Homework 4

36-462/662, Fall 2019

Due at 10 pm on Wednesday, 25 September 2019

AGENDA: Theory and application of local linear embedding, and more practice with PCA.

1. *Online questions* are omitted this time because the professor messed up releasing the homework.

2. In local linear embedding, we obtain an $n \times n$ matrix $\mathbf{w}$, where w_{ij} is the weight on $\vec{x}_j$ we use to reconstruct $\vec{x}_i$. Each row of $\mathbf{w}$ sums to one. We then try to find coordinates $y_1, y_2, \ldots y_n$ which minimize

$$\Phi(\mathbf{Y}) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} w_{ij} y_j \right)^2 \tag{1}$$

where $\mathbf{Y}$ is the $n \times 1$ matrix of y_i values (this is the $q = 1$ case, for simplicity). The notes show that this is the same as minimizing

$$\Phi(\mathbf{Y}) = \mathbf{Y}^T \mathbf{M} \mathbf{Y} \tag{2}$$

where

$$\mathbf{M} = ((\mathbf{I} - \mathbf{w})^T (\mathbf{I} - \mathbf{w})) \tag{3}$$

   (a) (5) Show that $\mathbf{M}$ is a symmetric matrix.

   (b) (10) Show that $\mathbf{1}$ is an eigenvector of $\mathbf{M}$, and that its eigenvalue is zero. *Hint:* Each row of $\mathbf{w}$ sums to 1.

   (c) (5) Show that $\Phi(\mathbf{Y}) = \Phi(\mathbf{Y} + c\mathbf{1})$, where c is any constant and $\mathbf{1}$ is the $n \times 1$ matrix whose entries are all 1s. *Hint:* use the previous two parts of the problem.

   (d) (5) Show that $\Phi(\mathbf{Y})$ is minimized by $\mathbf{Y} = \mathbf{0}$.

   (e) (10) To avoid the trivial solution of setting all the y_i to zero, we impose the constraint that $n^{-1} \sum_{i=1}^{n} y_i^2 = 1$. We use a Lagrange multiplier to enforce this constraint; write down the Lagrangian for the constrained minimization problem.

   (f) (15) Show that a solution $\mathbf{Y}$ to the constrained minimization problem must be an eigenvector of $\mathbf{M}$.

3. Install (if you haven't already) the packages `ElemStatLearn` and `scatterplot3d` from CRAN. The data set for this problem is `zip.train` in `ElemStatLearn`. This consists of scans of about 7000 hand-written numeric digits from zip codes on envelopes, scanned in as 16×16 grey-scale images. Each row of the data frame represents a different digit; the first column is the actual digit (as verified by a human being), and the other 256 columns are the grey-scale values of the different pixels (centered around zero).[1] The digits are the classes. Some parts of this problem may take excessively long to run if you use all rows of the data set; it's OK to use just the first 500 rows, but if so, indicate that's what you're doing.

   (a) (5) Do a PCA of `zip.train`, being sure to omit the first column. What command do you use? Why should you omit the first column?

   (b) (10) Make plots of the projections of the data on to the first two and three principal components. (For the 3D plot, use the function `scatterplot3d` from that package.) Include the commands you used as well as the plots. On both plots, indicate which points come from which digits, and make sure that this is legible in what you turn in. (E.g., if you use colors, make sure they look distinct on your printout. You might try `pch=as.character(zip.train[,1])`.) Comment on the results.

   (c) (5) Use the code from the notes to do an LLE with $q = 3$. Include the commands you used.

   (d) (10) Make 2D and 3D plots of the data, as before, but with the LLE coordinates.

   (e) (10) Comment on the plots you obtained from PCA and LLE. Which one seems to be doing a better job, in this case, of revealing interpretable structure in the data? Justify your answer by referring to specific features of each plot.

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. Questions which ask for a plot or table are answered with both the figure itself and the command (or commands) use to make the plot. 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

[1] You can visualize them with the function `zip2image`; see the example at the end of `help(zip.train)`. This is not needed for the problem.

sentences, and raw computer code or output are only shown when explicitly asked for.