APM120, Homework #4

Applied Linear Algebra and Big Data

Last updated: Sunday 5th February, 2023, 20:49

Assigned Feb 21, **due Friday, March 3, 7:00 pm**, via gradescope, **in pdf $\leq$ 20Mb, $\leq$ 30 pages.**

Principle Component Analysis, Singular Value Decomposition

Show all steps in all calculations explicitly. Attach code used, well documented, and relevant plots and Matlab/python output, attaching code and figures *immediately following* the relevant question solution. A code printout is not a substitute for complete solutions, your solution should stand alone without the Matlab/python code or output. See needed python preliminaries at end of this HW[1]. It is fine to use Matlab/python unless a hand calculation (using only a simple calculator) is required *in orange*. For all questions, make sure you show all steps as if you are doing the problem by hand, and do not use library functions unless explicitly allowed in the question. Make sure you can do all calculations using no more than a hand-held calculator. See the end-note for guidelines and examples of hand-calculations.[2]

1. **PCA:** (**A**) Consider the following data that represent the deviations from mean prices of two products in 8 different stores Calculate by hand element $C_{2,1}$ of the covariance matrix C, what does it represent? Then calculate the covariance entire matrix, principal components, Calculate by hand element $t_{1,4}$ of the expansion coefficient matrix T, what does it represent? Calculate the full expansion coefficient matrix, and the percent of the variance explained by each mode. (**B**) Come up with a scenario that corresponds to what the data show and analyze and interpret the results in economic/ commercial terms. (**C**) Briefly explain what "variance explained by each PCA mode" means. (**D**) Outline the derivation and explain the formula for the explained variance.

```
F=[   -6,    47,   -42,   -66,    35,    21,     9,     2;
      89,   -64,     4,    56,    20,    11,   -59,   -57];
```

2. **PCA:** Consider the following data, representing price anomalies of $M = 5$ stocks, on $N = 1300$ different days, and constructed using three column vectors,

```
% Matlab
%
N=1300; t=1:N;
V1=[1;2;0;-1.1;0];
V2=[0;-1;0;-0.8;0];
V3=[1.5;0;0;-0.6;1.5];
F=3*V1*cos(t/5) ...
  +2*V2*cos(t/3)+1.5*V3*cos(t/7);
```

```
# python:
from numpy.matrixlib.defmatrix import matrix
N=1300; t=np.arange(0,N); t.shape=(1,N);
V1=np.array(matrix("[1;2;0;-1.1;0]"));
V2=np.array(matrix("[0;-1;0;-0.8;0]"));
V3=np.array(matrix("[1.5;0;0;-0.6;1.5]"));
F=3*V1@np.cos(t/5) \
  +2*V2@np.cos(t/3)+1.5*V3@np.cos(t/7);
```

(**A**) Plot the elements of F as M time series — showing only some of the time range so that the oscillations are legible — on the same set of axes, with an appropriate legend (see PCA_small_data_example_using_covariance.m/py). (**B**) Calculate, show and interpret the covariance matrix, verify that it is symmetric, and interpret each of its entries on and above the diagonal (a brief sentence for each). (**C**) Calculate and interpret the eigenvalues λ_i and principal components $\mathbf{u}_i$. (**D**) Calculate and plot the expansion coefficients t_{jn} (where $\mathsf{T} = \mathsf{U}^T\mathsf{F}$) as M line plots. (**E**) Check and demonstrate explicitly what is the number k of PCAs that are required to reconstruct the data to a

very good approximation $\mathsf{F}_{\text{reconst}} = \sum_{i=1}^{k} t_{jn}\mathbf{u}_i$, and explain why this makes sense given the way the data were constructed.

3. **SVD basics:** (**A**) Calculate by hand the SVD of `A=[-10,-9,8;-0,4,3]`: Calculate U, V and Σ via the eigenvectors and eigenvalues of AA^T **or** $\mathsf{A}^T\mathsf{A}$, and explain which one should be used in this case and why. Explain also why we should not use both matrices to calculate U and V. (**B**) Show that the decomposition works in the sense that $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^T$. Compare your results to Matlab's `[U,S,V]=svd(A)` or python's `U,Sigma,V=np.linalg.svd(A)` and analyze and justify the differences, if any. Explain why there may be such differences in general.

4. **Image compression:** using SVD: (**A**) Read the image `coronavirus.jpg`, convert it to a double precision matrix and plot it using

```
% Matlab
A=double(rgb2gray(imread('coronavirus.jpg'))); A=A-mean(A(:));
figure(1); clf; imagesc(A); colormap(gray); colorbar
```

```
# python:
import imageio; from skimage.color import rgb2gray;
A=imageio.imread('coronavirus.jpg');
A=rgb2gray(A); A=np.double(A); A=A-np.mean(A);
fig1=plt.figure(1,dpi=300);
plt.imshow(A); plt.set_cmap('gray');plt.colorbar()
```

   Calculate the SVD of the image matrix, plot the $\log_{10}$ of singular values from large to small. Use the plot to predict how many SVD modes you expect to be required for a successful reconstruction of the image. (**B**) Reconstruct and plot the image using 5, 25, 50 and 100 modes, vs the full image; calculate the explained variance for each reconstruction. (**C**) How many modes give a sufficiently accurate reconstruction for putting this image on your web page? What is the saving in storage?

5. **Matrix norm and condition number:** (**A**) Use SVD to calculate the norm and condition number of the matrix A below and compare to the values given by Matlab's `norm(A)` and `cond(A)` or python's `np.linalg.cond(A,2)` and `np.linalg.norm(A,2)`. (**B**) Find the vector $\mathbf{x}$ for which $||\mathsf{A}\mathbf{x}||/||\mathbf{x}|| = \text{norm}(\mathsf{A})$. (**C**) Find the solution to $\mathsf{A}\mathbf{x} = \mathbf{b}$, and then again to $\mathsf{A}\mathbf{x} = \mathbf{b} + \delta\mathbf{b}$. Calculate the relative error to the solution to $\mathsf{A}\mathbf{x} = \mathbf{b}$, compare to the relative perturbation to $\mathbf{b}$. (**D**) ***** optional extra credit:** given the matrix A_1 below, find a rhs $\mathbf{b}$ of norm 1 and a perturbation $\delta\mathbf{b}$ of norm 0.01 that maximize the relative error $|\delta\mathbf{x}|/|\mathbf{x}|$. How big is this maximum error amplification?

```
A=[  1.97,   3.59,  -0.177,   0.726;
    -4.06,   0.879,  -0.513,    3.4;
     0.411,   5.16,  -0.308,   0.756;
     3.34,  -2.41,  -0.0974,   3.33];
b      =[ 0.476;  0.268; 0.741; -0.391];
delta_b=[-0.744; -0.153; 0.626;  0.176];
```

```
A1=[  2.98,   0.383,   2.11,  -0.942;
      0.938,   2.14,   3.81,  -1.98;
     -0.649,   0.775,   1.16,  -0.416;
      1.49,   0.118,   3.61,   0.483];
```

6. *****optional extra credit:** Consider a vector u_i, $i = 1, \ldots, N$. Its second order

derivative may be written as,

$$\left.\frac{d^2u}{dx^2}\right|_i = (u_{i+1} - 2u_i + u_{i-1})/\Delta x^2.$$

Write the operator $d^2u/dx^2 - \gamma u$ as a matrix A times the vector $(u_1, \ldots, u_N)^T$. The boundary conditions are "periodic", this mean that when u_{N+1} is encountered, it is replaced by u_1, similarly, $u_0 = u_N$. Calculate the condition number of A and plot it for $N = 10, \ldots, 30$ for $\gamma = 2$ and $\Delta x = 1/N$.

* [What's the point of ***optional extra credit challenge problems: apart from the fun of doing them, they may bring the total score of this HW assignment up to 110%, making up for problems you may have missed in this or other HW assignments...]

Python preliminaries & notes

1 python commands within the HW assume you have first used the followings: `import numpy as np; from numpy import linalg; import scipy as scipy; from scipy import linalg; import matplotlib.pyplot as plt; import matplotlib;`
Input a matrix A, column vector **b** and row vector **c** into python in the form
`A=np.array([[a11,a12,a13],[a21,a22,a23],[a31,a32,a33]]; b=np.array([[b1],[b2],[b3]]); c=np.array([[c1,c2,c3]]);` or convert Matlab arrays given in HW directly to python arrays using, e.g., `A=np.array(np.matrixlib.defmatrix.matrix('[1 2 3; 4 5 6]'));`

2 **Hand calculations**, in which you are asked to use only a simple hand-held calculator, are required only when we want to make sure you understand exactly what each step of an algorithm is doing. These also prepare you for the quizzes that involve similar hand calculations. **How much work to show?** Just don't use scratch paper, show all steps that you actually use for the hand calculations, but no more, carrying out calculations to **three significant digits**. Trust the graders to be reasonable, they were students in the course last year. **Examples: (i)** If you are asked, *not* in orange, to calculate the LU decomposition of a matrix, you may use Matlab/python as in `A(2,:)=A(2,:)-A(1,:)*(A(2,1)/A(1,1))` etc, but you may not use a library routine as in `[L,U,P]=lu(A)` except for checking your results. **(ii)** If required to calculate by hand the element $C_{2,3}$ of a matrix product $\mathsf{C} = \mathsf{AB}$, you need to explicitly multiply using a hand calculator the second row of A with the third column of B. You may not use Matlab/python to calculate `C=A*B` and then take the needed element from that product, nor to calculate `C23=A(2,:)*B(:,3)`.