Homework Assignment #5

36-350, Data Mining

Due at the start of lecture, 23 October 2009

1. The `state.x77` data set is available by default in R; it's a compilation of data about the US states put together from the 1977 *Statistical Abstract of the United States*, with the actual measurements mostly made a few years before.[1]

   The variables are:

Population	in thousands
Income	dollars per capita
Illiteracy	percentage of adults unable to read and write
Life Exp	average years of life expectancy at birth
Murder	number of murders and non-negligent manslaughters per 100,000 people
HS Grad	percentage of adults who were high-school graduates
Frost	mean number of days per year with low temperatures below freezing
Area	in square miles

   `help(state.x77)` has a little more detail. Also built in to R are `state.center`, giving the longitude and latitude of the geographic center of each state (except for Alaska and Hawaii, which are artificially put somewhere off the west coast), `state.name` for the names of the states, and `state.abb` for the names' two-letter abbreviations.

   (a) Create a plot showing the location of each state, with longitude on the horizontal axis, latitude on the vertical axis, and the states' names or abbreviations in the appropriate positions. Include your code.

   (b) Using the `factanal` command from R with the `scores="regression"` option, do a one-factor analysis of `state.x77`. Include the command you used and R's output.

   (c) Describe the factor you obtained in the previous part in terms of the observable features.

[1] The *Statistical Abstract* is "the best book published in America" (P. Krugman), an immensely valuable compilation of data about a huge range of aspects of American life, put out every year by the Census Bureau. It's available for free online at `http://www.census.gov/compendia/statab/`.

(d) Plot the states by location, with the labels of the states being a linearly increasing function of their factor scores. You should control the minimum and maximum size of the labels. (Remember that many of the factor scores will be negative.) Include your code, and comment on the map it produces. *Hint:* The `cex` option to functions like `text` can be a vector.

Alternately, use the `scatterplot3d` command, from the package of that name, to make a three-dimensional plot, with the z axis being the factor score. If you do this, make sure to orient the plot so it is legible, and the states are clearly distinguished.

(e) Part of the output of the `factanal` command is the p-value of the likelihood ratio test for comparing the fitted factor model to the unrestricted multivariate Gaussian. Plot this p-value against q, the number of factors. Include your code.

(f) Is it plausible that there is really only one factor? Explain, and justify your answer in terms of R's output, *not* your general knowledge of US geography.

2. 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).[2] 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) 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) 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, 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) Use the code from lecture to do an LLE with $q = 3$. Include the commands you used.

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

[2] 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.

(e) Run k-means with $k = 10$ on (i) the raw data, (ii) the 3D PCA projections and (iii) the 3D LLE. Calculate the variation-of-information distance of all three clusterings from the true classes (as given by the first column of `zip.train`). Comment.

3. (Extra credit) Download the `diffusionMap` package from CRAN. Prepare a 3D scatterplot of the data, as in problem 2, using `diffuse`. Repeat the clustering from the end of problem 2 with the `diffusionKmeans` function, and calculate the distance of this clustering from the true classes. Comment on these results.