CS168, Spring 2015

Mini-Project #5

Due by 11:59 PM on Tuesday, May 5th.

Instructions

- You can work individually or with one partner. If you work in a pair, both partners will receive the same grade.
- If you've written code to solve a certain part of a problem, or if the part explicitly asks you to implement an algorithm, you must also include the code in your pdf submission. The code for all parts should go in an appendix. This is a change from the first few assignments, where the code was embedded in each part.
- Make sure plots you submit are easy to read at a normal zoom level.
- Detailed submission instruction can be found on the course website (`http://cs168.stanford.edu`) under the "Coursework - Assignment" section. If you work in pairs, only one member should submit all of the relevant files.
- **Reminder:** No late assignments will be accepted, but we will drop your lowest mini-project grade when calculating your final grade.

Part 1: SVD for image compression

Description:

Download `http://web.stanford.edu/class/cs168/p5_image.gif`. It is a 480×342, black and white drawing of Alice conversing with a Cheshire Cat. We will think of this image as a 480×342 matrix, with each black pixel represented as a 0, and each white pixel represented as a 1. We will observe this matrix under various approximations induced by its SVD.

Exercises:

(a) Before running SVD, describe qualitatively what you think the rank 1 approximation given by the SVD might look like (when viewed as a 480×342 pixel image). We are not grading for correctness, and would be surprised if your guess matched reality. We are looking for an educated guess along with some reasoning.

(b) Run SVD and recover the rank k approximation, for $k \in \{1, 3, 10, 20, 50, 100, 150, 200, 300, 342\}$. In your assignment, include the recovered drawing for $k = 150$. Note that the recovered drawing will have pixel values outside of the range $[0, 1]$; feel free to either scale things so that the smallest value in the matrix is black and the largest is white (default for most python packages and matlab), or to clip values to lie between 0 and 1.

(c) Why did we stop at 342?

(d) How much memory is required to efficiently store the rank 150 approximation? Assume each floating point number takes 1 unit of memory, and don't store unnecessary blocks of 0s.

(e) Bonus: Details of the drawing are visible even at relatively low k, but the gray haze / random background noise persists till almost the very end (you might need to squint to see it at $k = 300$). Why is this the case?

(f) Bonus: Say we inverted the colors before performing SVD, namely represented black pixels as 1, and white pixels as 0, and then inverted back afterwards. Would anything change? Why or why not?

Part 2: SVD for classification

The dataset at `http://web.stanford.edu/class/cs168/p5_dataset.tar` has 5 pictures of Barack Obama and 5 pictures of Michelle Obama, each 180×125. Unlike the last section, here we think of each image as a single $180 \cdot 125 = 22500$ pixel array, formed by e.g. concatenating the rows of the image together. Let O be the 10×22500 matrix induced by the 10 Obama images.

Exercises:

(a) Each pixel is actually a vector of three integers between 0 and 255, one for each of red, green, blue. Normally one would just use 3 columns per pixel and turn O into a $10 \times 3 \cdot 22500$ array, but for simplicity we'll just use the red channel (namely, the first coordinate of the three). Center the data. What is the image corresponding to the first component of the SVD (namely, the first row of V^T)? Feel free to display it in grayscale. In the computer vision community these components are often referred to as "eigenfaces".

(b) Project the 10 images onto the first column of V, where $O = USV^T$. How might one separate the images of Barack from the images of Michelle?

Part 3: SVD for matrix completion

Let T be a 10×100 matrix where each entry is chosen independently and uniformly from $[-1, 1]$. Let R be a 50×10 matrix where each entry is chosen independently from the Gaussian distribution $\mathcal{N}(0, 1)$. Let $M = RT$.

Exercises:

(a) Plot the singular values of M from highest to lowest. At some point you should see a precipitous drop. Why is there such a drop? Just write the explanation, no need to include the plot.

(b) M has 5000 entries. We will simulate a scenario where 10% of its entries are missing, and there is some measurement error in the rest. Choose 500 random entries of M and pretend they are "missing": replace each of these missing entries with zero. For each of the remaining 4500 entries, add an independent Gaussian drawn from $\mathcal{N}(0, .1)$. Call this new matrix $\widehat{M}$. Use SVD to get a rank 10 approximation M' to $\widehat{M}$. What is the Frobenius norm of $M' - M$? How about $\widehat{M} - M$? Did the SVD help?

(c) In (b), we replaced "missing" values with 0's. Was 0 a reasonable choice? If so, why, and if not, what might have been a better approach to populating the missing entries before computing the SVD?