CPSC-540 Machine Learning 2007

Homework # 1

Due Thursday Feb 1 in class.

NAME:____

Signature:______________________________________

STD. NUM: ____________________________________

General guidelines for homeworks:

You are encouraged to discuss the problems with others in the class, but all write-ups are to be done on your own.
Homework grades will be based not only on getting the "correct answer," but also on good writing style and clear presentation of your solution. It is your responsibility to make sure that the graders can easily follow your line of reasoning.
Try every problem. Even if you can't solve the problem, you will receive partial credit for explaining why you got stuck on a promising line of attack. More importantly, you will get valuable feedback that will help you learn the material.
Please acknowledge the people with whom you discussed the problems and what sources you used to help you solve the problem (e.g. books from the library). This won't affect your grade but is important as academic honesty.
When dealing with Matlab exercises, please attach a printout with all your code and show your results clearly.

1. **(Learning Matlab)**

   (i) Write a MATLAB m file that plots the following 2D Gaussian density function:

$$p(x,y) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}$$

   for x and y in the range [-3,:0.1:3]. You will need to learn the commands `help`, `figure`, `clf`, `meshgrid` and `surf`. Your answer should consist of the `surf` plot and the code.

   (ii) Load the special MATLAB function called `peaks`. You can do this by typing `[x,y,z] = peaks;`. Use the commands `contour` and `clabel` to generate a countour plot of the function `peaks`. Repeat the exercise using the functions `contourf` and `contour3`. You should hand in your code and a <u>single</u> figure with 4 subplots (you'll need to learn the command `subplot`) consisting of the `surf` plot and the 3 countour plots.

   (iii) Now we'll try to visualise a function in 4D! In particular, use the command `slice` to generate a plot of the 3D contours of the following Gaussian:

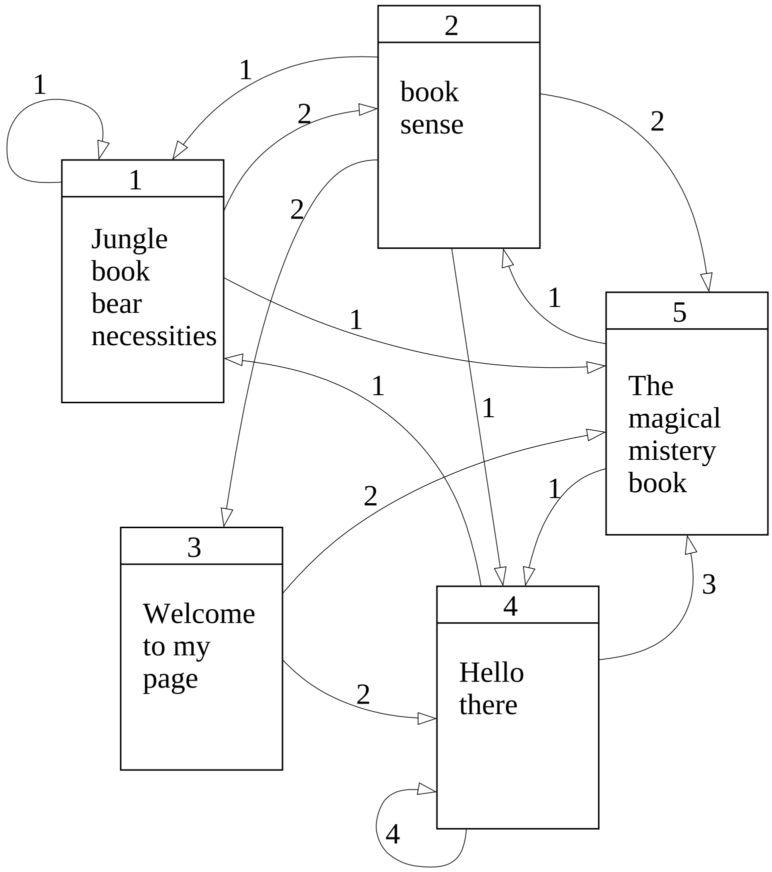
$$p(x,y,z) = \frac{1}{(2\pi)^{3/2}} e^{-\frac{1}{2}(x^2+y^2+z^2)}$$

   for x, y and z defined in the range [-3,:0.1:3].Use the commands `xlabel`, `ylabel` and `zlabel` to label the axes appropriately. Hand in the code and plot.

2. **(Eigenproblems)** Compute the eigenvalues and eigenvectors of the following matrix by hand and using matlab:

$$\mathbf{A} = \begin{pmatrix} -2 & 2 & -3 \\ 2 & 1 & -6 \\ -1 & -2 & 0 \end{pmatrix}$$

3. **(Google)** Given the following web graph:



(i) Construct the graph transition matrix. Write down the appropriately normalised version of this matrix and call it **G**.

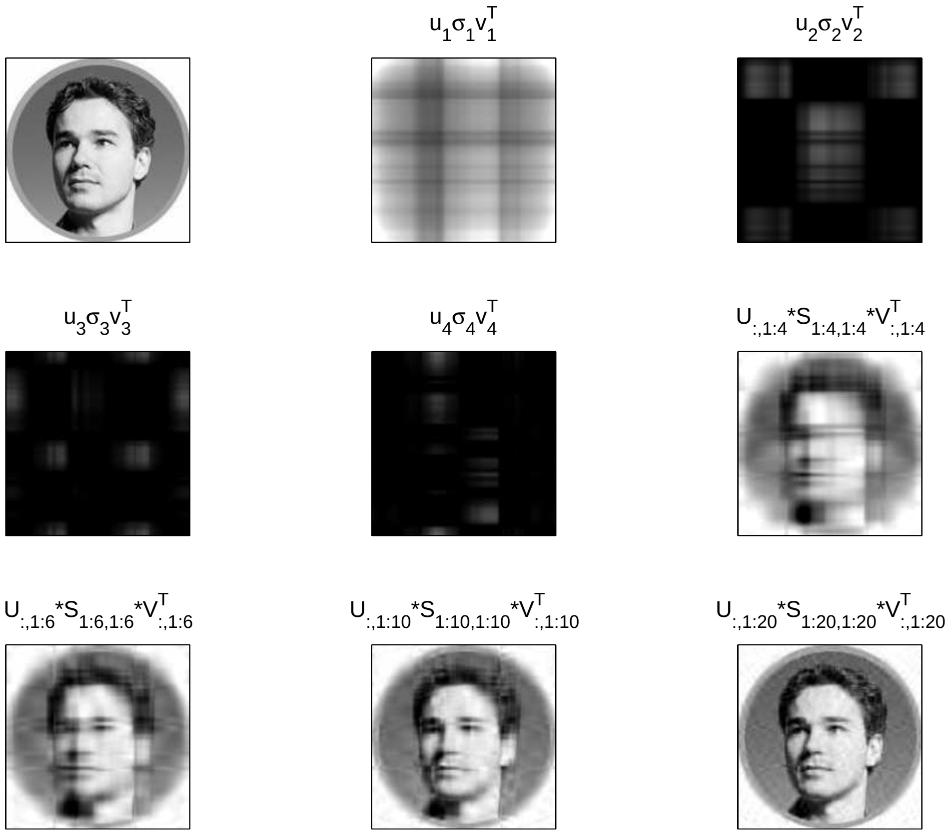
(ii) Using MATLAB, compute the rank vector using the PageRank algorithm described in class.

(iii) Write down a MATLAB script that plots the individual entries of the rank vector on the vertical axis and the number of iterations on the horizontal axis. This allows us to see how the algorithm is converging and determine how many iterations are necessary. Hand in the code and plot. Label the axes appropriately.

(iv) What are the eigenvalues of $\mathbf{G}^{\mathbf{1000}}$.

(v) For the query "book", what pages (in order and using exact word matching) does this simple search engine return?

4. **(Image compression)** Repeat the image compression exercise presented in class. However, this time I want you to use your own photo. Initially you should use any image package to generate a jpg colour image of yourself that is 100 by 100 pixels. Then load this image to matlab using the command `imread`. Convert the image from rgb colour space to gray-scale using `rgb2gray`. Convert the gray-scale image to double precision using the command `double`. Finally, apply the SVD and generate the following plot (except it should have your own picture!).



What is the compression gain achieved in each of the last four images? What is the corresponding compression error?

5. **(Text retrieval)**

This exercise teaches you how to build a simple text search engine and text compression scheme. The file `matrix.dat` contains a matrix of documents by words for the documents that appear in the file `documents.ps`. That is, according to the lecture notes $A = matrix^T$, i.e. you have to transpose *matrix*. The size of A is $m \times n$. The list of the m words (in order) appears in file `literals.dat`.

(i) Compute the SVD of A and plot the eigen-spectrum. That is, plot the diagonal of the matrix Σ.

(ii) Truncate the SVD to 3D. That is, $k = 3$. Create a plot with 2 subplots using the matlab command `subplot`. In the first subplot do `image(A)`. In the second subplot, plot the image $U_k \Sigma_k V_k^T$. What do you conclude from comparing the two plots? What's the compression gain? What's the compression error?

(iii) Using the first document as a query ($q = A(:,1)$), what are the angles, in degrees, between the projected query in 3D and the 9 projected documents? Explain how this query could help you debug your code.

(iv) If the query is the word "*abducted*" ($q(23, 1) = 1$), what document would be retrieved? What is the smallest angle in this case? Does this make sense when you look a the text in the actual documents?

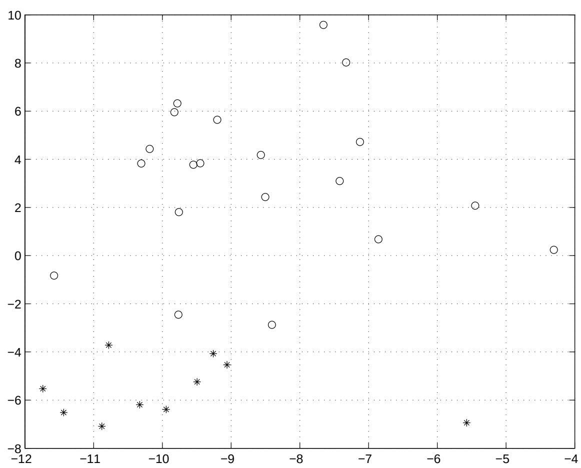
6. **(PCA)**

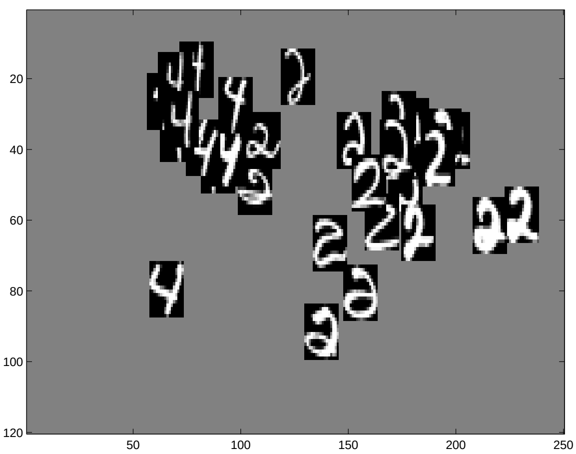
In this homework, you need to implement the image visualization PCA tool discussed in class. Your task is to modify the following piece of code. Whenever you encounter a ??? sign, replace it with the appropriate code.

```
load twosfours;
A = data;
% A has 30 rows corresponding to 30 16x16 images.
% The first 20 rows are images of 2s and the last
% 10 rows are images of 4's

[U,S,V] = ???;      % Compute SVD of A.
Uk = U(:,???);
Sk = S(???,???);
pc = ???;          % Principal components of A.

figure(1)
% Plot principal components.
clf;
plot(pc(1:20,1),pc(1:20,2),'or',???,???,'*b');
grid;
```

```
% Plot images on a background of size 120 by 250.
a = zeros(120,250);
% a is the background image on which small images are placed.
% now we scale the components to be integeres for indexing:
pc = pc + abs(min(min(pc)))*ones(size(pc)) + ones(size(pc));
pc = round(10*pc)
[r,c] = size(pc);
% now we add the individual images to the background image:
for i=1:r,
  a(pc(i,1):pc(i,1)+15,pc(i,2):pc(i,2)+15) = reshape(???,???,16)';
end;

figure(2)
% Visualization:
clf;
colormap('gray')
imagesc(a)
print -deps pca2
```

Plot answers:





Hand in all your code.