Math 365 **Homework 8** **Due December 12, 2016**
Grady Wright

Note: Here are two of the homework problems. If there is a 93% or higher response rate on the course evaluation by 2:00pm on Wednesday December 9, then I will not assign any more problems. If there is not, then there will be three more problems.

1. (**Image compression**, 30 pts) The computer representation of an image is just a matrix of pixel values. For a monochrome image, the value is a single intensity of the pixel ranging between black and white. For a color image, there are three numbers stored for each pixel, which are typically the red, green, and blue intensities of the pixel, but other color schemes are possible. A color image of size $m \times n$ therefore consists of 3 matrices of size $m \times n$. Usually the intensities are stored as 8 bits, resulting in integers between 0 and 255.

   As discussed in class, the singular value decomposition (SVD) can be used to compress images. For monochrome images stored in a matrix A, the SVD technique consists first decomposing A into the matrices U, Σ, and V^T, (i.e. $A = U\Sigma V^T$), where Σ is a diagonal matrix of the singular values, and U and V^T are orthonormal matrices of size $m \times m$ and $n \times n$, respectively. The rank p matrix $A^{(p)}$ that best approximates A in the 2−norm sense is then given by

   $$A^{(p)} = \sum_{j=1}^{p} U_j(\Sigma_{j,j} V_j^T),$$

   where U_j and V_j are the jth columns of U and V respectively. In terms of images, $A^{(p)}$ is now a "compressed" version of of the original image A insofar as the space required to store the relevant columns of U and (ΣV^T) is less than that of the image as a whole. This space is $p \times (m + n)$. If no lossless compression methods are applied to U and (ΣV^T) such as Huffman coding, then the compression level of the SVD is $p(m + n)/(mn)$.

   For color images, the above procedure can also be applied. However, it is useful to first reshape the 3 intensity matrices of size $m \times n$ into one matrix of size $m \times 3n$. When displayed, this reshaped matrix looks like one monochrome image with the 3 color planes side by side. The compression ratio for the SVD approach is now $R = p(m + 3n)/(3mn)$ and the compression level is $1 - R$.

   (a) Download the jpeg image file `bsubronco.png` from the course webpage. Load the image into Matlab and then display the image using the commands

   ```
   >>A = imread('bsubronco.png');
   >>imagesc(A);
   ```

   Note that the image must be stored in your current working Matlab directory. Upon execution of the first command the variable `A` will contain a $m \times n \times 3$ matrix of the image as discussed above. Reshape `A` into a $m \times 3n$ matrix using Matlab's `reshape` command. Display the reshaped matrix with the `imagesc` command as given above. Use the gray colormap to make the image grayscale.

   (b) Using the SVD compression technique described above, create images with compression ratios of 1%, 5%, 10%, 20%, 30%, 40%, 50%, and 75%. For each of these compression ratios display the compressed image with the `imagesc` command. The displayed image should be the combined red, green, blue (RGB) color image, not three panels of the same image in grayscale. Comment on the quality of the image as p increases. When do you start seeing an increase in p having litte effect? Turn in the plots at the various compression levels.

Some notes:

i. To use MATLAB's `svd` command on `A` you have do first convert the entries of `A` to double precision floating point. You should also rescale the intensity values to be between 0 and 1. This is done in one command as follows:
```
>>A = double(A)/255;
```

ii. The SVD approximation to `A` may cause individual intensity values to fall outside of the range 0 to 1 even if the original intensity was within this range. For display purposes, you may want to fix this with the following command:
```
>>Ap = min(1,max(0,Ap));
```
where `Ap` contains the p-compressed image.

iii. To reshape the `Ap` image to be of size $m \times n \times 3$ for plotting, use the command:
```
>>Ap = reshape(Ap,[m n 3])
```

(c) Plot the singular values of `A` (i.e. the diagonal entries of Σ). Use a semilogy plot. Mark the compression ratios from part (b) on this plot.

2. (**Image classification**, 30 pts) As discussed and demonstrated in class, the singular value decomposition (SVD) can be used to classify handwritten images. In this problem you will work with the MNIST data set of handwritten images to determine how the SVD classification system depends on the number of basis vectors used in the classification. In particular, you will write code to determine how the accuracy of the predictions increases (or decreases) as the number of basis vectors changes from 1 to 40. You should test the accuracy against the 10,000 test images that are part of the MNIST data set. The result of your code should be a plot similar to that shown in Figure 1. Include this plot in your write-up as well as your modified code.

Hint: You should start this problem by modifying the `mnist_classification_ex.m` file on the course webpage. The parameter in this code that you want to vary is `k`. You need to write a code that loops over the values $k = 1, 2, \ldots, 40$, and for each k it then loops over all the 10,000 images in the test image database computing the predicted number and recording if it is correct. From these numbers you will determine how the accuracy changes with increasing k. Be prepared for your code to take a long time to run. You may wish run the code once, save the data to a file, and then just load this saved file in to produce the plot in your final homework assignment.

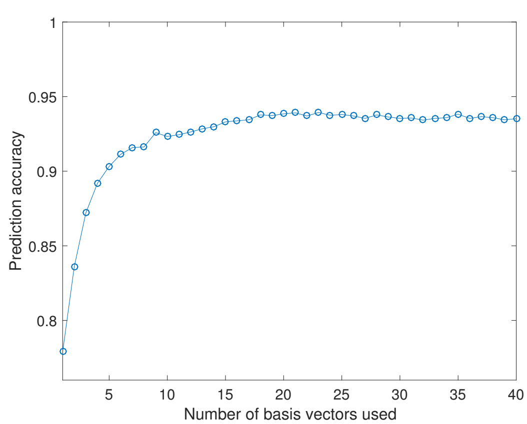

Figure 1: Accuracy of the SVD prediction scheme for the MNIST test images..