Advanced Numerical Analysis Homework 6

Michael Nelson

Throughout this homework, $\|\cdot\|$ denotes the ℓ_2-norm and we also denote $\langle \cdot, \cdot \rangle$ to be the standard Euclidean product on $\mathbb{C}^n$ (i.e. $\langle x, y \rangle = x^\top y$ for all $x, y \in \mathbb{C}^n$). We also denote $\varepsilon = \varepsilon_{\text{mach}}$ to be the machine coefficient.

1 Problem 1

Exercise 1. Let $P \in \mathbb{C}^{m \times m}$ be a nonzero projector. Show that $\|P\| \geq 1$, with equality if and only if P is an orthogonal projector.

Solution 1. Choose $x \in \mathbb{C}^m$ such that $Px \neq 0$. Then note that

$$\|Px\| = \|P(Px)\| \leq \|P\|\|Px\|$$

implies $\|P\| \geq 1$.

Now we show that P is an orthogonal projector if and only if $\|P\| = 1$. First assume that P is an orthogonal projector. Then

$$\begin{aligned}
\langle Px, x - Px \rangle &= \langle Px, Px \rangle - \langle Px, x \rangle \\
&= \langle P^2 x, x \rangle - \langle Px, x \rangle \\
&= \langle Px, x \rangle - \langle Px, x \rangle \\
&= 0
\end{aligned}$$

for all $x \in \mathbb{C}^m$. Thus by the pythagorean theorem, we have

$$\|x\|^2 = \|Px\|^2 + \|x - Px\|^2 \geq \|Px\|^2$$

for all $x \in \mathbb{C}^m$. In particular, this implies $\|P\| \leq 1$, which implies $\|P\| = 1$ since we've already show $\|P\| \leq 1$. Conversely, suppose $\|P\| = 1$. We wish to show that P is an orthogonal projection. It suffices to show that $P = P^*$. In other words, we want to show

$$\langle Px, y \rangle = \langle x, Py \rangle$$

for all $x, y \in \mathbb{C}^m$. We claim that $(\ker P)^\perp = \operatorname{im} P$. Indeed, suppose $x \in (\ker P)^\perp$. Then note that $x - Px \in \ker P$ since P is a projector, thus by the pythagorean theorem, we have

$$\begin{aligned}
\|x\|^2 &\leq \|x\|^2 + \|Px - x\|^2 \\
&= \|Px\|^2 \\
&\leq \|x\|^2.
\end{aligned}$$

It follows that $Px = x$, thus $x \in \operatorname{im} P$. This shows that $(\ker P)^\perp \subseteq \operatorname{im} P$. In fact, we must have equality since they have the same dimension:

$$\begin{aligned}
\dim((\ker P)^\perp) &= m - \dim(\ker P) \\
&= m - (m - \dim(\operatorname{im} P)) \\
&= \dim(\operatorname{im} P).
\end{aligned}$$

Thus we have $(\ker P)^\perp = \operatorname{im} P$ as claimed. To see why this implies P is self-adjoint, note that

$$\begin{aligned}
0 &= \langle x - Px, Py \rangle \\
&= \langle x, Py \rangle - \langle Px, Py \rangle \\
&= \langle x, Py \rangle - \langle Px, Py - y + y \rangle \\
&= \langle x, Px \rangle - \langle Px, Py - y \rangle - \langle Px, y \rangle \\
&= \langle x, Py \rangle - \langle Px, y \rangle
\end{aligned}$$

for all $x, y \in \mathbb{C}^m$.

2 Problem 2

Exercise 2. Review our analysis of the bound on the relative forward error of singular value computation by using a backward stable eigenvalue algorithm for $A^\top A$. That is,

$$\frac{|\widetilde{\sigma}_k - \sigma_k|}{\sigma_k} \le O((\sigma_1/\sigma_k)^2 \varepsilon), \tag{2.1}$$

where $\widetilde{\sigma}_k = \sqrt{\widetilde{\lambda}_k}$ where $\widetilde{\lambda}_k$ denotes the kth largest eigenvalue of $A^\top A$. Instead, if we use a backward stable eigenvalue algorithm for $H := \begin{pmatrix} 0 & A^\top \\ A & 0 \end{pmatrix}$, show that the relative forward error of singular value computation would be bounded by $O((\sigma_1/\sigma_k)\varepsilon)$, assuming that square root computation is exact. Explain the advantage of the new error bound.

Solution 2. First, let us recall why one obtains (2.1) when computing the singular values σ_k of A via the eigenvalues λ_k of $A^\top A$. When λ_k is computed stably, then we expect errors of order

$$|\widetilde{\lambda}_k - \lambda_k| = O(\varepsilon \|A^\top A\|) = O(\varepsilon \|A\|^2). \tag{2.2}$$

Thus when we square root to obtain σ_k, we have errors of order

$$|\widetilde{\sigma}_k - \sigma_k| = O(|\widetilde{\lambda}_k - \lambda_k| / \sqrt{\lambda_k}) = O(\varepsilon \|A\|^2 / \sigma_k), \tag{2.3}$$

where (2.3) is equivalent to (2.1) since $\|A\|^2 = \sigma_1$. On the other hand, note that the singular values of A are precisely the absolute values of the eigenvalues of H. Indeed, if $A = U\Sigma V^\top$ is an SVD for A, then

$$H = \begin{pmatrix} 0 & A^\top \\ A & 0 \end{pmatrix} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{pmatrix} \begin{pmatrix} V & V \\ U & -U \end{pmatrix}^\top$$

is an eigenvalue decomposition of H. Thus if we calculate the eigenvalues of H, then we obtain errors of the form (2.2), and since these are already the singular values of A (in absolute value), we do not need to take square roots and obtain an error of the form (2.3). The adavantage for the new error bound is that the singular values which are far away from σ_1 (i.e. the smallest singular values of A) are computed to a higher relative accuracy than the previous algorithm. This is because the square term in $(\sigma_1/\sigma_k)^2$ is removed.

3 Problem 3

Exercise 3. Read the introduction to the Golub-Kahan-Lanczos method, at http://www.netlib.org/utk/people/JackDongarra/etemplates and the uploaded code implementation HW6 GKLsvds.m.

1. Give a general description of the functionality of GKL; describe the main difference between the original GKL and the code.
2. Download the zipped file HW6 pics.zip, unzip it, load the first jpeg file by

   ```
   picA = double(imread('picA.jpg'));
   ```

   and run

   ```
   rk = 160;
   tic; [Us1,Ss1,Vs1] = HW6 GKLsvds(picA(:,:,1),rk); toc;
   tic; [Us2,Ss2,Vs2] = HW6 GKLsvds(picA(:,:,2),rk); toc;
   tic; [Us3,Ss3,Vs3] = HW6 GKLsvds(picA(:,:,3),rk); toc;
   tic; [U1,S1,V1] = svd(picA(:,:,1),0); toc;
   tic; [U2,S2,V2] = svd(picA(:,:,2),0); toc;
   tic; [U3,S3,V3] = svd(picA(:,:,3),0); toc;
   ```

   Then, run MATLAB's command whos to see the memory used by picA, and by Us1,Vs1,Us2,Vs2,Us3 and Vs3 all together. Compare the timing used for computing and the memory used for storing the full and partial SVD of this picture. (Note: we are competing MATLAB code with the built-in C/FORTRAN code in timing, and our timing should improve considerably if our GKL code is in C/FORTRAN).

3. Finally, run MATLAB's command

```
picAh = zeros(size(picA));
picAh(:,:,1) = Us1*Ss1*Vs1';
picAh(:,:,2) = Us2*Ss2*Vs2';
picAh(:,:,3) = Us3*Ss3*Vs3';
disp([norm(picAh(:,:,1) - picA(:,:,1),'fro')/norm(picA(:,:,1),'fro') ...
norm(picAh(:,:,2) - picA(:,:,2),'fro')/norm(picA(:,:,2),'fro') ...
norm(picAh(:,:,3) - picA(:,:,3),'fro')/norm(picA(:,:,3),'fro')]);
figure(1); imshow(uint8(picA)); axis equal;
figure(2); imshow(uint8(picAh)); axis equal;
```

Repeat the above procedure for the other three pictures. Make some general comments on the computation and use of partial SVD for compressing images. In particular, give an estimate of the arithmetic cost of this partial SVD and full SVD applied to an image of dimension m-by-n, in a form of $O(-)$. For a given rank rk $\ll$ min{m, n}, how does the cost of partial SVD compared to full SVD as min{m, n} increases? In a recent development by random sketching, fulll orthogonalization of rk vectors of elements m or n (rk $\ll$ min{m, n}) needs only O(rk3) + O(max{m, n}rk) flops. Compare the cost of partial and full SVD if such a fast orthogonalization can be used.

4. Search the title of each artwork, the name of the artist, the approximate year of creation, and the current location of the artwork. Info of 4 paintings qualifies full extra credit. Do your own research, instead of using others' findings, even for this leisure problem.

Solution 3. 1. The GKL algorithm is performed in the first phase of SVD computation. In particular, we apply Householder reflectors alternately from left and right to the matrix A until we arrive at a bidiagonal matrix $UAV^\top$. The total cost of GKL is approximately

$$4mn^2 - \frac{4}{3}n^3 \text{ flops.}$$

In Dr. Xue's code, supplementary calculations are incorporated in order to preserve stability. For instance, the for-loops on lines 31 and 39 seem to guarantee the orthogonality of matrices U and V.

2.

3.

4. We give brief description for each of the paintings below:

1. The Last Judgment: Michelangelo, 1536-1541. A large fresco depicting the Second Coming of Christ and final judgment. Located at Sistine Chapel, Vatican City.
2. The Last Supper: Leonardo da Vinci, c. 1495-1498. Jesus and his disciples at the final meal before his crucifixion. Located at Santa Maria delle Grazie, Milan.
3. Dance at Bougival: Pierre-Auguste Renoir, 1883. A couple dancing outdoors at a party. Located at Museum of Fine Arts, Boston.
4. The Night Watch: Rembrandt van Rijn, 1642. A group of civic guards led by Captain Frans Banning Cocq. Located at Rijksmuseum, Amsterdam.
5. Girl with a Pearl Earring: Johannes Vermeer, c. 1665. A young girl wearing a turban and a pearl earring. Located at Mauritshuis, The Hague.
6. The Potato Eaters: Vincent van Gogh, 1885. A group of peasants sharing a meal of potatoes. Located at Van Gogh Museum, Amsterdam.
7. The Starry Night: Vincent van Gogh, 1889. A swirling night sky over the village of Saint-Rémy-de-Provence. Located at Museum of Modern Art, New York.