Homework set 1 — MATH 397C — Spring 2022

Due in class on ~~Thursday Feb. 10~~ Tuesday Feb. 15.

Problem 1: Suppose that $\mathbf{A}$ is a real symmetric $n \times n$ matrix. Let $\{\mathbf{v}_j\}_{j=1}^n$ denote an orthonormal set of eigenvectors so that $\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j$ for some numbers λ_j. Let us use an ordering where $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$. Draw a starting vector $\mathbf{g}$ from a Gaussian distribution, and then generate a sequence of vectors by setting $\mathbf{x}_1 = \mathbf{A}\mathbf{g}$, and then $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}$ for $k = 2,\, 3,\, 4,\, \dots$.

Set $\beta = |\lambda_2|/|\lambda_1|$ and assume throughout this problem that $\beta < 1$.

(a) Assume that $\lambda_1 = 1$. Prove that as $k \to \infty$, the vectors $\{\mathbf{x}_k\}_{k=1}^\infty$ converge to $c\mathbf{v}_1$ for some constant c.

(b) What is the speed of convergence of $\{\mathbf{x}_k\}_{k=1}^\infty$?

(c) For the more general case where λ_1 is not necessarily 1 (but $\beta < 1$ still), consider the vectors

$$\mathbf{y}_k = \frac{1}{\|\mathbf{x}_k\|}\mathbf{x}_k.$$

When does the sequence $\{\mathbf{y}_k\}_{k=1}^\infty$ converge? If it converges, what is the limit point? What is the speed of convergence? *Note: You are encouraged to* ***try*** *to prove your assertions for part (c). This is voluntary, however, for everyone except the person handing in a reference solution.*

Hint: *The starting vector* $\mathbf{g}$ *has an expansion* $\mathbf{g} = \sum_{j=1}^{n} c_j\, \mathbf{v}_j$. *Every coefficient* c_j *is nonzero with probability one. (In fact, each* c_j *is random number drawn from a standard normal distribution.*

Problem 2: In this exercise, we explore blocked versions of power iteration where we run power iteration on several vectors at once. Such techniques often go under the name "subspace iteration" in the literature.

Let $\mathbf{A}$ be an $n \times n$ matrix, and assume that $\mathbf{A}$ has an eigenvalue decomposition $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$. In a simple version of subspace iteration, we start with ℓ Gaussian vectors $\{\mathbf{g}_j\}_{j=1}^{\ell}$ that we collect into a matrix $\mathbf{G} = [\mathbf{g}_1,\, \mathbf{g}_2,\, \dots,\, \mathbf{g}_\ell] \in \mathbb{R}^{n\times\ell}$. The iteration then computes the matrices

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{A}\mathbf{G}, \\ \mathbf{Y}_k &= \mathbf{A}\mathbf{Y}_{k-1}, \qquad k = 2,\, 3,\, 4,\, \dots \end{aligned}$$

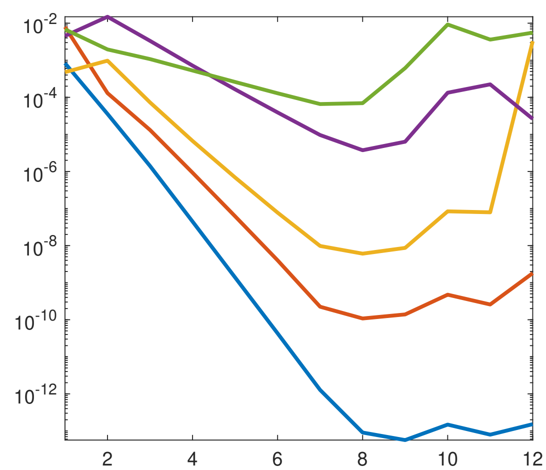
The idea is that as k grows, the space $\mathrm{col}(\mathbf{Y}_k)$ will successively better align with the space spanned by the dominant k eigenvectors. After k steps, we orthonormalize the vectors $\mathbf{Y}_k$ to form an orthonormal matrix $\mathbf{Q}_k$ whose columns are an ON basis for $\mathrm{col}(\mathbf{Y}_k)$. Practically, we execute this via a QR factorization $\mathbf{Y}_k = \mathbf{Q}_k\mathbf{R}_k$. We then compute the ℓ eigenvalues of the matrix $\mathbf{B}_k = \mathbf{Q}_k^*\mathbf{A}\mathbf{Q}_k$, and claim that these ℓ values typically converge to the k largest (in modulus) eigenvectors of $\mathbf{A}$. The figure on the left shows a simple Matlab code that sets up a test matrix, executes the iteration, and computes the corresponding errors. The figure on the right shows the errors computed in the array `ERR`.

```
%%% Set random number generator, and various problem parameters.
rng(0)
m     = 20; % Size of the matrix.
ell   = 5;  % Block size in block power iteration.
nstep = 12; % Number of steps of power iteration.

%%% Generate a test matrix.
dd = 2.^(-linspace(0,m-1,m))';  % The vector of eigenvalues.
V  = randn(m) + eye(m);
A  = V*diag(dd)*inv(V);

%%% Execute the power iteration
Y     = randn(m,ell);
ERR   = NaN*ones(20,ell);
for i = 1:nstep
  Y        = A*Y;
  [Q,~]    = qr(Y,0);
  ee       = eig(Q'*A*Q);
  ERR(i,:) = min(abs(dd*ones(1,ell) - ones(m,1)*ee'));
end

semilogy(ERR)
```



(a) The reason convergence quickly stops is that round-off errors aggregate at every step. Insert the line `[Y,~] = qr(Y,0)` immediately after the line with the `for` command. Then rerun the experiment. Hand in a figure of the new convergence plot.

(b) Estimate numerically the slopes of the five lines in the graph you generate in (a). Form an hypothesis of the speed of convergence as a function of the eigenvalues and of ℓ and k. (Observe that $\lambda_j = 2^{-j}$ in this example.)

(c) Replace the line `V = randn(m) + eye(m);` by `[V,~] = qr(randn(n));`. Observe that this results in a matrix $\mathbf{A}$ that is symmetric, since then $\mathbf{V}^{-1} = \mathbf{V}^*$. Run the experiment again. Hand in the graph of the errors, and form a new conjecture on the speed of convergence.

(d) Choose some other matrix, and some other value of ℓ, and see if your hypotheses from (b) and (c) hold up. Turn in one representative example.

(e) *[Optional, except for reference solution:]* Prove that if all computations were executed in exact arithmetic, then the modification described in (a) would make no difference to the estimates of the eigenvalues.

Problem 3: Exercise 35.4 from Trefethen & Bau.

Problem 4: Exercise 35.6 from Trefethen & Bau.

Problem 5: Exercise 38.2 from Trefethen & Bau.

Problem 6: Exercise 38.4 from Trefethen & Bau — only parts (a) and (c).

Problem 7: Exercise 38.5 from Trefethen & Bau.