Homework 7

36-462/662, Spring 2020

Due at 10 pm on Thursday, 4 March 2020

Agenda: Practice with principal components analysis of actual data; practice with dealing with data by representing it as feature vectors.

The data file `amazon-dress-jpgs.zip` is an archived directory containing images for the top 205 results Amazon (as of early September 2019) returns for "dresses". The accompanying file `eigendresses` contains code to do the following:

- Convert an image (stored as a three-dimensional array) into a one-dimensional vector.
- Read in all the image files in a directory, re-size them to a common width and height, and convert them into a data-frame.
- Convert a vector into an image.

This code relies on two libraries, `imager` and `plyr`, which you should install.

Advice: Doing PCA on the full data takes a bit less than two minutes on my computer; you will want to cache your results so you're not repeating that every time you knit. (If you have not yet read the handout on using R Markdown for class reports, now is an excellent time to do so.)

1. (10) *Online questions* are online and due at 10 pm on Sunday.
2. (5) Comment the `image.directory.df` function.
3. (4) Use the `image.directory.df` function to create a data frame storing all of the re-sized images. (You will have to change at least one of the default settings.) Give the dimensions of the data frame. Explain why it has that number of columns and that number of rows. (You may need to look carefully at the code.)
4. Using the data frame you created and the `prcomp()` function, do a principal components analysis of the dress images.
    a. (5) Explain why there are exactly 205 principal components.
    b. (5) Explain why the `$x` component has 205 rows *and* 205 columns.
    c. (5) Explain why the `$rot` component has 205 columns but many more rows. Why does it have the number of rows it does?
5. (4) Use R to find the correlation between scores on PC1 and scores on PC2. Why is this what you should have expected?
6. (4) Uswe R to find the inner product between the PC1 vector and the PC2 vector. Why is this what you should have expected?
7. (4) Plot the cumulative variance retained by the first k components, for $k \in 1:205$. How much variance is retained by the first component? By the first five components? How many components are needed to keep 75% of the variance? To keep 95% of the variance?
8. a. (5) Make two images, one from taking the first principal component vector, and the other from taking the *negative* of the PC1 vector. Give the commands you use, as well as the images.
    b. (5) Find the three images with the largest positive score on PC1, and the three images with the largest negative score. Include the images, the values of the scores, and the commands you used.

c. (5) Describe the contrast between positive and negative values of PC1.

9. (10) Repeat the previous problem with the PC2 vector.

10. In this problem, we'll try to reconstruct a particular image, `71fz7YInecL._UY879_.jpg` (or `71f` for short) using only a limited number of principal components. (I picked `71f` at random.) In all cases, give the commands you used, as well as an explanation of why those commands are the right one to use.

    a. (4) Create an image from the 1-component approximation, by multiplying `71f`'s score on PC1 by the PC1 vector, and transforming the vector into an image.
    b. (4) Create an image using the first 2 principal components. (You will need to add two vectors, or get R to do something which is implictly adding two vectors.)
    c. (4) Create images using the top 5, 10, 100 and 205 principal components.
    d. (3) Why are the images change less and less as you use more components?
    e. (4) Why is the image you made using all 205 PCs not the same as the original `71f` image? How could you make it identical?

Rubric (10): The text is laid out cleanly, with clear divisions between problems and sub-problems. The writing itself is well-organized, free of grammatical and other mechanical errors, and easy to follow. All plots and tables are generated using code embedded in the document and automatically re-calculated from the data. Plots are carefully labeled, with informative and legible titles, axis labels, and (if called for) sub-titles and legends; they are placed near the text of the corresponding problem. All quantitative and mathematical claims are supported by appropriate derivations, included in the text, or calculations in code. Numerical results are reported to appropriate precision. Code is properly integrated with a tool like R Markdown or knitr, and both the knitted file and the source file are submitted. The code is indented, commented, and uses meaningful names. All code is relevant; there are no dangling or useless commands. All parts of all problems are answered with actual coherent sentences, and raw computer code or output are only shown when explicitly asked for.

Extra credit (5): Write a function which takes the index number of an image, an integer k and another integer q, and returns the indices of the k images with which are closest to the target in terms of their scores on the top q principal components. How could you check that this is working properly, at least for small k and q? What are the 5 images closest to `71f` using only the top 3 principal components?