Mini-Project #5  
Due by 11:59 PM on Tuesday, May 3rd.

Instructions

- You can work individually or with one partner. If you work in a pair, both partners will receive the same grade.
- Detailed submission instruction can be found on the course website (http://cs168.stanford.edu) under “Coursework - Assignment” section. If you work in pairs, only one member should submit all of the relevant files.
- Use 12pt or higher font for your writeup.
- Make sure the plots you submit are easy to read at a normal zoom level. All plots must be included in the writeup.
- 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. Also, make sure to mark all pages relevant to the question on Gradescope. Keep variable names consistent with those used in the problem statement, and with general conventions. No need to include import statements and other scaffolding, if it is clear from context. Use the `verbatim` environment to paste code in LaTeX.

```python
def example():
    print "Your code should be formatted like this."
```

- Reminder: No late assignments will be accepted, but we will drop your lowest assignment grade when calculating your final grade.

Part 1: Election data

In this question you’ll be playing with data from the 2016 US Presidential Primaries. You will need 2 files `county_facts_dictionary.csv` and `county_results.csv`, they can be downloaded from the course website. We are providing data for approximately the first 13 states which voted (up to Super Tuesday). In the file `county_results.xls` you can find the fraction of votes received by 5 Republican candidates- Donald Trump, Ted Cruz, Marco Rubio, Ben Carson and John Kasich and 2 Democratic candidates- Hillary Clinton and Bernie Sanders in 1008 counties across 13 states. The file also contains the demographic information for these counties. `county_facts_dictionary.csv` contains descriptions for the keywords used to refer to the demographics in `county_results.xls`.

a. (0 points) Make sure you can import the given datasets into whatever language you’re using. If you’re using MATLAB, you can import the data using the GUI. Also, make sure you understand what each row and column represents.

b. (5 points) Let the matrix $M$ be the matrix of voting data and demographics for each county. Each row of the matrix represents a county. The first 7 columns would be the fraction of votes obtained by each candidate in the county and the next 51 columns would be the 51 demographic indicators. Your matrix should have 1008 rows and 58 columns. Normalize your data so that each column has mean 0 and unit L2 norm, call this normalized matrix $\tilde{M}$. Now find the SVD of the data to express $\tilde{M} = UDV^T$. Plot the singular values $D$. Does the matrix seem to be low rank?
c. (5 points) We’ll first interpret the right singular vectors (columns of \( V \)). Take the right singular vector \( v_1 \) corresponding to the largest singular value. Note that the entries of this vector correspond to the 7 candidates and 51 demographics, we’ll call these our 58 variables. Sort the coordinates of \( v_1 \) by absolute value, and observe what the top 15 variables and their (signed) entries are. What information does the top right singular vector roughly capture?

d. (5 points) Do the same procedure for the second right singular vector. It should capture some voting patterns for the Democratic race. What demographics seem to favor Clinton and what favors Sanders?

e. (5 points) We’ll now project our data onto fewer dimensions to visualize it. Remember that the right singular vectors are the principal components of the data matrix \( M \) and the left singular vectors are the projection of each county along the principal components. Project each county onto the first and second right singular vectors. Hence county \( i \) will have coordinates \( \{U(i,1), U(i,2)\} \). Create a scatter plot by plotting each county by its 2-D coordinates, and labeling it as ‘Clinton’ or ‘Sanders’ based on who won that county (the Democratic/Republican winner of a county is the Democratic/Republican candidate receiving the highest number of votes). Make sure to include a legend in your plot. What do you observe? Similarly, now project all counties onto the 1st and 3rd right singular vectors and label the points as ‘Trump’, ‘Cruz’ or ‘Rubio’ based on the winner (you don’t need to plot counties where none of the 3 win). What do you observe?

f. (5 points) You would expect counties in the same state to be similar demographically. Let’s project counties along two of the top right singular vectors and mark them by the state they lie in for 5 states- Georgia, South Carolina, Iowa, Oklahoma, Texas. Choose the two singular vectors from the top 5 singular vectors that seem to give a good clustering for the projected counties. Are there states which seem to be close along all of these singular vectors? Why could this be the case?

Now, pretend that results for 100 of these 1008 counties were not available. Your task is to make predictions about winners in these 100 counties based on results in the other counties and available demographic information. We will use matrix completion to solve the problem.

g. (10 points) Do the following-

- Randomly choose 100 of the 1008 counties and set their voting data to 0. We will now complete the matrix to infer the winner in these 100 counties.
- First, set the missing entries equal to the respective column mean based on the remaining 908 entries. Next, normalize and center the columns to have mean 0 and unit L2 norm.
- Use SVD to get a rank-\( k \) approximation of the matrix. Try the following values of \( k \)- 5, 10, 20, 25.
- Once you have the low rank approximation, rescale the columns back by their original L2 norms and recenter them by adding back their original means.
- Find the Democratic and Republican winner in each of the 100 counties based on your imputed voting data and find the accuracy by comparing with the ground truth winner.
- Repeat the experiment 50 times by selecting a new subset of 100 counties each time and find the mean accuracy for Democrats and Republicans. Answer the following questions-
  - Report the mean accuracies for Democrats and Republicans for each value of \( k \).
  - How do the results change as \( k \) changes? Why is this the case?
  - How good are the accuracies for the two parties? Are there very simple approaches that perform better than this low-rank matrix completion approach?

h. (Bonus question) Can you point out any other interesting observation or analysis of the data? For example, some other interpretations of the left or right singular vectors? For this part, your answers do not need to be limited to SVD based explorations.

Deliverables: Plots for part b, e and f. Discussions for part b, c, d, e, f, g.
Part 2: SVD for image compression

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.

a. (4 points) 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. (6 points) Run SVD and recover the rank k approximation, for k ∈ {1, 3, 10, 20, 50, 100, 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. (2 points) Why did we stop at 342?

d. (3 points) 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 question) 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 question) 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?

Deliverables: Discussions for part a, b, c, d, e, f. Code for all parts in the appendix. Plot for part b.