Homework 10: Eigendresses

36-462/662, Spring 2020

Due at 6 pm on Thursday, 14 April 2022

Agenda: Practice with principal components analysis of actual data.

The data file `amazon-dress-jpgs.zip` is an archived directory containing images for the top 205 results which Amazon's clothing store returned for "dress" (as of early September 2019). The accompanying code file `eigendresses.R` 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 a good time to do so.)

1. (10) *Online questions* are online and due at 6 pm on Monday.
2. (3) 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 in `eigendresses.R`.)
3. Using the data frame you created in Q2 and the `prcomp()` function, do a principal components analysis of the dress images.
   a. (1) Why does it make sense to *not* scale the coordinates to have variance 1 before doing PCA here?
   b. (5) The number of principal components you find will be much smaller than the dimension of the data vectors. How many principal components are there? Why are there *exactly* that many principal components?
   c. (5) Explain why the `$x` element of `prcomp()`'s return value is a square matrix; why does it have the number of rows and columns it does. Would it necessarily be a square matrix on a different data set?
   d. (5) Explain why the `$rot` element has 205 columns but many more rows. Why does it have the number of rows it does?
4. **Some redundant calculations**
   a. (4) Use R to find the correlation between scores on PC1 and scores on PC2. Why is this what you should have expected?
   b. (4) Use R to find the inner product between the PC1 vector and the PC2 vector. Why is this what you should have expected?

5. **Variance** (4) Plot the cumulative share of variance retained by the first q components, for $q \in 1:205$. What fraction of the variance is retained by the first component? By the first two? By the first five components? How many components are needed to keep 75% of the variance? To keep 95% of the variance?

6. **Interpreting PC1**

   a. (3) Make two images, one from taking the first principal component vector, and the other from taking the *negative* of the PC1 vector. Label them clearly.

   b. (3) Find the three images with the most positive scores on PC1, and the three images with the most negative scores. Include the images and the values of the scores, and the commands you used.

   c. (4) Describe the contrast between positive and negative values of PC1.

7. **Interpreting PC2, and the joint distribution of PC1 and PC2**

   a. (5) Repeat Q6 with the PC2 vector.

   b. (5) Make a scatter-plot where the horizontal axis is score on PC1 and the vertical axis is the score on PC2. Describe the distribution of scores.

8. **Reconstructing an image from its principal component scores** In this problem, we'll try to approximate (or "reconstruct") one particular image, `71fz7YInecL._UY879_.jpg` (or `71f` for short) using only a limited number of principal components? (I picked `71f` at random.) When you're asked to create an image, include the image.

   a. (3) 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. (3) 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. (3) Create images using the top 5, 10, 100 and 205 principal components.

   d. (3) Why do the images change less and less as you use more components?

   e. (3) Why is the image you made using all 205 PCs not the same as the original `71f` image? How could you make it identical?

9. **Using the scores** The file `ratings.csv` has two columns: one giving the file names, and the other whether volunteer A.E. rated the dress as "absolutely not" (0) or "possibly acceptable" (1). As is the case with most user rating data, the majority of the items have no rating (NA).

   a. (2) Using the scores on the first three principal components, fit a logistic regression to predict A.E.'s ratings. Report the coefficients in a table. In words, what kind of dresses is A.E. more likely to approve of, and which ones is she less like to approve of? (You may want to do something like Q6, or Q7a, for PC3.)

   b. (3) Make a plot of the classification accuracy versus threshold, and another plot of false negative rate as a function of the false positive rate, using all the rated dresses. Does the logistic regression show any ability to predict these classifications? *Hint*: HW 6, HW 7.

   c. (1) Repeat Q9b, but using leave-one-out cross-validation. That is, for each rated dress, re-fit the model using only the *other* rated dresses, and calculate the probability for the left-out dress using those coefficients. (This is a more accurate assessment of predictive power, but computationally annoying.)

   d. (3) For each of the dresses *not* rated by A.E., use the model (estimated with all the ratings) to calculate the probability that it would be classified as "possibly acceptable". Make a histogram of these probabilities, and give the filenames and images of the four most probably-approved dresses.

e. (1) Compare the dresses from Q9d to four random dresses from the training set which were approved, and four random training dresses which were not approved. Do the predictions *look* reasonable?

f. (3) Suppose that recommending a dress which A.E. approves of has a net worth \$1 to the retailer, but recommending a dress she does not approve of has a net cost of \$0.01. (Not recommending a dress has no *monetary* cost.) What is the minimum probability of approval before it is worth showing a dress? Which (if any) of the four most probably-approved dresses should be recommended? *Hint*: HW 2, HW 7.

10. **Timing** (1) How long, roughly, did you spend on this problem set?

Presentation 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. 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 plots and tables are generated by code included in the R Markdown file. 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. All parts of all problems are answered with actual coherent sentences, and raw computer code or output are only shown when explicitly asked for. Text from the homework assignment, including this rubric, is included only when relevant, not blindly copied.

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? Does this look reasonable? — In this problem, for full credit, show and *properly comment* your code. *Hint*: `FNN`.