = A/\lambda_1$, then estimate the trace of $\psi(B)$ using Hutchinson's estimator. Here, the projector is approximated by

$$P \approx 3B^2 - 2B^3.$$

Clearly, a degree 3 filter of this type is likely to give only a very rough estimate of the rank. We can extend the McWeeny filter to any degree by using Hermite interpolation at the points 0 and 1. In fact it is important to vary the degree of smoothness at zero and at one. It may be more important to have a higher degree of matching at point one since we wish the values of the filter to be very close to one for the larger singular values.

Figure 1 shows four different filters using various degrees of matching at zero and one. These extended McWeeny filters have been studied in a different context (Saad, 2006). A systematic way of generating them is through interpolation in the Hermite sense, using two integer parameters m_0, m_1 that define the degree of matching or smoothness at two points τ_0 and τ_1 respectively. In the following, we denote by $\Theta_{[m_0,m_1]}$ the interpolating (Hermite) polynomial that satisfies the following conditions:

$$\Theta_{[m_0,m_1]}(\tau_0) = 0; \Theta'_{[m_0,m_1]}(\tau_0) = \cdots = \Theta^{(m_0-1)}_{[m_0,m_1]}(\tau_0) = 0$$
$$\Theta_{[m_0,m_1]}(\tau_1) = 1; \Theta'_{[m_0,m_1]}(\tau_1) = \cdots = \Theta^{(m_1-1)}_{[m_0,m_1]}(\tau_1) = 0.$$

Thus, $\Theta_{[m_0,m_1]}$ has degree $m_0 + m_1 - 1$ and the two parameters m_0 and m_1 define the degree of smoothness at the points τ_0 and τ_1 respectively. The polynomials $\Theta_{[m_0,m_1]}$ can be easily determined by standard finite difference tables. The paper (Saad, 2006) also gives a closed form expression for $\Theta_{[m_0,m_1]}$ when $\tau_0 = -1$ and $\tau_1 = 1$:

$$\Theta_{[m_0,m_1]} = \frac{\int_{-1}^{t} (1-s)^{m_1-1}(1+s)^{m_0-1}\, ds}{\int_{-1}^{1} (1-s)^{m_1-1}(1+s)^{m_0-1}\, ds}. \tag{8}$$

Furthermore, when $m_0 + m_1 > 2$ (at least 3 conditions imposed), the function has an inflexion point at :

$$t = \frac{m_0 - m_1}{m_0 + m_1 - 2}.$$

When translated back to the interval $[0,\ 1]$ this point becomes $(t+1)/2 = (m_0 - 1)/(m_0 + m_1 - 2)$.

Let us consider for example the choice: $m_0 = 2, m_1 = 14$. The inflexion point is at $1/14$ and this can be viewed as a

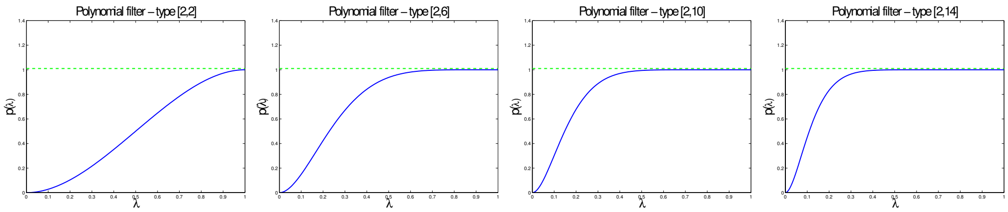

Figure 1. Four different polynomial filters based on the McWeeny idea (first curve corresponds to McWeeny filter).

cut-off value. Recall that all singular values are scaled by σ_1 so they are all ≤ 1. The filter will take all eigenvalues $\lambda_i = (\sigma_i/\sigma_1)^2$ that are larger than $1/14$, and move them close to one. All other eigenvalues will be moved close to zero. In this case, the eigenvalues larger than 1/14 are deemed to contribute to the rank – and these are termed 'relevant' in the sequel. Looking at the plot indicates that when the relevant eigenvalues are in the interval $[0.35,\ 1]$, we will get an accurate approximation of the rank by using this simple polynomial of degree 15. A good accuracy will be also obtained if all relevant eigenvalues are in the interval $[0.25,\ 1]$. The approximation will become poorer if there are eigenvalues below 0.2 and closer to the inflection point. These cases can be handled by a higher degree polynomial. Thus, once the threshold ε is computed, say by the method in section 4, an appropriate degree for the polynomial can be easily selected. However, this may lead to a very large degree for smaller ε value.

So far we have looked at polynomials $\Theta_{[m_0,m_1]}$ based on the interpolation knots: $\tau_0 = 0, \tau_1 = 1$. A look at the curves reveals that to the right of $t = 1$ the polynomial stays close to one in an interval that extends well beyond the value $t = 1$. Therefore, we can take $\tau_0 = 0$, and $\tau_1 < 1$ to reduce the degree m_1. In fact we can move τ_1 back toward 0.5 as far as possible before $p(1)$ departs from 1 by a certain threshold. A little analysis shows that τ_1 must be larger than 0.5. Thus, we can use dichotomy to find the best value of τ_1 and the degree m_1 based on the ε value computed (details in sec. 4 and the supplementary material).

3.2. Chebyshev filters

Chebyshev polynomials are commonly used to expand the step function h, i.e., $h(t)$ is approximately expanded as :

$$h(t) \approx \sum_{k=0}^{m} \gamma_k T_k(t), \tag{9}$$

where each T_k is the k-degree Chebyshev polynomial of the first kind, formally defined as $T_k(t) = \cos(k\cos^{-1}(t))$. Since Chebyshev polynomials are based on the interval $[-1,\ 1]$ we will assume first that A has eigenvalues between -1 and 1. Let a, b such that $-1 \leq a < b \leq 1$. The expansion coefficients γ_k for the polynomial to approximate a step function $h(t)$, which takes value 1 in $[a,\ b]$ and 0 elsewhere, are known:

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

Once the γ_k's are known, the desired Chebyshev expansion of the projector P will be given by: $P \approx \psi_m(A) = \sum_{k=0}^{m} \gamma_k T_k(A)$.

The approximate matrix rank r_ε can be determined by setting the interval $[a, b] = [\varepsilon, \lambda_1]$. As a result, the approximate rank of a matrix A using the Chebyshev polynomial filtering method is estimated by:

$$r_\varepsilon = \eta_{[\varepsilon,\lambda_1]} \approx \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} \left[\sum_{k=0}^{m} \gamma_k (v_l)^\top T_k(A) v_l \right]. \tag{10}$$

It remains to determine the threshold ε and a method for this purpose will be be described in the next section. Details on the practicalities of Chebyshev polynomial approximation can be found in the supplementary material.

4. Threshold selection

The method we described so far requires a threshold parameter ε that separates the small eigenvalues, those assumed to be perturbations of the zero eigenvalue, from the relevant larger eigenvalues that contribute to the rank. We now describe a method to select ε based on the Lanczos spectroscopic method (LSM) and the spectral density. Related to this is the need to select appropriate polynomial degrees for the extended McWeeny and Chebyshev filters. This is discussed at the end of the section.

4.1. LSM and spectral density

The Lanczos spectroscopic approach (Lanczos, 1956) consists of representing the matrix spectrum as a collection of frequencies and computes these frequencies using Fourier analysis. Suppose the eigenvalues of A are in the interval $[-1,\ 1]$, then LSM considers samples of the following continuous function:

$$f(t) = \sum_{j=1}^{n} \beta_j^2 \cos(\theta_j t), \tag{11}$$

where θ_j's are related to the eigenvalue of A by $\theta_j = \cos^{-1}\lambda_j$, and β_j's are scalars whose values depend on the

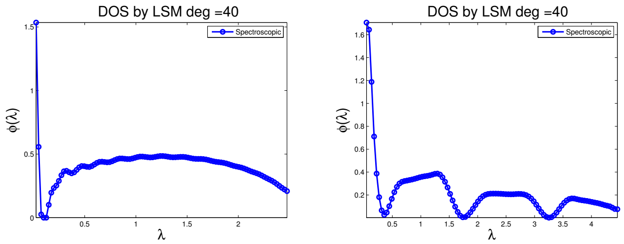

Figure 2. Typical spectral density plots by LSM for a low rank (left) and a numerically low rank (right) matrices.

function $f(t)$ considered. The above function is sampled at $t = 0, 1, \dots, m$. Then, taking the Fourier transform of $f(t)$ reveals the spectral information of A, i.e., with sufficient number of samples, the Fourier transform of the sampled function will have peaks near $\cos^{-1}\lambda_j,\ j = 1, \dots, n$. Recently, Lin et. al (Lin et al., 2016) showed that an approximate spectral density can be obtained from this method.

The *spectral density* or the *Density of States* (DOS) of a real symmetric matrix (popular in solid-state physics) is a probability density distribution that measures the likelihood of finding eigenvalues of the matrix near a point on a real line. Given an $n \times n$ symmetric matrix A, the Density of States (DOS) is defined as

$$\phi(t) = \frac{1}{n}\sum_{j=1}^{n}\delta(t - \lambda_j), \tag{12}$$

where δ is the Dirac δ-function or Dirac distribution, and the λ_j's are the eigenvalues of A. Efficient algorithms for computing the DOS without computing all the eigenvalues of the matrix have been developed in the literature (Wang, 1994; Lin et al., 2016).

Back to the spectroscopic method, since the λ_j's are not available, $f(t)$ in (11) cannot be computed directly. However, we observe that $f(t)$ is closely related to the Chebyshev polynomials. In particular, $m+1$ uniform samples of $f(t)$, say $f(0), f(1), \dots, f(m)$ can be computed as the average of

$$v_l^\top v_l, v_l^\top T_1(A)v_l, \dots, v_l^\top T_m(A)v_l,$$

where $v_l, l = 1, \dots, \mathrm{n_v}$ are random starting vectors. For the DOS, we just need the mean of β_j's to be one. This fact helps us compute $f(t)$ as the average of $v_l^\top T_k(A)v_l$, see Theorem 3.1 in (Lin et al., 2016). The discrete cosine transform of $f(t)$ is given by

$$F(p) = \frac{1}{2}(f(0) + (-1)^p f(m)) + \sum_{k=1}^{m-1} f(k)\cos(\frac{kp\pi}{m}),$$

for $p = 0, \dots, m$. An approximate spectral density $\phi(t)$ can be obtained from $F(p)$ using an interpolation procedure (Lin et al., 2016). Next, we describe a method to estimate the threshold ε based on the plot of spectral density $\phi(t)$ obtained by the LSM.

4.2. Analyzing the spectral density plots

In order to describe the threshold selection method, we will consider two matrices with the following spectral distributions. The first matrix has an exact low rank, and its DOS plot will serve as a motivation for the proposed technique for selecting the threshold. As an example we consider 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. The second matrix is a typical numerically rank deficient matrix (the kind of matrices observed in the applications) which has a large number of eigenvalues related to noise that are close to zero and a number of larger relevant eigenvalues (forming few clusters), that contribute to the approximate rank. The approximate spectral density plots of these two matrices obtained by LSM using Chebyshev polynomials of degree $m = 40$ are plotted in figure 2.

In the left plot, since the matrix has a large number of eigenvalues equal to zero, the plot has a high value at zero, and then drops quickly to almost a zero value, representing the region where there are no eigenvalues. At 0.2, the plot increases again due to the presence of new eigenvalues. So, a gap in the matrix spectrum will correspond to a sharp drop to zero or a valley in the spectral density plot of the matrix. We observe a similar behavior in the numerically low rank matrix case (right plot of fig. 2) as well. The spectral density has a high value near zero and displays a fast decrease due to the gap between the noise related eigenvalues and the relevant eigenvalues. The curve increases again due the presence of larger relevant eigenvalue clusters.

The rank k of the first matrix can be estimated by counting the eigenvalues in the interval $[\varepsilon,\ \lambda_1] = [0.2, 2.5]$. The value $\lambda_1 = 2.5$ is estimated as discussed earlier. The threshold value $\varepsilon = 0.2$, which is a cutoff point between zero eigenvalues and relevant ones is located at the point where the spectral density curve ceases to decrease, in the valley corresponding to the gap. That is, the point is a local minimum of the function $\phi(t)$. Thus, it can selected as the left most value of t for which the derivative of $\phi(t)$ becomes zero, i.e., ε can be defined as:

$$\varepsilon = \min\{t : \phi'(t) = 0, \lambda_n \le t \le \lambda_1\}. \tag{13}$$

Since a numerically rank deficient matrix is a perturbed version of some low rank matrix, the same idea based on the spectral density plot can be employed to determine its numerical rank. The threshold is now a cutoff point between noise related eigenvalues and relevant ones, and this point must be in the valley corresponding to the first local minimum in the DOS plot. Thus, equation (13) can be used to estimate the threshold ε as well. In our experiments, we observe that the spectral density plots obtained by LSM capture the local minima (the gaps) of DOS quite well. A more practical version of formula (13) is the following :

$$\varepsilon = \min\{t : \phi'(t) \geq tol, \lambda_n \leq t \leq \lambda_1\}. \tag{14}$$

We found that $tol = -0.01$ works well in practice.

4.3. Choosing appropriate polynomial degrees

Once the separation point ε is found, we can select appropriate degrees and type of Θ, i.e., m_0, and m_1 in the case of extended McWeeny filters. For the Hermite filters, we always select $m_0 = 2$ for a number of reasons. We found that adding in smoothness at τ_0 does not help. Then in order for the inflexion point to be just around the gap center ε, we start by taking $\tau_1 = 1$ and $m_1 = \lceil 1/\varepsilon \rceil$ and then use dichotomy to choose an appropriate τ_1 (between $(0.5, 1]$) and m_1 (as low as possible) such that the inflexion is around ε and $\psi(1)$ is close to 1.

For the Chebyshev filters, the cut-off value ε dictates the choice of the interval $[a, b]$ to use, but not the degree. The degree should be selected to reflect the sharpness of the filter. For example, if we have an interval $[-1,\ \varepsilon_0]$ which should contain small eigenvalues and a second interval $[\varepsilon_1,\ 1]$ which contains relevant eigenvalues, then we will select the cut-off value $\varepsilon = (\varepsilon_0 + \varepsilon_1)/2$ and then the degree m should be such that

$$\max_{t\,\in\,[-1,\ \varepsilon_0]} |\psi_m(t)| \leq \delta; \qquad \max_{t\,\in\,[\varepsilon_1,\ 1]} |1 - \psi_m(t)| \leq \delta,$$

where $\delta \geq 0$ is a small number. When ε_1 and ε_0 are close, this condition will require a high degree polynomial. We choose $\varepsilon_0 = \varepsilon - \delta$ and $\varepsilon_1 = \varepsilon + \delta$ in our experiments.

5. Algorithm and analysis

This section, describes the proposed algorithms and their computational costs. Convergence analysis for the methods is also briefly discussed at the end of the section.

Algorithm 1 describes our approach for estimating the approximate rank r_ε by the two polynomial filtering methods discussed earlier.

Algorithm 1 Numerical rank estimation by polynomial filtering

Input: An $n \times n$ symmetric PSD matrix A, λ_1 and λ_n of A, and number $\mathrm{n_v}$ of sample vectors to be used.
Output: The numerical 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 to $B = A/\lambda_1$, choose degree m for DOS and form the matvecs

$$B^k v_l : l = 1, \ldots, \mathrm{n_v}, k = 0, \ldots, m.$$

3. Form the scalars $v_l^\top T_k(B) v_l$ using the above matvecs and obtain the DOS $\tilde{\phi}(t)$ by LSM.
4. Estimate the threshold ε from $\tilde{\phi}(t)$ using eq. (14).
5. McWeeny filter: Estimate m_1 and τ_1 from ε. Compute $\Theta_{[m_0, m_1]} v_l$ using the above matvecs (compute additional matvecs if required). Estimate the numerical rank r_ε using eq. (6).
Chebyshev filter: Compute the degree m and estimate the coefficients γ_k for the interval $[\varepsilon, \lambda_1]$. Compute the numerical rank r_ε using (10) and the above matvecs.

Computational cost. The core of the computation in the two rank estimation methods is the matrix vector product of the form $T_k(A) v_l$ or in general $A^k v_l$ for $l = 1, \ldots, \mathrm{n_v}, k = 0, \ldots, m$ (step 3). Note that no matrix-matrix products or factorizations are required. In addition, the matrix vector products $A^k v_l$ computed during the estimation of the threshold, for the spectral density, can be saved and reused for the rank estimation, and so the related matrix-by-vector products are computed only once. All remaining steps of the algorithm are essentially based on these 'matvec' operations.

For an $n \times n$ dense symmetric PSD matrix, the computational cost of Algorithm 1 is $O(n^2 m \mathrm{n_v})$. For a sparse matrix, the computation cost will be $O(\mathrm{nnz}(A) m \mathrm{n_v})$, where $\mathrm{nnz}(A)$ is the number of nonzero entries of A. This cost is linear in the number of nonzero entries of A for large matrices and it will be generally quite low when A is very sparse, e.g., when $\mathrm{nnz}(A) = O(n)$. These methods are very inexpensive compared to methods that require matrix factorizations such as QR or SVD.

Remark 1 *In some of the rank estimation applications, it is perhaps required to estimate the corresponding eigenpairs or the singular triplets of the matrix, after the approximate rank estimation. These can be easily computed using a Rayleigh-Ritz projection type methods, exploiting again the vectors $A^k v_l$ generated for estimating the rank.*

On the convergence. The convergence analysis of the trace estimator (5) is well documented in (Roosta-Khorasani & Ascher, 2014) for starting vectors with Rademacher (Hutchinson), Gaussian and uniform unit vec-

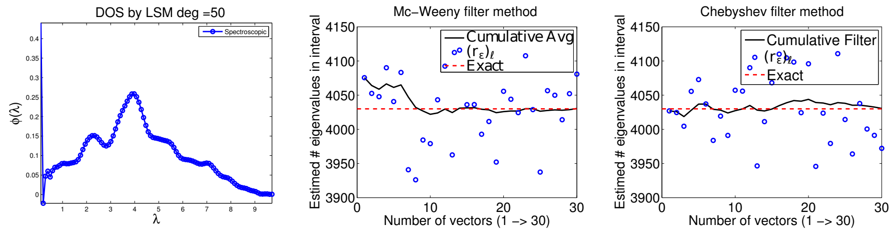

Figure 3. Left: Spectral density plot by LSM. Middle: Numerical ranks estimated by McWeeny filter method for the example `ukerbe1`. Right: Numerical ranks estimated by Chebyshev filter method.

tor probability distributions. The best known convergence rate for (5) is $O(1/\sqrt{\mathrm{n_v}})$ for Hutchinson and Gaussian distributions (see Theorem 1 and 3 in (Roosta-Khorasani & Ascher, 2014), respectively).

Theoretical analysis for approximating a step function as in (4) is not straightforward since we are approximating a discontinuous function. Convergence analysis on approximating a step function is documented in (Alyukov, 2011). A convergence rate of $O(1/m)$ can be achieved with any polynomial approximation (Alyukov, 2011). 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.

Improved theoretical results can be obtained if we first replace the step function by a piecewise linear approximation, and then employ polynomial approximation. Article (Saad, 2006) shows that uniform convergence can be achieved using Hermite polynomial approximation (as in sec. 3.1) when the filter is constructed as a spline (piecewise linear) function. For example,

$$\psi(t) = \begin{cases} 0 & : for\ t \in [0, \varepsilon_0) \\ \Theta_{[m_0.m_1]} & : for\ t \in [\varepsilon_0, \varepsilon_1) \\ 1 & : for\ t \in [\varepsilon_1, 1] \end{cases}. \qquad (15)$$

It is well known that uniform convergence can be achieved with Chebyshev polynomial approximation if the function approximated is continuous and differentiable, see Theorem 5.7 in (Mason & Handscomb, 2002). Further improvement in the convergence rate can be accomplished, if the step function is replaced by a function whose $p+1$st derivative exists, for example, $\psi(t)$ can be a shifted version of $\tanh(pt)$ function. In this case, a convergence rate of $O(1/m^p)$ can be achieved with Chebyshev polynomial approximation, see Theorem 5.14 (Mason & Handscomb, 2002). However, such complicated implementations are unnecessary in practice. The bounds achieved for both the trace estimator and the approximation of step functions discussed above are too pessimistic, since in practice we can get accurate ranks for $m \sim 50$ and $\mathrm{n_v} \sim 30$.

6. Numerical experiments

In this section, we illustrate the performance of the rank estimation techniques on matrices from various typical applications. In the first experiment, we use a $5,981 \times 5,981$ matrix named `ukerbe1` from the AG-Monien group (the matrix is a Laplacian of an undirected graph), available in the University of Florida Sparse Matrix Collection (Davis & Hu, 2011) database. The performances of the Chebyshev Polynomial filter method and the extended McWeeny filter method for estimating the numerical rank of this matrix[2] are shown in figure 3.

Figure 3 (Left) gives the spectral density plot obtained by LSM using Chebyshev polynomials of degree $m = 50$ and a number of samples $\mathrm{n_v} = 30$. Using this plot, the threshold ε estimated by the method described in section 4 was $\varepsilon = 0.169$. Figure 3 (Middle) plots the numerical ranks estimated by the McWeeny filter method with 30 sample vectors. The degrees $[m_0, m_1]$ for the Hermite polynomials estimated were $[2, 54]$. In the plot, the circles indicate the approximate ranks estimated with the ℓth sample vectors and the dark line is the cumulative (running) average of these estimated approximate rank values. The average numerical rank estimated over 30 sample vectors was equal to 4030.47. The exact number of eigenvalues above the threshold is 4030, indicated by the dotted line in the plot. Similarly, figure 3 (Right) plots the numerical ranks estimated by the Chebyshev filter method with $\mathrm{n_v} = 30$. The degree for the Chebyshev polynomials selected by the method in section 4.3 was $m = 96$. The average numerical rank estimated over 30 sample vectors is 4030.57.

Timing Experiment : Here, we provide an example to illustrate how fast these methods can be. We consider a sparse matrix of size 1.25×10^5 called `Internet` from the UFL database, with $\mathrm{nnz}(A) = 1.5 \times 10^6$. The estimation of its rank by the Chebyshev filter method took only 7.18 secs on average (over 10 trials) on a standard $3.3GHz$ Intel-i5 machine. Computing the rank of this matrix by an approximate SVD, for example using the svds/eigs function matlab which relies on ARPACK, will be exceedingly

[2]Matlab codes are available at `http://www-users.cs.umn.edu/~ubaru/codes/rank_estimation.zip`

Table 1. Numerical rank estimation of various matrices

Matrices (Applications)	Size	Threshold ε	Eigencount above ε	McWeeny Filter m_1	McWeeny Filter r_ε	McWeeny Filter time	Chebyshev Filter m	Chebyshev Filter r_ε	Chebyshev Filter time	SVD time
Erdos992 (undirected graph)	6100	3.39	748	64	747.52	1.82	106	747.68	2.45	876.2 secs
deter3 (linear programming)	7047	10.01	591	58	592.59	1.61	72	590.72	1.72	1.3 hrs
dw4096 (electromagnetics)	8192	79.13	512	62	512.42	1.81	68	512.21	1.83	1.2 hrs
California (web search)	9664	11.48	350	78	348.83	3.61	116	350.81	4.56	18.7 mins
FA (Pajek network graph)	10617	0.51	471	64	472.35	17.8	98	470.31	24.95	1.5 hrs
qpband (optimization)	20000	0.7	15000	42	15004.6	0.62	50	14997.1	0.91	2.9 hrs

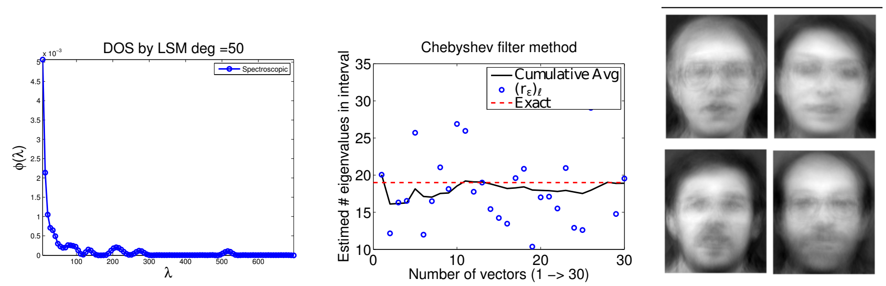

Figure 4. Left and middle: The spectroscopic plot by LSM and the numerical ranks estimated by Chebyshev filtering method for the ORL dataset. Right: Eigenfaces recovered with rank $k = 20$ using randomized SVD.

expensive. It took around 2 hours to compute 4000 singular values of the matrix on the same machine. Methods based on rank-revealing QR factorizations or the standard SVD are not even possible for this problem on a standard workstation such as the one we used.

Table 1 lists the threshold selected using spectroscopic plot, the degree of the polynomial used and the ranks estimated by the two filtering methods for a set of matrices from various applications. All matrices were obtained from the UFL database (Davis & Hu, 2011). The matrices, their applications and sizes are listed in the first two columns of the table. The threshold ε, computed from the DOS plot by LSM and the actual number of eigenvalues above the threshold for each matrices are listed in the next two columns. The degrees for the polynomials estimated, the corresponding numerical ranks computed and the average time taken (in seconds, using Matlab `cputime` function) over 10 trials, by the extended McWeeny filter and the Chebyshev filter methods using $\mathrm{n_v} = 30$ are listed in the last six columns. We observe that the McWeeny filter requires lower degree polynomials than the Chebyshev filter in all examples. Moreover, all these methods accurately estimate the numerical ranks, with fewer computations compared to traditional methods requiring QR or SVD.

Eigenfaces. It is well known that face images lie in a low-dimensional linear subspace and the low rank approximation methods are widely used in applications such as face recognition. *Eigenfaces* is a popular method used for face recognition which is based on Principal Component Analysis (PCA) (Turk & Pentland, 1991). Such PCA based techniques require the knowledge of the dimension of the smaller subspace. Here, we demonstrate how our rank estimation methods can be combined with the randomized-SVD method (Halko et al., 2011) in the application of face recognition. As an illustration, we consider the ORL face dataset obtained from the AT&T Labs Cambridge database of faces (Cambridge, 2002). There are ten different images of each of 40 distinct subjects. The size of each image is 92×112 pixels, with 256 gray levels per pixel. So, the input matrix is of size 400×10304, which is formed by vectorizing the images. The matrix is mean centered (required for eigenfaces method) and scaled.

In figure 4 (left and middle plots) the DOS and the numerical rank are plotted for the ORL image matrix, both estimated using Chebyshev polynomials of degree $m = 50$ and $\mathrm{n_v} = 30$. The numerical rank estimated over 30 sample vectors was found to be 18.90. There are 19 eigenvalues above the threshold, estimated using (14) with $tol = -0.01$. The four images (on the right) in the figure are the eigenfaces of 4 individuals recovered using rank $k = 20$ (top 20 singular vectors) computed using the randomized SVD algorithm (Halko et al., 2011).

7. Conclusion

We discussed two fast practical methods based on polynomial filtering to estimate the numerical rank of large matrices. Numerical experiments with matrices from various applications demonstrated that the ranks estimated by these methods are fairly accurate. In addition, because they require a relatively small number of matvecs, the proposed methods are quite inexpensive. As such, they can be easily incorporated into standard dimension reduction techniques such as PCA, online PCA, or the randomized SVD, that require the numerical rank of a matrix as input.

Acknowledgements

This work was supported by NSF under grant NSF/CCF-1318597.

References

Alyukov, Sergey Viktorovich. Approximation of step functions in problems of mathematical modeling. *Mathematical Models and Computer Simulations*, 3(5):661–669, 2011.

Arora, Rajkumar, Cotter, A, Livescu, Karen, and Srebro, Nathan. Stochastic optimization for PCA and PLS. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pp. 861–868. IEEE, 2012.

Avron, H. and Toledo, S. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8, 2011. doi: 10.1145/1944345.1944349. URL http://dl.acm.org/citation.cfm?id=1944349.

Bura, Efstathia and Cook, R Dennis. Rank estimation in reduced-rank regression. *Journal of Multivariate Analysis*, 87(1):159–176, 2003.

Camba-Méndez, Gonzalo and Kapetanios, George. Statistical tests and estimators of the rank of a matrix and their applications in econometric modelling. 2008.

Cambridge, AT&T Laboratories. The Database of Faces. 2002. URL http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

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

Comon, P. and Golub, G.H. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, Aug 1990. ISSN 0018-9219. doi: 10.1109/5.58320.

Crammer, Koby, Dekel, Ofer, Keshet, Joseph, Shalev-Shwartz, Shai, and Singer, Yoram. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.

Davis, Timothy A and Hu, Yifan. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

Di Napoli, Edoardo, Polizzi, Eric, and Saad, Yousef. Efficient estimation of eigenvalue counts in an interval. *ArXiv preprint ArXiv:1308.4275*, 2013.

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

Golub, Gene, Klema, Virginia, and Stewart, Gilbert W. Rank degeneracy and least squares problems. Technical report, DTIC Document, 1976.

Golub, Gene H. and Van Loan, Charles F. *Matrix computations*, volume 3. JHU Press, 2012.

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

Halko, N., Martinsson, P., and Tropp, J. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. doi: 10.1137/090771806. URL http://dx.doi.org/10.1137/090771806.

Hannan, Edward. Estimating the dimension of a linear system. *Journal of Multivariate analysis*, 11(4):459–473, 1981.

Hansen, P. *Rank-Deficient and Discrete Ill-Posed Problems*. Society for Industrial and Applied Mathematics, 1998. doi: 10.1137/1.9780898719697. URL http://epubs.siam.org/doi/abs/10.1137/1.9780898719697.

Hutchinson, Michael F. 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.

Jolliffe, Ian. *Principal component analysis*. Wiley Online Library, 2002.

Kritchman, S. and Nadler, B. 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. ISSN 1053-587X. doi: 10.1109/TSP.2009.2022897.

Lanczos, C. *Applied analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1956.

Li, X.-P., Nunes, R. W., and Vanderbilt, D. Density-matrix electronic-structure method with linear system-size scaling. *Phys. Rev. B*, 47:10891, 1993.

Lin, Lin, Saad, Yousef, and Yang, Chao. Approximating Spectral Densities of Large Matrices. *SIAM Review*, 58(1):34–65, 2016.

Mason, John C and Handscomb, David C. *Chebyshev polynomials*. CRC Press, 2002.

McWeeny, R. Some recent advances in density matrix theory. *Rev. Mod. Phys.*, 32:335–369, 1960.

Perry, Patrick O and Wolfe, Patrick J. Minimax rank estimation for subspace tracking. *Selected Topics in Signal Processing, IEEE Journal of*, 4(3):504–513, 2010.

Reinsel, Gregory C and Velu, Raja P. *Multivariate reduced-rank regression*. Springer, 1998.

Roosta-Khorasani, Farbod and Ascher, Uri. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, pp. 1–26, 2014.

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

Saad, Yousef. Analysis of subspace iteration for eigenvalue problems with evolving matrices. *SIAM Journal on Matrix Analysis and Applications*, 37(1):103–122, 2016. doi: 10.1137/141002037. URL http://dx.doi.org/10.1137/141002037.

Turk, Matthew and Pentland, Alex. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

Ubaru, Shashanka, Mazumdar, Arya, and Saad, Yousef. Low rank approximation using error correcting coding matrices. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 702–710, 2015.

Wang, Lin-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.

Zhang, Yuchen, Wainwright, Martin J, and Jordan, Michael I. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 457–465, 2015.

Supplementary Material : Fast methods for estimating the Numerical rank of large matrices

Shashanka Ubaru UBARU001@UMN.EDU
Yousef Saad SAAD@CS.UMN.EDU
Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN USA

1. Additional Details

In this supplementary material, we give additional details on the two polynomial filters discussed in the main paper. First, we give an example to illustrate how the choice of the degree in the extend McWeeny filter method affects the inflexion point and the rank estimated. Next, we discuss some details on the practical implementation of the Chebyshev polynomial filter method. In section 4, we propose an alternate method for the threshold ε selection using multiple filters. Finally, we present some additional numerical experiments and an application from signal processing where our rank estimation methods can be useful.

2. McWeeny filter: An example

In the main paper, we discussed how the cut-off or the inflexion point of the extended McWeeny filter depends on the choice of the degree m_1. We also know that a higher degree m_1 implies a better filter (captures the relevant eigenvalues better) as depicted in figure 1 of the main paper. Here we give a small toy example to illustrate the performance of the four filters from figure 1 of the main paper. We consider a random matrix X_0 of size 25×15 and rank exactly 5 to which we add noise to obtain the matrix $X = X_0 + 0.1 \times randn(m, n)$ (matlab notation used). The 15 nonzero exact singular values after division by the largest one and squaring are :

$$1.0000,\ 0.7919,\ 0.4774,\ 0.3639,\ 0.3499,$$
$$0.0098,\ 0.0083,\ \cdots,\cdots,0.0004.$$

The exact rank is 15 but there is a sharp drop after the fifth eigenvalue suggesting that the data we have is the result of perturbing a matrix of rank 5, as is indeed the case. The traces of $\psi(A)$ obtained for each of the cases shown in figure 1 of the main paper are :

$$\text{Trace}(\Theta_{2,2}(A)) = 2.9375, \quad \text{Trace}(\Theta_{2,6}(A)) = 4.4811,$$
$$\text{Trace}(\Theta_{2,10}(A)) = 4.8936, \quad \text{Trace}(\Theta_{2,14}(A)) = 4.9991.$$

The $\Theta_{2,2}$ polynomial misses the desired rank by about 2. This is because the only singular values that are moved really close to one are those close to 0.8 and higher. The cut-off here is $1/2$ and is is not adequate for this case. The second polynomial $\Theta_{2,6}$ gives slightly better result. The traces of both $\Theta_{2,10}(A)$ and $\Theta_{2,14}(A)$ are rather close to the exact value of 5. To estimate the trace, we have used the Hutchinson's trace estimator. Using just 20 random vectors, the trace of $\Theta_{2,14}(A)$ was evaluated to be 5.02 while that of $\Theta_{2,10}(A)$ was evaluated to 4.9, though these estimates show some small variance between different runs. In the main paper, we saw how the choice of m_1 affected the inflexion point and how a dichotomy method can be used to choose appropriate τ_1 and m_1 to get the ideal filter for a given threshold ε.

Though we have not analyzed the rounding error or the stability of the scheme based on Hermite polynomials, we observed that even when very high degree polynomials are used (a few hundreds) no numerical difficulties of any sort were encountered.

3. Pacticalities with Chebyshev Filters

In this section, we discuss some details on the practical implementation of the Chebyshev polynomial filter method.

Damping. When we expand discontinuous functions using Chebyshev polynomials, oscillations known as *Gibbs Phenomenon (Oscillations)* appear near the discontinuities. To suppress this behavior or limit its extent, damping multipliers are included, i.e., we replace eq. (11) in the main paper by

$$P \approx \psi_m(A) = \sum_{k=0}^{m} g_k^m \gamma_k T_k(A).$$

In effect, each γ_k is multiplied by a smoothing factor g_k^m and this factor tends to be quite small for larger k, i.e., for the highly oscillatory terms in the expansion. In the simplest case where no damping is applied we set $g_k^m = 1$. The most popular damping used in the literature is called Jackson smoothing whereby the coefficients g_k^m are given

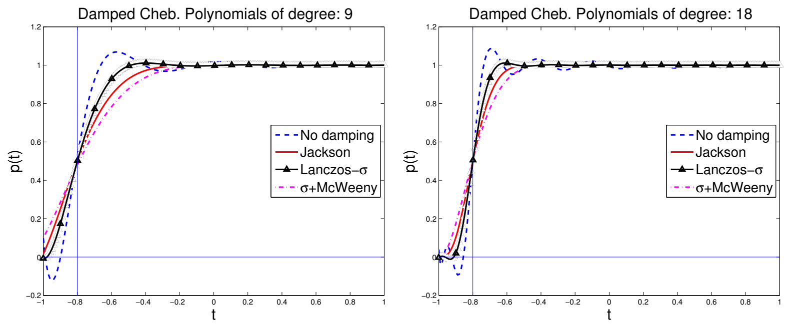

Figure 1. Four different ways of dampening Gibbs oscillations for Chebyshev approximation. All the final polynomials have the same degree (9 on the left and 18 on the right).

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 $\alpha_m = \frac{\pi}{m+2}$. Details on this expression can be found in (Di Napoli et al., 2013). Another form of smoothing proposed by Lanczos (see Chap. 4 in (Lanczos, 1956)) which is referred to as σ-smoothing can also be used. It uses simpler damping coefficients called σ factors given by:

$$\sigma_0^m = 1; \quad \sigma_k^m = \frac{\sin(k\theta_m)}{k\theta_m}, \quad k = 1, \ldots, m$$

with $\theta_m = \frac{\pi}{m+1}$.

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. Thus, 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 1. To the three forms of damping just discussed (no-damping, Jackson, σ-damping) we have added a fourth one which consists of compounding the degree 3 McWeeny filter with the Chebyshev filter. In the numerical experiments presented in the main paper and here, Lanczos σ-smoothing were used.

Recurrence. In general, since the input matrix A does not necessarily have eigenvalues between -1 and 1, we will transform A linearly into the matrix :

$$B = \frac{A - cI}{h} \qquad \text{with} \quad c = \frac{\lambda_1 + \lambda_n}{2}, \quad h = \frac{\lambda_1 - \lambda_n}{2} \tag{1}$$

whose spectrum is included in $[-1,\ 1]$. In practice, λ_1 and λ_n in the above formulas are replaced by upper and lower bounds respectively obtained from the Lanczos process, see for e.g., (Saad, 2011) for details. Note that the interval $[\varepsilon,\ \lambda_1]$ must be mapped to $[\hat{\varepsilon},\ 1]$, where $\hat{\varepsilon}$ is the linear transformation of ε using (1).

Another important practical consideration is that Chebyshev polynomials obey a three term recurrence which allows an economical computation of vectors of the form $T_k(B)v$. Indeed,

$$T_{k+1}(t) = 2tT_k(t) - T_{k-1}(t)$$

with $T_0(t) = 1, T_1(t) = t$. This results in the following iteration for computing $w_k = T_k(B)v$;

$$w_{k+1} = 2Bw_k - w_{k-1}, k = 1, 2, \ldots, m; \quad .$$

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

Remark 1 *If the input matrix X is rectangular or non symmetric, we can consider B to be of the form $B = Y^\top Y$, where Y is a linear transformation of the data matrix X using the mapping (1). This matrix B need not be computed explicitly since only matrix-vector product operations are required.*

4. Threshold selection: An alternate method

In the main paper, we presented a threshold selection method based on the spectral density plots and used the Lanczos spectroscopic method (LSM) to estimate these spectral densities. Here we present an alternate method to estimate the threshold ε using multiple filters.

Let us consider the Chebyshev polynomial rank expression given by

$$r_\varepsilon = \eta_{[\varepsilon,\lambda_1]} \approx \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} \left[\sum_{k=0}^{m} \gamma_k (v_l)^\top T_k(A) v_l \right]. \qquad (2)$$

We observe that the only terms in the above expression that depend on the choice of ε are the γ_k's and the expensive computations $(v_l)^\top T_k(A)v_l$ remain the same for any ε chosen. So, once the scalars $(v_l)^\top T_k(A)v_l$ are computed, we can define any number of filters (defined over different intervals $[a,\ b]$) with no additional cost. Hence, an alternate method for threshold selection/rank estimation is to define several filters and count the eigenvalues by setting $a = \lambda_n$ and incrementing b from λ_n to λ_1 with a small step-size. The eigenvalue counts obtained by these filters will be increasing and in the region of a gap in the spectrum, this count will remain the same. That is, the plot of the eigenvalue-counts obtained by the multiple filters will be an increasing function with plateaus in the region of gaps. We know that the threshold ε must be located at the first gap encountered this way. Thus, we can select the threshold as *the value of* b *for which the first plateau in the eigenvalue count plot occurs*, and the corresponding rank will be the difference between the matrix size and the value of the eigencount at this plateau. From another viewpoint, if we consider the differences between the eigen-counts obtained by these filters, then the threshold ε can be selected as *the value of* b *for which the eigencount difference plot becomes zero for the first time*.

As an illustration, let us consider the numerically rank deficient matrix discussed in the threshold selection section (sec. 4.2) of the main paper. The eigenvalue count plot for this matrix obtained using Chebyshev filters of degree $m = 50$ is given in the left plot of fig. 2. The step size for incrementing b was chosen to be 0.01. We see that the plot increases from 0 to 0.1 indicating the existence of a large number of smaller noise related eigenvalues. Next, there is a plateau between $0.1 - 0.5$. This region corresponds to the gap in the spectrum and we can select the threshold and the corresponding rank in this region. The differences between the eigencounts are plotted in the right plot of fig. 2. We see that the difference plot goes to zero at the gap, and we can choose the corresponding b value as the threshold.

Interestingly, we note that the eigencount difference plot appears similar to the spectral density plot obtained for the matrix (see right plot of fig. 2 in the main paper). Indeed the eigencount difference plot is equivalent to the spectral density plot, since the eigencount over an interval $[a,\ b]$ is just the integration of the spectral density function over the interval. So, in a sense, the two threshold selection methods are equivalent. However, in the multiple filter method, we need to select an incremental step size for b. Experiments show that the spectral density plot method performs better than the multiple filter method for threshold selection.

5. Additional Experiments

In the main paper, we presented several numerical experiments to illustrate the performances of the two rank estimation methods proposed. In this section, we give some additional numerical experimental results. We also give an application from signal processing where our rank estimation methods can be useful.

5.1. 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 matrices namely `deter3` and `dw4096` from UFL database, the matrices consdiered in rows 2 and 3 of table 1 in the main paper. Figure 3 plots their spectra and the corresponding spectral densities obtained by LSM using Chebyshev polynomial of degree $m = 50$. 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 1, main paper) and clearly this value is in the gap.

Similarly, in the second spectrum (third plot of figure 3, 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 1, main paper). These two examples not only show us that the thresholds selected are indeed in the gap of the matrix spectra, but also let us visualize the connection between the actual matrix spectra and the corresponding spectral density plots.

In the next two sections, we shall consider two applications and illustrate how the rank estimation methods based on polynomial filtering perform on these application matrices.

5.2. Matérn covariance matrices

The first application is with the Matérn covariance functions, that are commonly used in statistical analysis applications such as Machine Learning (Rasmussen & Williams, 2006). We demonstrate the performance of the two rank estimator techniques on two such covariance matrices ob-

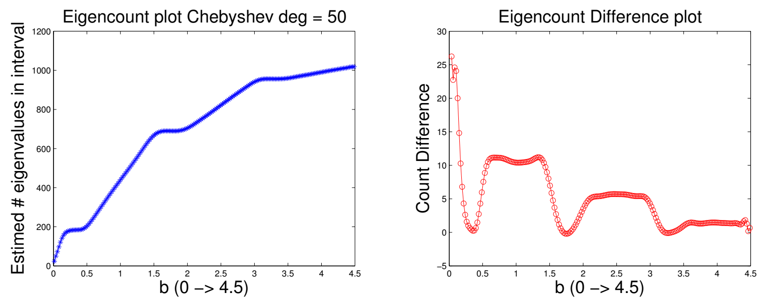

Figure 2. Eigenvalue count plot for the numerically low rank matrix by Chebyshev filters (left) and the eigencount difference plot (right).

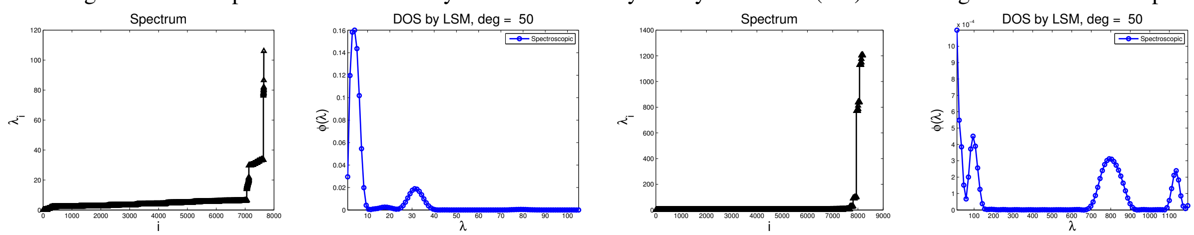

Figure 3. The sprectra and the corresponding spectral densities obtained by LSM.

tained for a 1D and a 2D regular grids[1]. It is found that such covariance matrices are numerically low rank and a low rank approximations of the matrices suffices in many applications.

First, we consider a 1D regular grid of dimension 1024 and define its covariance matrix using a Matern scaling factor $\ell = 7$ (Chen et al., 2013). The covariance matrix will be a 1024×1024 PSD matrix. The approximate rank estimated for this matrix by the extended McWeeny filter method of degree $[2, 42]$ and 30 samples was 16.74. The actual count above the threshold selected is 17. The approximate rank computed by Chebyshev filter method with degree 50 and $\mathrm{n_v} = 30$ was 17.10.

Next, we consider a 2D regular grid with dimension 32×32. The corresponding Matern covariance matrix will be of size 1024×1024. The approximate rank estimated by the extended McWeeny filter method of degree $[2, 48]$ and 30 samples was 64.21. The actual count above the threshold is 65. The approximate rank estimated by Chebyshev filter method with degree 50 and 30 samples was 68.20.

5.3. Estimation of the number of signals

The second application we will consider here comes from Signal Processing. The objective is to detect the number of signals r embedded in the noisy signals received by a collection of n sensors (equivalent to estimating the number of transmitting antennas). This can be achieved by finding the numerical rank of the corresponding sample covariance matrix of the received signals. Here we demonstrate how the rank estimation techniques discussed in the main paper can be employed to estimate the number of signals r, by computing the numerical rank of the sample output covariance matrix.

We consider $n = 1000$ element sensor array receiving $r = 8$ interference signals incident at angles $[-90^0, 90^0, -45^0, 45^0, 60^0, -30^0, 30^0, 0^0]$. The output signal $y(t)$ can be represented as

$$y(t) = \sum_{i=1}^{r} s_i(t) a_i + \eta(t) = As(t) + \eta(t), \qquad (3)$$

where $A = [a_1(\theta_1), a_2(\theta_2), \dots, a_r(\theta_r)]$ is an $n \times r$ mixing matrix, $s(t) = [s_1(t), s_2(t), \dots, s_r(t)]$ an $r \times 1$ signal vector (signals sent from the transmitters) and $\eta(t)$ is a white noise vector, with the noise power set to $-10DB$. The covariance matrix $C = \mathbb{E}[y(t)y(t)^\top]$ is a numerically rank deficient matrix. That is, the matrix is a noisy version of a low rank r matrix. Hence, we can employ the rank estimation methods to estimate the numerical rank of this matrix, in turn estimating the number of signals r in the received signals.

Figure 4 (left) shows the spectral density obtained using Chebyshev polynomial of degree $m = 50$ and number of samples $\mathrm{n_v} = 30$. The threshold ε was estimated by the method described in the main paper using this spectral density plot. Figure 4 (middle) shows the estimated numerical rank by the extended Mc-Weeny filter method with 30 samples. The degree of Hermite polynomial estimated was $[m_0, m_1] = [2, 44]$. The average of approximate ranks estimated over 30 sample vectors was equal to 8.07. The actual count in the interval is 8 (we know there are 8 signals).

[1] These matrices were generated using the codes available at http://press3.mcs.anl.gov/scala-gauss/.

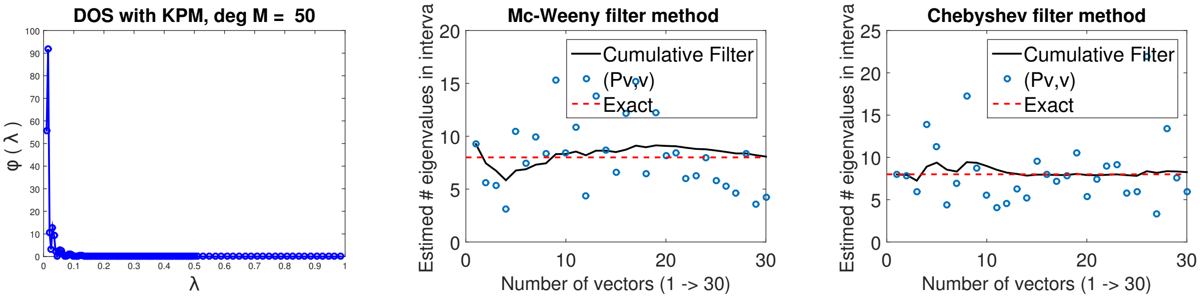

Figure 4. (Left): The spectral density found by KPM. (Middle) Approximate rank estimation by Mc-Weeny filter method for the adaptive beamforming example. (Right) Approximate rank estimation by Chebyshev filtering.

Similarly, figure 4 (right) shows the estimated numerical rank by the Chebyshev filtering method using degree $m = 50$ and $\mathrm{n_v} = 30$. The average of approximate ranks estimated over 30 sample vectors was equal to 8.25. Clearly, both the methods have accurately estimated the number of interference signals embedded in the received signals.

It is observed that the accuracy of these rank estimation techniques in the estimation of the number of signals depends on the interference signal strength and noise power used. There exists a gap between signal related eigenvalues and eigenvalues due to noise in the covariance matrix only when the signal strength is high and noise power is low. In practice the rank estimation will be affected by factors such as the angle of incidence, the number of arrays, the surrounding noise and others.

References

Chen, J, Wang, L, and Anitescu, MIHAI. 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.

Di Napoli, Edoardo, Polizzi, Eric, and Saad, Yousef. Efficient estimation of eigenvalue counts in an interval. *ArXiv preprint ArXiv:1308.4275*, 2013.

Lanczos, C. *Applied analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1956.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning series, MIT Press, 2006.

Saad, Yousef. *Numerical Methods for Large Eigenvalue Problems- classics edition*. SIAM, Philadelpha, PA, 2011.