MATH 170A (WI21, Dumitriu)

Homework 5

Jiahong Long

[redacted]

18 February 2021

1. *On the Frobenius norm*

   (a) *Norm equality*

We perform a column-vector decomposition of A:

$$A = \begin{bmatrix} a_{:,1} & a_{:,2} & a_{:,3} & \dots & a_{:,n} \end{bmatrix}$$

By the definition of matrix multiplication, it follows immediately from this decomposition that

$$UA = \begin{bmatrix} Ua_{:,1} & Ua_{:,2} & Ua_{:,3} & \dots & Ua_{:,n} \end{bmatrix}$$

Recall that the Frobenius norm of a matrix M is defined as

$$||M||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |m_{i,j}|^2}$$

which is in turn equivalent to the sum of the Euclidean norm of its columns

$$= \sqrt{\sum_{i=1}^{n} ||m_{:,i}||_2^2}$$

Given that U is $n \times n$ orthogonal, we know that the norm of a vector in $\mathbb{R}^n$ is the same as its image under U. Then it immediately follows that

$$||m_{:,i}||_2 = ||Um_{:,i}||_2 \qquad \rightarrow \qquad ||m_{:,i}||_2^2 = ||Um_{:,i}||_2^2$$

for all $i = 1, 2, \dots, n$. Then, substituting the above into the formula for the definition of the Frobenius norm immediately yields equality:

$$\begin{aligned} ||M||_F &= \sqrt{\sum_{i=1}^{n} ||m_{:,i}||_2^2} \\ &= \sqrt{\sum_{i=1}^{n} ||Um_{:,i}||_2^2} \end{aligned}$$

Then, by definition,

$$= ||UM||_F$$

Let $M = A$, and

$$||A||_F = ||UA||_F$$

as required. ■

(b) *Norm equality redux*

We begin by transposing both sides of the equation

$$\begin{aligned} A &= AV \\ A^T &= (AV)^T \\ &= V^T A^T \end{aligned}$$

Define $A' = A^T$, and

$$A' = V^T A'$$

Since V is orthogonal, we know that V^T is also orthogonal. The proof at this point proceeds in an identical fashion to part (a); substitute A' for A and V^T for U. ■

2. *Frobenius norm and singular values*

(a) *Verification*

We begin with the singular value decomposition of A:

$$\begin{aligned} A &= U\Sigma V^T \\ ||A||_F &= ||U\Sigma V^T||_F \end{aligned}$$

Recall that U and V^T are both orthogonal matrices in $\mathbb{R}^n$. It follows, therefore, that, applying the results of (1a-b), we have

$$\begin{aligned} ||U\Sigma V^T||_F &= ||\Sigma V^T||_F \\ &= ||\Sigma||_F \end{aligned}$$

Using the definition of the Frobenius norm, we rewrite this as a function of entries in Σ as

$$||\Sigma||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |\Sigma_{i,j}|^2}$$

Since Σ is nonzero only on the main diagonal, where the entries are the singular values of A, this is given by

$$= \sqrt{\sum_{i=1}^{n} |\sigma_i|^2}$$

Singular values are all positive, so we may drop the absolute value to obtain

$$||A||_F = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$$

which is as required. ■

(b) *Condition number*

We begin by finding $||A^{-1}||_F$ for arbitrary $A \in \mathbb{R}^n$. To begin, note that the "SVD" of A^{-1} is given by the expression

$$A^{-1} = V\Sigma^{-1}U^T$$

where Σ^{-1} is the element-wise reciprocal of Σ. Since the Frobenius norm of matrices is unchanged through multiplication with orthogonal matrices, we may state

$$||A^{-1}||_F = ||V\Sigma^{-1}U^T||_F = ||\Sigma^{-1}||_F$$

and since Σ^{-1} has nonzero values strictly on the diagonal equal to the reciprocal of the (necessarily positive) singular values, this is in turn given by

$$||A^{-1}||_F = \sqrt{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}}$$

Recall that the condition number is defined as $\kappa(A) = ||A||\ ||A^{-1}||$. Under the Frobenius norm, this is given by

$$||A||_F||A^{-1}||_F$$

and with the above results, we have

$$\boxed{\sqrt{\left(\sum_{i=0}^{n} \sigma_i^2\right)\left(\sum_{i=0}^{n} \frac{1}{\sigma_i^2}\right)}}$$

3. *$n \times m$ matrix*

(a) *SVD calculation*

We know that

$$A = U\Sigma V^T$$

It follows immediately that

$$A^T = V\Sigma U^T$$

and as such

$$\begin{aligned}
A^TA &= V\Sigma U^T\left(U\Sigma V^T\right) \\
&= V\Sigma^T \overset{I_n}{\cancel{U^TU}}\Sigma V^T \\
&= V\Sigma^T\Sigma V^T \\
&= V \underbrace{\begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3^2 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_m^2 \end{bmatrix}}_{\hat{\Sigma_2}} V^T
\end{aligned}$$

where the middle matrix $\hat{\Sigma_2}$ is just the reduced singular value matrix with all entries squared.

$$= \boxed{V\hat{\Sigma_2}V^T}$$

We posit that this in fact the singular value decomposition of A^TA because V and its transpose are both orthogonal in $\mathbb{R}^n$, and $\hat{\Sigma_2}$ is a diagonal matrix with all nonnegative entries in increasing order.

(b) *Conditioning equality*

Spectral norm proof
Let $A = U\Sigma V^T$, and per part (a) let $A^TA = V\hat{\Sigma_2}V^T$. Recall that the spectral matrix norm of a matrix is given exactly by its largest singular value σ_1. It therefore follows that

$$||A||_2 = \sigma_1$$

and since the SVD of A^TA is given by $V\hat{\Sigma_2}V^T$, the largest singular value of A^TA must be the leftmost entry on the top row of $\hat{\Sigma_2}$. From the previous problem this is given as exactly σ_1^2, and we have

$$||A^TA||_2 = \sigma_1^2 = (\sigma_1)^2 = ||A||_2^2$$

as required. ■

Condition number proof
We begin by finding the condition number of A under the spectral norm. Recall that the spectral matrix norm is just the largest singular value of that matrix

$$||A||_2 = \sigma_1$$

It follows that the spectral norm of the matrix *inverse* must be the largest singular value of the matrix inverse, which is just the reciprocal of the smallest singular value of the matrix. Here, as A is full-rank, we therefore have

$$||A^{-1}||_2 = \frac{1}{\sigma_n}$$

It follows that

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}$$

We now find an expression for $|| \left(A^TA\right)^{-1} ||_F$:

$$\begin{aligned}\left(A^TA\right)^{-1} &= \left(V\hat{\Sigma_2}V^T\right)^{-1} \\ &= V\hat{\Sigma_2}^{-1}V^T\end{aligned}$$

Recall that the spectral norm simply returns the largest singular value of the matrix. Then we must have

$$|| \left(A^TA\right) ||_2 = ||V\hat{\Sigma_2}^{-1}V^T||_2$$

Note that σ_n is nonzero since A is full-rank. Furthermore, since $\hat{\Sigma_2}$ is diagonal, its inverse is obtained by taking the reciprocal of each entry along the diagonal; it follows that the largest entry in the inverse must be the reciprocal of the smallest singular value. We thus have

$$\begin{aligned} &= \frac{1}{|\sigma_n^2|} \\ &= \frac{1}{\sigma_n^2}\end{aligned}$$

It follows immediately that

$$\kappa_2(A^T A) = ||A^T A||_2 \; || \left(A^T A\right)^{-1} ||_2 = \frac{\sigma_1^2}{\sigma_n^2}$$

and since

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}$$

we immediately have $\kappa_2(A^T A) = \kappa_2(A)^2$ as required. ■

4. *Computations*

   (a) *Rank*
   Using elementary row operations, we obtain

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 3 & 6 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 1 & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

   Thus the rank of A is $\boxed{1}$.

   (b) *Size*

   Since the rank of A is 1, there must only be one nonzero singular value. It follows immediately that

$$\boxed{\Sigma_r : 1 \times 1 \quad \text{and} \quad U_r : 3 \times 1 \quad \text{and} \quad V_r^T : 1 \times 2 \;\rightarrow\; V_r : 2 \times 1}$$

   (c) *Submatrices*

   We know that there is just one singular value. Additionally, it's trivially true that

$$\underbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}}_{u} \underbrace{\begin{bmatrix} 1 \\ 2 \end{bmatrix}^T}_{v} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{bmatrix}}_{A}$$

   It must follow that to obtain the reduced SVD of A, we need only normalise u to obtain U_r, and normalise v to obtain V_r. (We note here that in this case, Σ_r is vacuously equal to the scalar matrix containing the product of $||u||_2$ and $||v||_2$.) It follows that we immediately have

$$\boxed{U_r = \frac{\sqrt{14}}{14} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \text{and} \qquad V_r = \frac{\sqrt{5}}{5} \begin{bmatrix} 1 \\ 2 \end{bmatrix}}$$

   Furthermore, simple algebra yields that $\sigma_1 = \sqrt{70}$ and $\Sigma_r = [\sqrt{70}]$.

5. *Low-rank approximation*

   (a) *Line-by-line*

   The code in question is attached:

```
% load image
A = imread('street2.jpg');
A = rgb2gray(A);
B = double(A);

% compute SVD
size(B)
r = rank(B)
[U,S,V] = svd(B);

% approximate image
ranks = [1 2 4 8 16 32 64 r];
l = length(ranks);

for i = 1:l
  % compute rank i approximation
  k = ranks(i);
  approxB = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
  approxA = uint8(approxB);

  % plot images
  figure(1)
  subplot(2,4,i)
  imshow(approxA);
  title(sprintf('rank %d approximation',k))
end
```

The line-by-line description is as follows (skipping over comments and empty newlines):

i. The image is loaded to an integer array.
ii. The image is converted to grayscale.
iii. The image is cast to a floating-point representation.
iv. The size of the image is obtained in pixels.
v. The numerical rank of the image matrix is calculated.
vi. The full SVD of the image matrix is calculated.
vii. An array of proposed ranks is created.
viii. The number of proposed ranks is stored.
ix. We begin looping over each of the proposed ranks.
x. We get the current proposed rank k.
xi. The proposed rank is used to generate the rank-k approximation of B by multiplying out the truncated SVD of B.
xii. The low-rank approximation of B is converted to an integer array.
xiii. The current `Figure` window is set to 1 (so we use the same window for plotting).
xiv. We set the plot location to the ith subplot in the 2×4 grid of subplots.
xv. The image is plotted.
xvi. The image is given the title `rank k approximation` where k is the current rank.

Hence we have a line-by-line description of the algorithm.

(b) *Algorithm*

The algorithm loads an image as a matrix representing the grayscale values of each pixel in the image. It then calculates seven different low-rank approximations of the matrix, and then plots the results as pixel arrays, along with the original (full-rank) grayscale array.

(c) *Original vs. approximation*

Low-rank approximations will use less storage and be more compact, since they allow us to represent "most" of the image using submatrices of U, Σ, and V. However, computing the actual full SVD in addition to the low-rank approximation is computationally intense, since (modern) digital images tend to have high pixel counts, leading to large matrices which are computationally difficult to deal with. For instance, a modestly sized 4000 pixel square image took about 20 seconds just to compute the rank on my computer, and even longer for the full SVD.

(d) *Reasonable k*

Heuristically (and informed, to some extent, by data science), each of the singular values σ_i can be correlated with how much of the information entropy (in data science, "variance") is "captured" by adding the ith dimension. It stands to reason that a reasonable k can be found by fixing some proportion of the total "variance" (ie. sum of all the singular values), and finding the k where the sum of the first kth singular values meets or exceeds that proportion.

Alternatively, for a more quantitative (albeit computationally heavy) approach, one can fix some ε, and run logarithmic binary search on $k = 1, 2, 3, \ldots, \operatorname{rank}\{A\}$ to find the value of k for which all entries in the rank-k approximation of A are within ε of A. We note that logarithm binary search is a valid search method: as informed by principal component analysis in data science, scree plots of singular values necessarily decrease in a logarithmic fashion.