OPTIMAL BOUNDS FOR EIGENVALUE ESTIMATION USING THE LANCZOS AND POWER METHODS

JARED CHOY

Abstract. The paper published by J Kuczynski and Henry Wozniakowski titled *Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms With a Random Start* [JH92], discusses two algorithms, Power and Lanczos. The algorithms are utilized in order to solve for the largest eigenvalue of a positive semi-definite (PSD) $n \times n$ symmetric matrix. Utilizing a random β vector drawn uniformly, we can initialize the algorithms to see that solving for the largest eigenvalue is possible. In doing so, we measure the Average Relative Error (ARE) and Probalistic Relative Failure (PRF) of both methods and compare between both algorithms. In both algorithms and both theorems, we see that the bounds provided by Lanczos are preferred compared to the Power algorithm. Additionally, we consider a computational simulation of both algorithms, to determine the superior method when computing the largest eigenvalue. The result of our simulation seems to be in line with the claims in the paper.

1. Introduction

A potential problem with power and Lanczos algorithms is that if the initial vector b is orthogonal to the eigenvector η_1 associated with λ_1, then the algorithm will fail to converge to λ_1. To prevent this problem, the initial vector b is chosen randomly on the n-dimensional unit sphere. In doing so, we minimize the probability of choosing an initial vector b orthogonal to the eigenvector η_1. Here, we define ARE as the expected relative error of our estimate integrated over the unit sphere. We also define probabilistic relative failure as the probability an estimate fail to estimate λ_1 with relative error at most ϵ. Regardless of the distribution of the eigenvalues of A, the ARE of power algorithm is bounded above by $0.564\ln(n)/(k-1)$. The bound is sharp in the sense that for every k, there exists a PSD matrix A such that the relative error is bounded below by $0.5\ln(n)/(k-1)$. The ARE for the Lanczos algorithm is bounded above by $2.575(\ln(n)/(k-1))^2$. Detailed simulated tests were performed to confirm this bound [MCJ+19]. The rate of convergence for power algorithms depends on p, the multiplicity of λ_1, and θ, the ratio between the largest and second largest eigenvalues. When $p \geq 3$, the rate of convergence for the power algorithm is θ^{2k-1}. When $p = 1$, the rate of convergence is θ^k. If b is chosen deterministically and not orthogonal to η_1, then the rate of convergence for the power algorithm is θ^{2k-2}. When eigenvalues are known, let λ_{p-1} denote the second largest eigenvalue and λ_n denote the smallest eigenvalue. Then, $\sigma = \sqrt{(\lambda_1 - \lambda_{p+1})(\lambda_1 - \lambda_n)}$. The relative error of the Lanczos algorithm is bounded above by $2.589\sqrt{n}((1-\sigma)/(1+\sigma))^{k-1}$. The paper also shows that for a randomly selected initial vector b, the average probability that the power algorithm fails to achieve a relative error of ϵ is bounded by $\sqrt{n}(1-\epsilon)^k$, while the probability for failure of Lanczos algorithms is $1.648\sqrt{n}e^{\sqrt{\epsilon}(2k-1)}$.

2. Algorithm Description

The two algorithms the paper focuses on is the power algorithm and the Lanczos algorithm. The idea behind the power method depends on the fact that if A is an $n \times n$ positive semi-definite matrix, its eigenvectors $v_1, v_2 ... v_n$ form an orthonormal basis for R^n. Therefore, any vector c_0 in R^n can be expressed in the following way.

$$c_0 = r_1 v_1 + r_2 v_2 + ... + r_n v_n = \sum_{i=1}^{n} r_i v_i$$

$$c_1 = Ac_0 = \sum_{i=1}^{n} r_i \lambda_i v_i, \;\; c_2 = Ac_1 = \sum_{i=1}^{n} r_i \lambda_i^2 v_i, \;\; \dots \;\;, c_k = A^k c_0 = \sum_{i=1}^{n} r_i \lambda_i^k v_i$$

We can observe that as n increases, the difference between λ_1^n and other λ_i^n increases. We also observe that $r_1 \lambda_1^n v_1$ becomes the dominant component of c_n. Now we normalize c_i with respect to λ_1 to get the following equation.

$$c_{i+1} = r_1 v_1 + r_2 \frac{\lambda_2}{\lambda_1}^{i+1} v_2 + ... + r_n \frac{\lambda_n}{\lambda_1}^{i+1} v_n$$

With each iteration c_i converges to $r_1 v_1$, and as such we compute $\lambda_{hat} = c_i^\top c_{i+1}$.

The fundamental idea behind the Lanczos algorithm depends on the fact:

$$\lambda_1 = \max \frac{v^\top A v}{v^\top v} \text{ for } v \in R^n.$$

Instead of simply applying matrix multiplication to a vector and normalizing the result at each iteration, we seek a solution to $\lambda_1^\top = \max v^\top v^\top / v^\top v$ for $v \in V$. where V is some subspace. We define the Krylov information $K(A, q, k)$ as $[q, Aq, A^2 q, ... A^k q]$. At each iteration of the Lanczos algorithm, we wish to find the following.

$$\lambda^k = \max \frac{v^\top A v}{v^\top v} \text{ for } v \in span[K(A, q, k)]$$

Since $span(K(A, q, k))$ is in $span[K(A, q, k+1)]$, we know that λ^k is increasing with each subsequent k. Let Q_k be a matrix such that its columns form an orthonormal basis for $span(K[A, q, k])$. Thus, $\lambda^k = max(\frac{v^\top Q_k^\top A Q_k v}{v^\top v})$, and finding λ^k is equivalent to finding the largest eigenvector of $Q_k^\top A Q_k$. We can always express $Q_k^\top A Q_k$ as a tri-diagonal matrix T.

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & \cdots & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & \cdots & \cdots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & 0 & \cdots & 0 \\ \vdots & 0 & \beta_3 & \alpha_4 & \beta_4 & \ddots & \vdots \\ \vdots & \vdots & 0 & \beta_4 & \ddots & \ddots & \beta_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & 0 & \cdots & \cdots & \beta_{n-1} & \alpha_n \end{bmatrix}$$

Let $Q = Q_k$, which allows us to we have T in the form of and we wish to derive T through recursion. Since Q is orthonormal, finding T is equivalent to solving $AQ = QT$. At each column k, we have $Aq_k = QT_k = \beta_{k-1} q_{k-1} + \alpha_k q_k + \beta_k q_{k+1}$. We first set $\beta_0 q_0 = 0$, to do so we set $\beta_0 = 1$ and $q_0 = 0$. We set $q_1 = q$, as the initial vector used for Krylov information. Since Q is orthonormal, its columns q_i are also orthogonal and thus we can solve for α_k.

$$q_k^\top A q_k = q_k^\top \beta_{k-1} q_{k-1} + q_k^\top \alpha_k q_k + q_k^\top \beta_k q_{k+1} = q_k^\top \alpha_k q_k = \alpha_k ||q_k||^2 = \alpha_k$$

Now we wish to derive the quantity q_{k+1}. We first define the following.

$$r_k = \beta_k q_{k+1}$$

$$r_k = A q_k - \beta_{k-1} q_{k-1} - \alpha_k q_k = (A - \alpha_k I) q_k - \beta_{k-1} q_{k-1}$$

Since q_i possesses unit length vectors, we define $\beta_k = ||r_k||^2$ and $q_{k+1} = r_k/\beta_k$. Through this procedure we can derive a series of tri-diagonal matrix T_k such that the largest eigenvalue of T_k is the Lanczos estimator for the largest eigenvalue of A. We may compute the eigenvalues of each T_k through more efficient algorithms such as QR decomposition or divide and conquer algorithms. At $k = n$, Q_k becomes an $n \times n$ orthonormal matrix, and $T = Q^\top A Q$, $A = QTQ^\top$. Let v_1,λ_i be the eigen pair associated with eigenvalue of T. Let $y = Qv_i$. $Ay = QTQ^\top Q v_i = QTv_i = \lambda_1 Q v_i = \lambda_1 y$. Thus, A and T possess the same eigenvalues.

3. Main Results

3.1. Average Relative Error: Power.

The average relative error of the power algorithm is given by a function bounded by a constant and fractional term, both dependent on the size of n.

$$\mathbb{E}[\xi^{\text{pow},A,k}] \le \alpha(n) \frac{\ln n}{k-1} \tag{1}$$

In theorem (1), we let $\xi^{\text{pow},A,k}$ be the power algorithm we defined in prior sections, A be any PSD matrix, and $k \ge 2$. Let us also define $\alpha(n)$ as the constant term who's value depends on the size of the matrix, and $(k-1)$ be the iterative count of our algorithm. In this theorem, we have the following:

$$\pi^{-\frac{1}{2}} \le a(n) \le 0.871 \text{ and for large } n,\ a(n) \sim \pi^{-\frac{1}{2}} + 0.564$$

This theorem tells us that for any distribution of eigenvalues that compose the matrix A, the average relative error of the power algorithm will always be bounded by $0.871 \frac{lnn}{k-1}$, and when n is large, we can replace the constant of 0.871 with 0.564.

By the theorem, we see that we have strong dependence on the size of n, and the iterative count of the algorithm. It's also important to note the logarithmic nature of the function, and how it is inversely proportional to the size of n and the iterative count. Due to this, it suggests that for large matrices, if we want to achieve a low average relative error, we will need to observe over a high number of iterations.

3.2. Average Relative Error: Lanczos.

Moving into the Lanczos algorithm, we consider the case when the matrix A is PSD, and the number of A's distinct eigenvalues, m. We assume the case when $k \ge 4$, and $n \ge 8$ as in our prior case. If $m \ge k$, then

$$\mathbb{E}[\xi^{\text{Lan},A,k})] = 0;$$

$$\mathbb{E}[\xi^{\text{Lan},A,k})] \le 0.103 \left(\frac{\ln\ (n(k-1)^4)}{k-1} \right)^2 \le 2.575 \left(\frac{\ln n}{k-1} \right)^2,\ \text{for } k \in [4, m-1] \tag{2}$$

By this theorem, we know that the upper bound of the Lanczos algorithm is less than that of the Power algorithm, and thus should give us a better average relative error. We see that the ARE of the Lanczos algorithm decreases with the total number of iterations, and does so faster than that of the Power method. One of the more pleasings parts of the Lanczos algorithm compared to the Power method, is that we see convergence within m steps for $m < n$. The aforementioned property implies that only a set of vectors b that have measure zero, the algorithm will fail to converge. This is extremely noteworthy, as it means the algorithm will successfully approximate the largest eigenvalue given any other starting vector of b.

Furthermore, we should note that the texture of the bound described by Lanczos, may not be sharp, meaning that there could be a better approximate bound or a slightly worse one. Hence, we might be overestimating the actual error being observed. The theoretical bound should be safe, and practical results support the idea that the Lanczos performs better. With error decreasing steadily with $O(k^{-2})$, this suggests that the term is being overestimated by a factor of $\ln n$.

3.3. **Probabilistic Relative Failure: Power.**

$$f^{\text{prob}}(\xi^{pow}, A, k, \epsilon) \leq \min\left\{0.824, \frac{0.354}{\sqrt{\epsilon(k-1}}\right\}\sqrt{n}(1-\epsilon)^{k-\frac{1}{2}} \tag{3}$$

The bounds presented in the PRF of the Power method are sharp in nature; however, they tend to converge to zero at a rate of $(1-\epsilon)^{k-\frac{1}{2}}$. Additionally, the integral depends on the function $g(\epsilon, k) = \frac{1}{\sqrt{2e}} \frac{(1-\epsilon)^{k-1/2}}{\sqrt{\epsilon(k-1)}}$. When ϵ is small, this is very unsatisfactory. With ϵ of good size (say $\epsilon = 0.5$) this rate of convergence is satisfying. It's important to note that the dependence of n in this equation, and how it influences the PRF more substantially than that of the ARE. We see though, that the total steps required for satisfactory accuracy depends depends primarily on the error tolerance, which affects the function linearly, while n and δ have logarithmic impact.

3.4. **Probabilistic Relative Failure: Lanczos.**

$$f^{\text{prob}}(\xi^{Lan}, A, k, \epsilon) = 0;$$

$$f^{\text{prob}}(\xi^{Lan}, A, k, \epsilon) \leq 1.648\sqrt{n}e^{-\sqrt{\epsilon}(2k-1)} \tag{4}$$

Using the same constraints as before, we see that the probabilistic case of Lanczos converges in m steps. The error is roughly bounded by $\sqrt{n}\exp(-\sqrt{\epsilon}(2k-1))$. We compare this with the sharp bound of the Power algorithm, and we see that we have a better bound in Lanczos. Additionally, Lanczos needs approximately $k^{\text{Lan}} \leq \ln(n/\delta^2)/(4\sqrt{\epsilon}$ steps to guarantee δ-failure. Comparing with Power's step count, $k^{\text{Pow}} = \ln(n/\delta^2)/(2\epsilon)$, we see that Lanczos is more appealing. We also observe weak dependence on δ which affects the number of steps in a logarithmic pattern. However, we also see that the Lanczos PRF depends more heavily on ϵ compared to Power.

4. Key Proof Ideas

4.1. **Constraints.** In the proofs of the average relative error, and the probabilistic relative failure, we first offer a few constraints. First, we require a symmetric PSD matrix A for all cases [PSS82]. Next, we assume that $n \geq 8$ in both the power algorithm and the Lanczos algorithm. This is done for simplicity in calculation of the average relative error. In the Lanczos ARE calculation, we require $k \geq m$, where k is the number of iterations and m the number of distinct eigenvalues. With the average relative error of the Lanczos algorithm, we also state that $k \geq 4$, and this is done to simplify some formulas in the proof. Moving to the probabilistic case, we assume that $\epsilon > 0$. For the probabilistic relative failure of the Power method, we require $k \geq 2$. This is due to convergence needs, as we can't expect convergence off of one iteration [GL96]. For the Lanczos algorithm, we similarly require $k \geq m$. Another key component of the proofs, is the usage of the unit ball. The proofs integrate using the space of the unit ball, in order to normalize and simplify the Power method's performance. The ball allows us to integrate over a well defined space, with orthogonal invariance, where transformations will not affect the ball's space.

4.2. **Average Relative Error of Power Proof.** The proof of the average relative error for the Power method involves some key concepts and properties that will be used for other proofs as well. We first decompose the matrix A to be a makeup of the eigenpairs, where the associated eigenvectors form an orthonormal basis. After, we rewrite the relative error of the power function in matrix notation to easily express the Kyrlov subspace. We then define the average relative error as an integral that resides over the unit ball B. The unit ball accounts for all possible vectors b, with respect to that of the uniform distribution. Once we have this integral we apply Schwarz's inequality: $\sum_{i=1}^{n} y_i z_i \leq (\sum_{i=2}^{n} y_i^2)^{1/2} (\sum_{i=2}^{n} z_i^2)^{1/2}$, in order to receive an upper bound for the ARE. We then introduce a function $H(t)$ that helps us to compute the integrals. Combining the bounds formed in the previous steps, we can yield an overall average bound.

4.3. **Average Relative Error of Lanczos Proof.** In the Lanczos proof of the Average Relative Error, we utilize the Rayleigh quotient that exists within the Krylov subspace. In doing so, we take the maximum of $(Ax, x)/(x, x)$ for $x \in \operatorname{span}(b, Ab, Ab^2, \ldots, Ab^{k-1})$. We can rewrite the resulting term and relate it to taking the relative error of the polynomial that is fitted by the Lanczos algorithm. Additionally, we restrict the polynomials P such that $P(\lambda_1) \neq 0$, bringing us to $Q(t) = P(\lambda_1 t)/P(\lambda_1)$ where $Q \in \mathcal{P}_{\|}$ and $Q(1) = 1$. We let $\mathcal{P}_{\|}(\infty)$ represent these types of polynomials, which allows us to rewrite our equation in terms Q. In calculating the bound, we first posit the assumption where $k \geq m$, which implies that we have m distinct elements in the set $x_1, x_2, \ldots, x_n$ along with $t_1, t_2, \ldots, t_m, t_1 = 1$. Taking $Q(x) = \prod_{i=1}^{m} (x - t_i)/(1 - t_i)$, we see that for $b_1 = 0$, the integrand disappears, and thus $e_k^{Lan} = 0$. Positing the case when $k \in [4, m-1]$ we can find an upper bound by rewriting the order of integration and taking the infimum of the integral. We bound the error using the Chebyshev polynomial of degree $k-1$, and yield the upper bound of the ARE for the Lanczos algorithm.

4.4. **Probabilistic Relative Failure of Power Proof.** The proof begins by defining the set Z, with which constitutes all the vectors b for which the algorithm fails. Like the proofs prior, we consider a function $H(x)$, who's maximum value lies at $(1-\epsilon)(1-1/(2k-1))$. The primary usage of this function is to provide a bound on the sum of the vectors b. In doing so, we receive an upper bound for the error term. Akin to the previous proofs, we take the integral over the space provided by the function $g(\epsilon, k)$. Computing the integral, we arrive at a final bound which gives us the probabilistic relative failure of this method.

4.5. **Probabilistic Relative Failure of Lanczos Proof.** As before in the Power case, we begin by defining the set Z, but this time we use polynomials Q of degree less than $k-1$. We then need to find the measure of the set, and offer two cases: $k \geq m$ and $k < m$. For the first case, we get the empty set of Z, thus the probability is zero. For the second case, we derive an upper bound on this probability. Utilizing the Chebyshev polynomial of the second degree $2(k-1)$, the equation is updated to accordingly. We define a function H(x), and look for it's maximum. Bounding on W, we compute the set equations and receive an upper bound.

5. Full Proof

5.1. **Average Relative Error of Power.** To compute error, we consider the equation $\xi^{\text{Pow}}(A,b,k) = (Ax,x)/(x,x)$. In the context of our problem, we can let $b = \sum_{i=1}^{n} b_i\eta_i$ where η represents the eigenvectors of A, with orthonormal basis. For bounding the error, we use the aforementioned equation to calculate $\xi^{\text{Pow}}(A,b,k)$.

$$\frac{\lambda_1 - \xi^{\text{Pow}}}{\lambda_1} = \xi^{\text{pow}}(A,b,k) = \frac{\sum_{i=1}^{n} b_i^2 \lambda_i^{2k-1}}{\sum_{i=1}^{n} b_i^2 \lambda_i^{2k-2}} \tag{5}$$

From the prior section, we know that the unit ball's bounds can be used to calculate the ARE, as . Thus, let us rewrite (5) as an integral.

$$e_k^{\text{pow}} = \frac{1}{c_n}\int_{B'} \sum_{i=2}^{n} b_i^2 x_i^{2(k-1)}(1-x_i)\left(\int_{b_1^2 \le 1-||b||_{n-1}^2} \frac{db_1}{b_i^2 + \sum_{i=2}^{n} b_i^2 x_i^{2(k-1)}}\right) d\overset{\rightharpoonup}{b}$$

In this integral, $c_n = \frac{\pi^{\frac{n}{2}}}{\gamma(1+\frac{n}{2})}$, and represents the Lebesgue measure of the unit ball. This is the standard way to assign a measure to higher dimensional subsets, and because of its orthogonally invariant property, it allows us to integrate with respect to the vector space b.

$$= \frac{2}{c_n}\int_{B'} \sum_{i=2}^{n} b_i^2 x_i^{2(k-1)}(1-x_i)\left(\sum_{i=2}^{n} b_i^2 x_i^{2(k-1)}\right)^{-1/2} \arctan(h(b))\, d\overset{\rightharpoonup}{b}$$

We use the property of the inverse trig functions to get $\arctan(h(b))$ where $h(b) = \sqrt{(1-|b||_{n-1}^2)/(\sum_{i=2}^{n} b_i^2 x_i^{2(k-1)})}$. This is important to us, as we are able to now yield a bounded equation using Schwarz's inequality: $\sum_{i=1} y_i z_i = (\sum_{i=1} y_i^2)^{1/2} (\sum_{i=1} z_i^2)^{1/2}$. We can use Schwartz's inequality for integrals to yield the following.

$$e_k^{\text{pow}} \le \frac{2}{c_n}\int_{B'} \left(\sum_{i=2}^{n} b_i^2 x_i^{2(k-1)}(1-x_i)^2\right)^{\frac{1}{2}} \arctan(h(b))\, db$$

$$e_k^{\text{pow}} = \frac{2c_{n-1}}{c_n}\left(\frac{1}{c_{n-1}}\int_{B'} \arctan^2 h(b)\left(\sum_{i:x_i<\beta} b_i^2 x_i^{2(k-1)}(1-x_i)^2 + \sum_{i:x_i>\beta} b_i^2 x_i^{2(k-1)}(1-x_i)^2\right) db\right)^{\frac{1}{2}}$$

We consider the function $H(t) = (1-t)^2 t^{2(k-1)}$. $H(t)$ is maximized at $t_0 = 1 - 1/k$, where it is also increasing from $[0, t_0]$. Let $\beta \le t_0$, since $\arctan(z) \le z$ and $\arctan(z) \le \frac{\pi}{2}$, we yield the following.

$$\arctan^2 h(b) \sum_{i:x_i<\beta} b_i^2 x_i^{2(k-1)}(1-x_i)^2 \le (\frac{\pi}{2})^2 \sum_{i=2}^{n} b_i^2 \beta^{2(k-1)}(1-\beta)^2$$

$$\arctan^2 h(b) \sum_{i:x_i>\beta} b_i^2 x_i^{2(k-1)}(1-x_i)^2 \leq (h(b))^2 \sum_{i=2}^{n} b_i^2 \beta^{2(k-1)}(1-\beta)^2 = (1-|b||_{n-1}^2)(1-\beta)^2$$

.

We combine these bounds and evaluate the integral in order to receive

$$e_k^{\text{pow}} \leq \frac{2c_{n-1}}{c_n}(1-\beta)\left(\frac{1}{c_{n-1}}\left(\frac{\pi^2}{4}\beta^{2(k-1)}-1\right)\int_{B'}||b||_{n-1}^2 db - 1\right)^{\frac{1}{2}}$$

We evaluate the integral $\int_{B'}||b||_{n-1}^2 db = \frac{n-1}{n+1}c_{n-1}$, and consequently rewrite the bounded equation as $e_k^{\text{pow}} \leq \frac{2c_{n-1}}{c_n}(1-\beta)\sqrt{(\frac{\pi^2}{4}\beta^{2(k-1)}+\frac{2}{n})}$. Now we wish to derive a bound for $\frac{c_{n-1}}{c_n}$. To do so, we must observe another inequality with that is relative to our gamma functions. We evaluate this as:

$$\sqrt{\frac{n}{2\pi}} \leq \frac{c_{n-1}}{c_n} = \sqrt{\frac{n}{2\pi}}\frac{\sqrt{n/2}\gamma(n/2)}{\gamma(n/2+\frac{1}{2})} \leq \sigma\sqrt{\frac{n}{2\pi}}\ ,\ \sigma = \frac{192}{105\sqrt{\pi}} \leq 1.032 \tag{6}$$

For $k-1 \geq \pi^{-\frac{1}{2}}ln(n)$, we take $\beta = 1-ln(n/(2(k-1)))$, then β in $(0,t_0]$ and evaluating the whole inequality, we arrive at our final bound, concluding our proof of: $e_k^{\text{pow}} \leq 2\sigma\sqrt{\frac{n}{2\pi}}\frac{ln(n)}{2(k-1)}\sqrt{\frac{\pi^2}{4n}+\frac{2}{n}} \leq 0.871\frac{ln(n)}{k-1}$.

5.2. **Average Relative Error of Lanczos.** The proof of the ARE of Lanczos is similar to the proof of Power in many ways. We look at the ratio between $(Ax,x)/(x,x)$ again but now we wish to take the maximum. We define $\mathcal{P}_k$ as the set of polynomials less than or equal to the degrees $k-1$. The average relative error of Lanczos can be defined by 7.

$$\frac{\lambda_1 - \xi^{\text{Lan}}}{\lambda_1} = \min_{P\in\mathcal{P}_k}\frac{\sum_{i=1}^n b_i^2 P^2(\lambda_i)(1-\lambda_i/\lambda_1)}{\sum_{i=1}^n b_i^2 P^2(\lambda_i)} \tag{7}$$

Due to continuity, we only use P polynomials that satisfy $P(\lambda_i) = 0$. Allow $Q(t) = P(\lambda_1 t)/P(\lambda_1)$, where $Q \in \mathcal{P}_k$ and $Q(1) = 1$. Similar to the power algorithm, we again take the integral with respect to the unit ball. We utilize the infimum in our equation, as we must do so if we want to find the polynomial function that minimizes the error. As in our constraints, we assume $k \geq m$ and thus the set of x_i should contain m distinct elements $\{t_1, t_2, \ldots, t_m\}$ and $t_1 = 1$. We now define $Q(x) = \prod_{i=2}^m (x-t_i)/(1-t_i)$. By assuming $k \in [4, m-1]$ we derive the upper bound.

$$e_k^{\text{Lan}} \leq \frac{1}{c_n}\inf_{Q\in\mathcal{P}_k}\int_{B'}\frac{\sum_{i=2}^n b_i^2 Q^2(x_i)(1-x_i/x_1)}{b_1^2\sum_{i=2}^n b_i^2 Q^2(x_i)}db$$

It should be noted that this integral is very similar to that of the power method, and if we evaluate the integral using the polynomial Q instead of x^{k-1} we can avoid any trig function and arrive at a very similar equation. With any uniform β and letting $w(\beta) = \inf_{Q\in\mathcal{P}_k}\max_{0\leq x\leq\beta} Q^2(x)(1-x)^2$ we rewrite the bound to the following using 6.

$$e_k^{\text{Lan}} \leq \frac{2c_{n-1}}{c_n}\left(\frac{\pi^2}{4}w(\beta)+\frac{2(1-\beta)^2}{n}\right)^{1/2} \to 0.412\sqrt{\pi^2 n w(\beta)+8(1-\beta)}$$

Now in order to get a final bound, we have to see what the upper bound for $w(\beta)$ is equal to. Using the Chebyshev polynomial T_{k-1} and the associated equation $Q(x) = T_{k-1}((2/\beta)x - 1)/T_{k-1}((2/\beta) - 1$, we can see that it approximates to $4e^{-4(k-1)\sqrt{1-\beta}}$. If we let $\gamma = \sqrt{1-\beta}$ we can handle complexity, and ultimately solve the equation.

5.3. **Probabilistic Relative Rate of Failure for the Power Algorithm.** We first define the set Z which contains all of the vectors b for which the algorithm goes to failure. We use the expression from given in the Power proof of ARE, and once more let $x_i = \lambda_i/\lambda_1$.

$$Z = \{b \in R^n : ||b|| = 1, \sum_{i=2}^{n} b_i^2(1-\epsilon-x_i)x_i^{2(k-1)} > \epsilon b_1^2\}$$

.

Now, let us substitute part of the summation using $H(x) = (1-\epsilon-x)x^{2(k-1)}$ for $x \in [0,1]$. From this equation, it is clear that H attains its maximum at $x^* = (1-\epsilon)(1-\frac{1}{2k-1})$. This allows us to rewrite the summation as $\sum_{i=2}^{n} b_i^2(1-\epsilon-x_i)x_i^{2(k-1)} \leq H(x^*)\sum_{i=2}^{n} b_i^2$. With this, we can redefine the set of Z.

$$Z^* = \{b \in R^n : ||b|| = 1, \sum_{i=2}^{n} b_i^2 > \alpha b_1^2\}, \text{ with } \alpha = \frac{\epsilon}{H(x^*)}$$

The probability of the function going to failure can be given by $f^{\text{prob}}(\xi^{\text{pow}}(A,b,k)) = \mu(Z) \leq \mu(Z^*)$. We can solve this inequality by $1-\mu(Z^*) = \frac{2c_{n-1}}{c_n}\int_0^1 \min(1-t^2, \alpha t^2)^{\frac{n-1}{2}}dt$. $min(1-t^2, \alpha t^2) = \alpha t^2$ when $t \leq \frac{1}{\sqrt{1+\alpha}} = g(k,\epsilon)$ and $\min(1-t^2,\alpha t^2) = 1-t^2$ for $t \geq g(k,\epsilon)$. The function $g(k,\epsilon)$ is used in computation of the bound, as it acts as the volume of the set. Intuitively, the function is used to measure the probability that a given algorithm fails to approximate the largest eigenvalue within an error ϵ.This function is defined by the following.

$$g(k,\epsilon) = \frac{(1-\epsilon)^{k-1/2}(1-\frac{1}{2k-1})^{k-1}}{\sqrt{(1-\epsilon)^{2k-1}(1-\frac{1}{2k-1})^{k-1})^{2(k-1)} + 2(k-1)\epsilon}}$$

Using the newly defined equation, and the set of Z^*, we now rewrite the bound for $1-\mu(Z^*)$.

$$1-\mu(Z^*) = \gamma\Big(\int_0^{g(k,\epsilon)} (\alpha t^2)^j dt + \int_{g(k,\epsilon)}^1 (1-t^2)^j dt\Big)$$
$$= \gamma\Big(\frac{1}{n\sqrt{1+\alpha}}\Big(\frac{\alpha}{1+\alpha}\Big)^j + \int_0^1 (1-t^2)^j dt + \int_0^{g(k,\epsilon)} (1-t^2)^j dt\Big)$$

In this bound, we have $j = (n-1)/2$ and $\gamma = 2c_{n-1}/c_n$. From our prior definition of $1-\mu(Z^*)$, we know that $c_n == 2c_{n-1}\int_0^1 (1-t^2)^{\frac{n-1}{2}}dt$. Coupling this with (6) we know that $\gamma \leq 2.064\sqrt{n/(2\pi)}$. Applying these to the bound, we arrive at the following theorem, completing our proof.

$$\mu(Z^*) \leq 0.824\sqrt{n}\int_0^{g(k,\epsilon)} (1-t^2)^(n-1)/2dt \leq 0.824\sqrt{n}g(k,\epsilon) \tag{8}$$

5.4. **Probabilistic Relative Rate of Failure for the Lanczos Algorithm.** Proof of PRF for Lanczos algorithm included in the appendix

6. Simulations

We performed our own Lanczos algorithm with a 250x250 matrix A with eigenvalues selected from the distribution $\lambda_i = 1 + cos\frac{(2i-1)\pi}{2n}$. With i = 1,2,...250. We perform Lanczos algorithm on the matrix with 30 random vectors selected from the unit ball. We define r1 as the theoretical upper bound of the average relative error over the observed relative error. We define $r2 = \frac{0.103r1}{ln^2(n(k-1)^4)}$. We iterate over k-1 = [10,20,30,40,50,60,70,80,90,1000]. The results are shown on the table below: From this table we can observe that the r1 values are

Average Error	Estimated Upper Bound	R1	R2	Min Error	Max Error
0.006087	0.223537	36.721486	0.017428	0.004084	0.008035
0.001987	0.078899	39.710424	0.013349	0.001281	0.002711
0.001021	0.041865	41.020095	0.011550	0.000556	0.001709
0.000661	0.026468	40.030922	0.010028	0.000316	0.001371
0.000339	0.018464	54.533428	0.012534	0.000107	0.000724
0.000203	0.013721	67.491154	0.014496	0.000039	0.000504
0.000136	0.010656	78.344620	0.015918	0.000020	0.000409
0.000101	0.008550	84.538241	0.016389	0.000011	0.000306
0.000084	0.007035	83.443889	0.015536	0.000007	0.000243
0.000066	0.005904	89.381655	0.016061	0.000005	0.000222

Table 1. Estimation Errors and Bounds

increasing as the number of iterations increased, the value of r1 increased. This observation does confirm the paper's claim that the theoretical relative error presented in this paper is likely an overestimation. We perform the same procedure for the power algorithm and present the result below.

Average Error	Estimated Upper Bound	R1	R2	Min Error	Max Error
0.020321	0.480919	23.665629	42.861172	0.016201	0.023555
0.011885	0.240460	20.232203	73.285686	0.009562	0.014085
0.008838	0.160306	18.138326	98.551778	0.007352	0.010747
0.007147	0.120230	16.822100	121.867022	0.005863	0.009404
0.006024	0.096184	15.966686	144.587513	0.004702	0.008466
0.005215	0.080153	15.369102	167.011252	0.003921	0.007727
0.004604	0.068703	14.922923	189.189891	0.003377	0.007106
0.004125	0.060115	14.574999	211.175979	0.002982	0.006563
0.003737	0.053435	14.297397	233.048049	0.002683	0.006077
0.003417	0.048092	14.073397	254.885385	0.002448	0.005635

Table 2. Estimation Errors and Bounds

From the much smaller values of r1 observed in this simulation it would appear that the theoretical upper bound for expected relative error of power algorithm is less likely to be an overestimation.

The total number of pages excluding contributions, references and the appendix should be at most 8 pages.

this will be last page of the main section of the project report. Top of page 9 will be contributions and then you can start with references.

7. Contributions

The following are the contributions of the individuals:

- Arthur Wu: Introduction, Algorithm Description, contributed to full proof of bounds for relative error of power algorithm and average rate of failure of power algorithm, Simulations.
- Jared Choy: Abstract, Main Results, Key Proof Ideas, Full Proof

References

[GL96] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, 3rd ed., The Johns Hopkins University Press, 1996.

[JH92] Kuczynski J and Wozniakowski H, *Estimating the largest eigenvalue by the power and lanczos algorithms with a random start**, Society for Industrial and Applied Mathematics **13** (1992), no. 4, 1094–1122.

[MCJ+19] Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I Jordan, *Sampling can be faster than optimization*, Proceedings of the National Academy of Sciences **116** (2019), no. 42, 20881–20885.

[PSS82] B. N. Parlett, H. Simon, and L. M. Stringer, *On estimating the largest eigenvalue with the lanczos algorithm*, Mathematics of Computation **38** (1982), no. 157, 153–165.

Appendix A. Additional Details of the Proofs

Like the Power algorithm, we define the set Z which contains the vectors that will fail to obtain error rate within ϵ.

$$Z = \{b : \|b\|_1 = 1, \lambda_1 - \xi^{\text{Lan}}(A, b, k) > \epsilon\lambda_1\}$$

As seen before in (7), we can rewrite the bound in terms of the infimum, and the polynomial equation of Q.

$$Z = \left\{ b : \|b\|_1 = 1, \inf_{Q \in \mathcal{P}_k} \sum_{i=2}^{n} b_i^2 Q^2(x_i)(1 - \epsilon - x_i) > b_1^2 \right\}$$

Looking back to the proof of the Lanczos ARE, we know that if we define $Q(x) = \prod_{i=2}^{m}(x - t_i)/(1 - t_i)$, then we will end up with Z being the empty set. Thus, the probability function will go to zero. For $\epsilon = 0$, the bound holds trivially because it implies there's no error tolerance, and therefore the probability of failing to approximate within zero error is one. Thus, we focus on the case when $\epsilon > 0$. In doing so, we must define the following function w_k as the set of polynomials with degree at most $k - 1$.

$$w_k = \inf_{Q \in \mathcal{P}_k(1)} \max_{0 \le x \le 1} Q^2(x)(1 - \epsilon - x)$$

With this new equation, we can then update the measure of Z in terms of b, and w. Using Z^* we are able to derive the upper bound of the probabilistic relative failure, $f_k^{\text{Lan}} \le \mu(Z^*)$

$$Z^* = \left\{ b : \|b\|_1 = 1, \sum_{i=2}^{n} b_i^2 > b_i^2 \frac{\epsilon}{w_k} \right\}$$

Recall from (8), we know $\mu(Z^*) = 0.824\sqrt{n}g(k,\epsilon)$, thus we can rewrite the probability equation to use our value of $g(k,\epsilon)$. Substituting $g(k,\epsilon) = 1/\sqrt{1+\alpha}$, where $\alpha = \epsilon/w_k$

$$f_k^{\text{Lan}} \leq 0.824\sqrt{n}g(k,\epsilon) = 0.824\sqrt{\frac{n}{1+\epsilon/w_k}}$$

Following the original formula of w_k, we update the equation as the following.

$$w_k = 4\epsilon\left(\frac{1-\sqrt{\epsilon}}{1+\sqrt{\epsilon}}\right)^{2k-1}\left(1-\left(\frac{1-\sqrt{\epsilon}}{1+\sqrt{\epsilon}}\right)^{2k-1}\right)^{-2}$$

Define $Q(x)$ as a polynomial constructed to bound the function of the Chebyshev polynomials, $U_{2(k-1)}$ of which have degree $2(k-1)$. We can represent this function as the ratio between these polynomials.

$$Q(x) = U_{2(k-1)}(\sqrt{x/(1-\epsilon)})/U_{2(k-1)}(\sqrt{1/(1-\epsilon)})$$

As U is always an even function, we know with perfect certainty that the degree of $Q(x)$ is $k-1$. Thus, $Q(1) = 1$ and as such $Q \in \mathcal{P}_k$. We now define the function of $H(x)$ to weight the polynomial in order to approximate tighter bounds.

$$H(x) = \sqrt{1-\epsilon-x}Q(x), \text{ where } x \in [0, 1-\epsilon]$$

Next, let us define t_i as a set of points specifically chosen to stem from Chebyshev nodes. The squared cosine function will allow the points to exist on the bound from $[0,1]$, which are then scaled by a factor of $(1-\epsilon)$ in order to change the interval. We use these points as they help us minimize the max error, which we want to do for the sake of approximating tight bounds.

$$t_i = (1-\epsilon)\cos^2\frac{(2i-1)\pi}{2(2k-1)}, \quad i = 1, 2, \ldots, k$$

With these points, let us rewrite the function $H(x)$ in terms of t_i. Reintroducing w_k, we can set this value to be lesser than the maximum of $H^2(x)$ within the bounds of zero and $(1-\epsilon)$. We then let $c = ((1-\sqrt{\epsilon})/(1+\sqrt{\epsilon}))^{2k-1}$, and compute a defined value for the bound.

$$H(t_i) = \frac{\sqrt{1-\epsilon}}{U_{2(k-1)}(1/\sqrt{1-\epsilon}}(-1)^{i-1}$$

$$w_k \leq a \to \max_{0\leq x\leq 1-\epsilon} H^2(x) = \frac{1-\epsilon}{U^2_{2(k-1}(1/\sqrt{1-\epsilon})} = 4\epsilon c(1-c)^{-2}$$

To compute the bound, let us consider the case when $w_k < a$. In this case, there is a polynomial P that satisfies both $P \in \mathcal{P}_k(1)$ and $\max_{x\in[0,1]} P^2(x)(1-\epsilon-x) < w_k$. Thus, the function $H(x)$ can again be rewritten as the following.

$$H(x) = \sqrt{1-\epsilon-x}(Q(x)-P(x)), \text{ where } x \in [0, 2-\epsilon]$$

This function is one which alternates for each t_i which means that the difference $Q(x) - P(x)$ satisfies having at least $k-1$ zeroes in the set of x. Because $x = 1$ is also a zero of $Q(x) - P(x)$, we know that $Q = P$ is a contradiction, and that $W_k = a$. With that, we can comfortably compute the bound, and finish our proof.

$$f_k^{\mathrm{Lan}} \leq 0.824\sqrt{\frac{4n}{(2+\frac{1}{c}}} \leq 1.648\sqrt{cn}$$