BI 501, Fall 2018: Mathematical Foundations of Bioinformatics

Module 2.2: First Problem Set

1.)[1] Consider the matrix

$$A = \begin{bmatrix} -2 & 11 \\ -10 & 5 \end{bmatrix}.$$

(1) Determine (on paper, i.e., by hand) a real SVD of A in the form $A = U\Sigma V^T$.

(2) List the singular values, the left singular vectors and the right singular vectors of A. Draw a careful, labeled picture of the unit circle in $\mathbb{R}^2$ and its image under the linear transformation A, together with the singular vectors, with their coordinates marked. You may try the function `eigshow` in MATLAB for a computerized version of this for comparison. See section 10.3 of *Numerical Computing with MATLAB* (in the Canvas site for this module), cited as *NCM* henceforth.

(3) Find A^{-1} using the SVD.

(4) Find the eigenvalues λ_1, λ_2 of A.

(5) Verify that $|\det A| = \sigma_1\sigma_2$.

(6) What is the area of the ellipse onto which A maps the unit disk of $\mathbb{R}^2$?

2.)[2] Consider a tiny data matrix X of size 10×2 called Height+Weight-PS1.xlsx. Analyze this matrix along the lines of *NCM*, pp.21-24 in the copy in Canvas of chapter X, where the "height-weight" matrix A is defined on p. 22. The command for importing such a data file to a MATLAB array can be found here:

https://www.mathworks.com/help/matlab/ref/importdata.html

In general, one can search https://www.mathworks.com/help/matlab/index.html for relevant commands in MATLAB. A fairly detailed guide to this calculation is in the file

Coursenotes-Yaounde+BI501-2017.2.pdf

in section 2.7, pp. 18-22.

[1]From *Numerical Linear Algebra*, by L. Trefethen and D. Bau, III. The full text is available through the UM library.

[2]After *NCM*, section 10.11.

3.)[3] The "Numerical Computing with MATLAB" file `imagesvd.m` helps you investigate the use of PCA[4] in digital image processing. You can also use your own photographs.[5] For an m-by-n color image in JPEG format, the statement

$$\mathtt{X} = \mathtt{imread}('\mathtt{myphoto.jpg}');$$

produces a three-dimensional m-by-n-by-3 array X with m-by-n integer subarrays for the red, green, and blue intensities. It would be possible to compute three separate m-by-n SVDs of the three colors. An alternative that requires less work involves altering the dimensions of X with

$$\mathtt{X} = \mathtt{reshape}(\mathtt{X}, \mathtt{m}, 3 * \mathtt{n})$$

and then computing one m-by-$3n$ SVD.

(a) The primary computation in `imagesvd` is done by

$$[\mathtt{V}, \mathtt{S}, \mathtt{U}] = \mathtt{svd}(\mathtt{X}', 0)$$

[Remember that X' in MATLAB is what we have called X^T, the transpose.]
How does this compare with `[U,S,V] = svd(X,0)`?

(b) How does the choice of approximating rank affect the visual qualities of the images? There are no precise answers here. Your results will depend upon the images you choose and the judgments you make.

[3] From *NCM*, chapter X. Cf. Canvas site.

[4] PCA = Principal Components Analysis. This is the same as using the low rank approximations to the data matrix, adding the rank one matrices $\sigma_j u_j \cdot v_j^T$, ordering them according to descending size of σ_j.

[5] If you have access to the MATLAB Image Processing Toolbox, you may want to use its advanced features. However, it is possible to do basic image processing, which is all we want here, without the toolbox.