CISC 271, Winter 2020
Assignment #2: Clustering Data by Principal Components
Due by 1:00PM on Tuesday, February 25, 2020

Please read the details and instructions carefully before you begin to work on the problem. The question in this assignment is modestly difficult because it is intended to be a practical introduction to a method evaluating algorithms for data clustering. There must be a single results section and a single discussion section on your report. The results section of the report must contain one table and one figure; more of either or fewer tables will produce deductions from your grade on this assignment.

Statement of Academic Integrity

This assignment is copyrighted by the instructor, so unauthorized dissemination of this assignment may be a violation of copyright law and may constitute a departure from academic integrity.

Sharing of all or part of a solution to this assignment, whether as code or as a report, will be interpreted as a departure from academic integrity. This includes sharing of the assignment after the due date and after completion of this course.

Question 1: Data Clustering by PCA — 12% of Final Grade

The evaluation of a clustering algorithm appears to be nearly as difficult to define as the term "cluster" is. One simple numerical score of a clustering result is the Davies-Bouldin (DB) index [1]. This measures a sum of ratios of the "scatter" of a pair of clusters to the distance between the centroids, or "centers", of the two clusters. The algorithm is sufficiently widely known that we can use a `MATLAB` implementation to find a score – also called an index – of a cluster.

As described in the class notes and the textbook, one implementation of principal components analysis (PCA) is to compute the singular value decomposition (SVD) of a data matrix in which the mean of each column of the matrix is zero. The computation is provided in the notes and the text.

This question will use data that the instructor downloaded from a well regarded public resource. One data set at the University of California at Irvine describes analytical chemistry values for 178 samples of wine that were grown in a region of Italy. There were three "cultivars", or grape types, that are coded numerically in the data file for this assignment. Each row of the instructor's file has comma-separated values (CSV) that first describe the row variable and then provide numerical values for the variable.

There are three problems for this question:

(a) find the pair of values in the data that provide the "best" clustering, which is the lowest DB index

(b) reduce the 13D data to 2D data using PCA and score the reduced clustering

(c) standardize the 2D PCA data and score the standardized clustering

For each problem, you will need to provide two results:

(i) a numerical score in the form of the DB index

(ii) a scatter plot of the clustering

In addition, for Problem (a), you will need to provide numerical values for the indexes of the variables; these will be distinct integers between 1 and 13.

1.1: Loading The Data

The data are in the file `wine.csv` in the ZIP file for this assignment. The data can be loaded using the `csvread` function that is provided in `MATLAB`. You will need to "skip" the first column when reading.

The data will need to be converted for processing. The conversion will be to a data matrix, in this document referred to as `Xmat`, and a cluster label, in this document referred to as `yvec`.

1.2: Plotting Data

Two column vectors, such as `MATLAB` variables `avec` and `bvec`, can be plotted in many ways. For this assignment, you will need to produce three "scatter" plots of vectors that have 178 entries each. A `MATLAB` function that can do this, using the cluster labels in the variable `yvec`, is

```
gscatter(avec,bvec,yvec)
```

Because user interaction is forbidden, you should use the `MATLAB` command `figure` to generate the three figures.

1.3: Finding Good Data Variables

Finding the variables is a straightforward computation. You will need to use nested `for` loops to search for the best pair of variable indexes. These indexes must be stored in a function variable for use when you generate the plot for Problem (a).

1.4: Computing The PCA

The computation must be done using the `MATLAB` function `svd`. Using `pca`, or any other method of data reduction, will result in an automatic grade of zero for the assignment.

The computation is straightforward. To compute a data matrix that has zero-mean columns, you can use either appropriate linear algebra or a "column-wise" invocation of the `mean` function in `MATLAB`. The SVD can be computed by using the `svd` function in `MATLAB` with three output arguments.

From the data matrix that is the input argument to the SVD, and the V matrix that is the result of the SVD, two "score" vectors can easily be computed as described in the course notes.

1.5: Standardizing The Reduced Data

Standardization of the reduced data is straightforward. To compute this, you can either calculate the mean and standard deviation of each of the two "score" vectors, or you can use the `MATLAB` function `zscore`. Note: a future assignment may require inverting the standardization (this is also an excellent test topic) so it may be prudent to understand the standardization process at this time.

1.6: Presenting The Results

The results must be displayed on the `MATLAB` console for grading by the TA's and must be lucidly presented in your report. A format such as that of Table 1 is acceptable.

Table 1: Summary of clustering results using the Davies-Bouldin (DB) index. The left column indicates the test and the right column is the DB index, presented using four digits of numerical precision. The integers in the third column are the indexes into the data variables that provide the numerically best clustering.

Test	DB Index	Variables
Data Columns	3.1416	$[1\ \ 2]$
Raw PCA Scores	2.7183	
Standardized PCA	1.4142	

1.7: Methods

Your methods must include a narrative description of your computation. For example, in finding the best pair of data variables, do not simply say something like "I used nested `for` loops". You should give a reader the logic that underlies your code, not a line-by-line description of its implementation.

1.8: Discussing The Results

Your discussion should compare and contrast your numerical results and any visual patterns that you observe in the scatter plots. You might also comment on the simplicity and/or difficulty of your implementation. For this assignment, you are encouraged to provide a modest amount of creativity – it is a good idea to try to engage the readers of your report.

2: Grading Guide

We will test your code by invoking the function that you uploaded. Your grade will be reduced if: you plot more or fewer than the specified number of figures; your code outputs anything other than the specified values; of you otherwise deviate in your implementation from these specifications.

The TA's have been instructed to use this guide when they mark your assignment. Your grade will be based on the numerical results and on the report. The distribution of points for the assignment grade are:

$11/44$ points: all and only the numerical values that are produced by the code and that are presented in the results

$11/44$ points: quality of the code in the modified "starter" functions, and any other changes in the submission file that was used to generate values and plots for the report

$22/44$ points: quality of the report, especially including the figures and descriptions; clarity may be assessed, in part, by the written discussion of results

Grading Considerations

- The quality of your report will be considered. You need, at minimum, to conform to the "student version" of the report style in the onQ website; you may wish to consider the "grader version" that we will use for assessing your report.
- The quality of your `MATLAB` code will be considered. Your code should be appropriately indented, sufficiently commented, and otherwise be appropriate software.
- The output of your code will be considered.
- Your code can use functions provided by `MATLAB`, but the code that you submit *must* be your original work. You may not use any builtin functions that perform dimensionality reduction or clustering, because one learning outcome of this assignment is to implement such a reduction using linear algebra.
- Code that causes `MATLAB` to produce an error or warning will result in a failing grade.
- You may assume that the file `wine.csv` is in the current directory when a grader tests your code.

What to turn in:

- You will submit your answers electronically as two files. The code will be tested by one or more graders. The PDF report will be read by one or more graders and will be checked, using electronic methods, to ensure that it meets professional standards for originality.
- The code must be in one `MATLAB` file (`a2_xxxxxxxx.m`). This file will contain all of the code needed to verify that the values and tables in the report can be reproduced. The functions must produce the values for your table and for each figure.

- Your functions must take no arguments, return no values, and require no user input or action such as using the "enter" key. Running this function should produce, on the console, every value that is in the report; the function should also produce every plot that is in your report, each plot in a separate figure. The function should produce no other values or figures. The graders will compare your computed values to the values in the report and may deduct marks from the report for differences between any reported value/plot and the corresponding computed value/plot.
- The report must be in a single PDF file (`a2_xxxxxxxx.pdf`). The PDF file must include a description of how you tested your code. You can also include notes, comments on the problems, and assumptions you have made, as appropriate.
- The assignment must be submitted using the Queen's "onQ" software.

Policies:

- You must complete these questions individually.
- Although you are allowed to discuss the questions with other students, you must write your own answers and `MATLAB` code.
- The syllabus standards apply to this assignment.
- Lateness policy applies starting the minute after the submission deadline, at a rate of 20% off the assignment value per calendar day. *Please note: the time in the onQ system is beyond your control, so submitting within an hour of the deadline is inherently a risky process for which you assume full responsibility.*

References

[1] Davies DL, Bouldin DW: A cluster separation measure. *IEEE Trans Pattern Anal Mach Intell* 1(2):224–227, 1979