Math 270a: Numerical Linear Algebra

Instructor: Randolph E. Bank

Fall Quarter 2017

Homework Assignment #6
Due Wednesday, November 8, 2017

Exercise 6.1. Let A be a symmetric $n \times n$ matrix, and let q be a vector satisfying $\|q\|_2 = 1$ which approximates an eigenvector v of A. Assume $\|v\|_2 = 1$. Let $v_1, v_2, \ldots, v_n$ by a set of orthonormal eigenvectors of A corresponding to eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. For convenience, suppose the v_i are ordered such that $v_1 = v$. The approximate eigenvector q can be expressed in terms of the v_i as

$$q = c_1 v_1 + c_2 v_2 + \cdots c_n v_n$$

a. Show that

$$\sum_{k=1}^{n} c_k^2 = \|q\|_2^2 = 1$$

b. Show that

$$\sum_{k=2}^{n} c_k^2 \le \|v - q\|_2^2$$

c. Derive an expression for the Rayleigh quotient $\rho = q^t A q$ in terms of the coefficients c_i and the eigenvalues λ_i.

d. Show that

$$|\lambda_1 - \rho| \le C\|v - q\|_2^2$$

where

$$C = \max_{2 \le k \le n} |\lambda_k - \lambda_1|$$

(Hint: $\lambda_1 = \sum_{k=1}^{n} \lambda_1 c_k^2$).

Thus, if $\|v - q\|_2 = O(\epsilon)$, then $|\lambda_1 - \rho| = O(\epsilon^2)$.

Exercise 6.2. Let

$$A_0 = A = \begin{pmatrix} 8 & 1 \\ -2 & 1 \end{pmatrix}.$$

a. Find an orthogonal Q_1 (a plane rotation) and an upper triangular R_1 such that $A_0 = Q_1 R_1$. Calculate $A_1 = R_1 Q_1$. Notice that A_1 is closer to upper triangular form than A_0, and the main diagonal of A_1 approximates the eigenvalues of A better than those of A_0 do.

b. Now perform one step of the QR algorithm with Rayleigh quotient shift. Compare your result with part a. Which is better?

Exercise 6.3. Do this exercise using Matlab. We will study the power method and its variants. Let T be the $n \times n$ symmetric, positive definite tridiagonal matrix

$$T = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

a. The eigenvalues of T are given by

$$\lambda_k = 2 - 2\cos\left(\frac{k\pi}{n+1}\right)$$

$1 \le k \le n$. Each eigenvalue is simple and the corresponding eigenvectors are

$$v_k = \sqrt{\frac{2}{n+1}} \begin{pmatrix} \sin\left(\frac{k\pi}{n+1}\right) \\ \sin\left(\frac{2k\pi}{n+1}\right) \\ \sin\left(\frac{3k\pi}{n+1}\right) \\ \vdots \\ \sin\left(\frac{nk\pi}{n+1}\right) \end{pmatrix}$$

By forming the product Tv_k, verify directly that $Tv_k = \lambda_k v_k$.

b. Now let $n = 31$. Apply 10 iterations of the power method using the starting vector $q_0^t = [1, 1, \cdots, 1]$. Rescale so that the first component of q_i is 1, ($q_i^t = [1, *, *, \cdots, *]$ where "$*$" is just a generic entry). Since we know the eigenvalues of T, compute the ratio $\lambda_{30}/\lambda_{31}$, which will give you a good idea of the rate of convergence. Compute $\|q_j - \alpha v_{31}\|_2$, $0 \le j \le 10$, where $\alpha = 1/(v_{31})_1$.

c. Now we will try the same problem using shifted inverse iteration. Let $\rho = \lambda_{31} + 10^{-6}$. Apply 3 iterations of the inverse power method using $T - \rho I$, again starting from $q_0^t = [1, 1, \cdots, 1]$. Rescale as in part b. Compute the ratio the ratio $(\rho - \lambda_{31})/(\rho - \lambda_{30})$ to estimate the rate of convergence. Compute $\|q_j - \alpha v_{31}\|_2$, $0 \le j \le 3$, where $\alpha = 1/(v_{31})_1$.

d. The inverse power method is able to compute intermediate eigenvectors with equal ease. To see this, let $\rho = \lambda_{16} + 10^{-6}$. Apply 3 iterations of the inverse power method using $T - \rho I$, starting from $q_0^t = [1, 1, \cdots, 1]$. Normalize as before. Compute αv_{16} where $\alpha = 1/(v_{16})_1$ for purposes of comparison. Compute the ratio the ratio $|(\rho - \lambda_{16})/(\rho - \lambda_{17})|$. Compute $\|q_j - \alpha v_{16}\|_2$, $0 \le j \le 3$, where $\alpha = 1/(v_{16})_1$.

e. The last experiment should illustrate the performance of the Rayleigh quotient iteration for different starting vectors. Do 4 steps of the Rayleigh quotient iteration starting form the vector $q_0^t = [1, 1, \cdots, 1]$. Then repeat the experiment, except choose the starting vector $q_0^t = [1, -1, 1, -1, \cdots, -1, 1]$.