CS 369 Assignment 1 2015

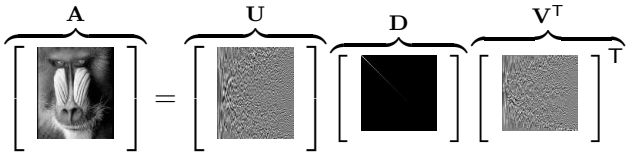
Due: Thursday, April 2 2015, 8:00 p.m.

Marked out of 40 points. 10% of the total course marks

Before you make a start, read over the whole assignment including the requirements section at the end.

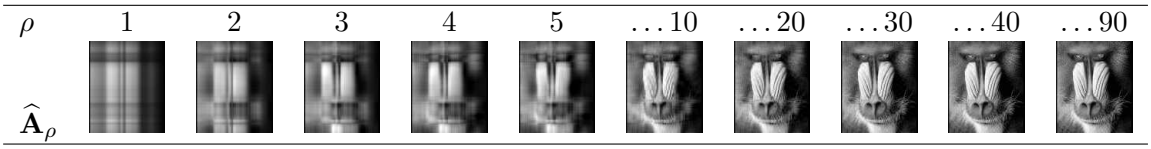
Problem 1: SVD based data compression *[15 points]*. For this problem, you will a need 120×100 pixel grayscale image of your own face (similar to the baboon picture below). You can use any digital camera and image processing software to capture a colour image of own face, crop and scale it down to 120 pixels (height) $\times$ 100 pixels (width), and convert it to a grayscale image. This will be used as a 120×100 rectangular matrix $\mathbf{A}$.

a. Perform the SVD $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T}$ of your facial image and illustrate the obtained results in your report: e.g.



For visualisation, values of the matrix elements U_{ij}, D_{ii}, and V_{ij} should be mapped linearly to the range $[0, 255]$, e.g. $255\frac{U_{ij}-U_{\min}}{U_{\max}-U_{\min}}$, where $U_{\max} = \max_{i,j} U_{ij}$ and $U_{\min} = \min_{i,j} U_{ij}$.

b. Approximate $\mathbf{A}$ with ρ singular values and columns of $\mathbf{U}$ and $\mathbf{V}$ for $1 \leq \rho \leq 100$ and illustrate the obtained results in your report: e.g.



To visualise every approximate matrix $\widehat{\mathbf{A}}_\rho$, values that are less than 0 should be set to 0 and the values larger than 255 should be set to 255.

c. Find the absolute approximation errors and the level of compression (see below) with respect to the original image for different values of ρ. Tabulate these in your report, e.g.:

ρ	1	2	3	4	5	...10	...20	...30	...40	...90
Max error	130	139	119	115	120	107	80	38	31	0.5
Mean error	31	23	20	16	14	10	6.5	4.5	2.9	0.04
Compression,%	98.2	96.3	94.5	92.6	90.8	81.6	63.2	44.8	26.3	n/a

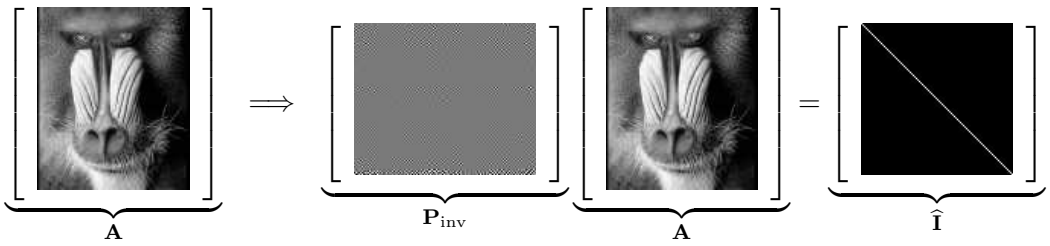
The compression achieved by an approximation is measured by the ratio of the number of real numbers used in the approximation to the number of real numbers used in the original. The number of real numbers used in the original is 120×100, so if k_ρ is the number used in the ρth approximation, the compression achieved is $1 - \frac{k_\rho}{12000}$. Since the compression should always be positive, there is no compression for large ρ.

Describe what negative values for compression correspond to and find the smallest value of ρ for which compression is negative.

d. Determine the most compressed approximation that is acceptable (to you) in terms of visual quality, explain why you chose this one and state the corresponding compression and mean error.

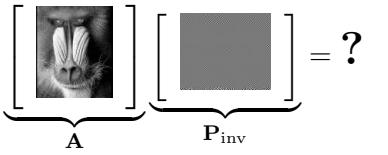
Problem 2: Pseudoinverse matrix *[5 points]*. For this problem, use the same image matrix as you used for question 1.

a. For the grayscale image $\mathbf{A}$, form a pseudoinverse 100×120 matrix $\mathbf{P}_{\text{inv}}$ that behaves like an inverse of $\mathbf{A}$ in that the product $\mathbf{P}_{\text{inv}}\mathbf{A} = \widehat{\mathbf{I}}$ yields the 100×100 identity matrix $\mathbf{I}$. Illustrate the obtained results in your report, e.g.:



b. Assess quantitatively the component-wise errors $\mathbf{E} = \mathbf{I} - \widehat{\mathbf{I}}$ between the computed, $\widehat{\mathbf{I}}$, and the ideal identity matrix $\mathbf{I}$ with 1s on the main diagonal and 0s otherwise. Present these results in your report, giving the error range, mean and standard deviation.

c. Form and illustrate the 120×120 matrix $\mathbf{B} = \mathbf{A}\mathbf{P}_{\text{inv}}$:



Describe the matrix $\mathbf{B}$ and explain why $\mathbf{B}$ has this form. Compare $\mathbf{B}$ to the matrix $\widehat{I}$ you obtained in part a.

Hint: Consider a 120×100 matrix formed by the product $\mathbf{B}\mathbf{A} = \mathbf{A}\mathbf{P}_{\text{inv}}\mathbf{A}$.

Problem 3: Principal components to find geographic structure *[15 points]*. You will need to download the `snps.txt` data file and the R script `readandplot.R` from the Resources page of the course website.

This question is based on the geographic analysis of genetic snp data from Novembre et al 2008, Nature, doi:10.1038/nature0733 discussed in Lecture 8.
The data matrix stored in `snps.txt` is a binary matrix with 100 rows and 2734 columns. Each row corresponds to an individual person and each column corresponds to a position in the individuals genome which varies across the population, known as a *single nucleotide polymorphism* or SNP. So each row is a trial and each column is a measurement (note that this is the reverse of the PCA setup in the notes).
The data comes from a geographically structured population with an unknown number of subpopulations. Your task here is to perform a principal components analysis of the data to determine the number of populations and allocate individuals to each population.

a. Write code to read in the data matrix and perform a principal components analysis of it (remember to centre the matrix). For your report, print out the first 5 entries of each of the first 5 principal component vectors, and print out the singular values corresponding to each of the 100 principal components in order.

b. Project each data vector along the first two principal components to get a 100×2 matrix $\mathbf{L}$ in which entry L_{ij} corresponds to the amount of the ith individual in the direction of the jth principal component. Write this data matrix to a white space delimited file called `locations.txt` and print the matrix in your report.

c. Using R (available on all lab computers or free to download) and the script `readandplot.R`, read in and plot $\mathbf{L}$. From your plot, determine how many subpopulations there are in the data set and describe a rule for partitioning the space into the different subpopulations. Include your plot in your report, the number of subpopulations, your partitioning rule and a list of the individuals (corresponding to row numbers) in the largest subpopulation.

Problem 4: Root finding [*5 points*].

a. Derive and implement the Newton's root finding algorithm to find zeros of the following functions:

$$\begin{aligned} f(x) &= 2x^3 - 15x^2 + 36x - 23 \\ g(x) &= \exp(0.1x) - \exp(-0.4x) - 1 \end{aligned}$$

For both functions, start Newton's algorithm at $x = 0$ and terminate after it converges to a stationary point such that the candidate root points x found at two successive steps do not differ up to the fourth significant fractional digit, i.e. $|x_i - x_{i-1}| \leq 0.0001$.

b. Show the results of each step until convergence. For example, for the function $f(x) = \log(x) - 1 + \exp(-x)$, you would show:

Step i	x_i	$f(x_i)$	$\frac{df(x)}{dx}\Big\vert_{x=x_i}$	x_{i+1}
0	1.00000	−0.63212	0.63212	2.00000
1	2.00000	−0.17152	0.36466	2.47034
2	2.47034	−0.01109	0.32025	2.50496
3	2.50496	−0.00005	0.31753	2.50511
4	2.50511	−0.00000	0.31752	2.50511

Requirements: Use Java to write your code, with files Problem1.java, Problem2.java etc. For matrix algebra, use the Java Matrix algebra package `JAMA`.
The JAMA jar file, a Java class for image input/output in pgm (grayscale), and R scripts and data for Problem 3 are available on the course resources page.

The image input subroutine `readFilePGM_PPM` allows you to input a given grayscale pgm-image. The image is placed into a linear byte array of size mn where m and n denote the numbers of image rows and columns, respectively.

- The byte at position $p = jn + i$ gives the grey level of the grayscale image point $a_{i,j}$ with the row j and column i.

To output a pgm image, you should form first the corresponding byte array and then write it to a file using the subroutine `saveFilePGM_PPM`.

Submission: Submit a PDF of your report and corresponding source code before or on Thursday, April 2, 2014, 8:00 pm via the ADB `https://adb.auckland.ac.nz/`.

Please complete your work on time as extensions will be granted only in special circumstances.

Brief marking scheme for Problems 1–4: You will be marked on the both your report and your code. Name code associated with each question sensibly, like Problem1.java.

- Your code must run and produce correct answers. Any instructions for running your code should be included. It should also be sensibly commented so the marker can understand your logic.
- Your report should describe your implementations of algorithms for solving Problems 1 – 4, present numerical and pictorial outputs produced by your algorithms, provide your evaluations of these algorithms and discussion of the results.
- Your report should be coherent and address the problems in full sentences. Do not just dump data to the report.