CISC 271, Winter 2021
Assignment #5: Logistic Regression and Kernel PCA
Due by 10:30AM on Thursday, March 31, 2021

The subject matter for this assignment is in two sets: data for post-secondary educational institutions, and a well known botanical data set.

Coding for this assignment is relatively modest. This assignment requires modification of starter code for two main tasks and two minor tasks. The major tasks are to implement the Perceptron Rule for learning a separating hyperplane from data and to implement a kernel PCA using a Gaussian kernel function. The minor tasks are to compute the accuracies of two hyperplanes and to output the results, which will be figures and numbers.

Please read the details and instructions carefully before you begin to work on the problem. 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 six figures; more or fewer, of either tables or figures, may 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.

Learning Outcomes

On completion, a successful student will be able to:

- Implement the Perceptron Rule for low-dimensional data
- Implement a kernel PCA method for clustering
- Compute the accuracies of separating hyperplanes for classifiers
- Evaluate a data set by comparing two classifications of the data

Preliminary: The Data

This assignment uses data from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the ASA Statistical Graphics Section's 1995 Data Analysis Exposition. The data were processed by replacing the "yes/no" encoding of an institution's public status with numerical codes. The original data, and a full description, were from

`https://rdrr.io/cran/ISLR/man/College.html`

The data are in the file `collegenum.csv` that is in the ZIP file for this assignment. The code will load the data and classify it for you.

This assignment also uses Fisher's Iris data, which are internally available in `MATLAB`. The code will load the data and classify it for you.

Preliminary: The Code

The instructor has provided "starter" code that you will modify. Of note is that the ROC curve is computed using a `MATLAB` function from the Statistics and Machine Learning toolbox.

The "starter" code has:

- A main function that will invoke the functions to answer the two question in this assignment. You should not modify this code. The main function will import the data and dimensionally reduce the college data to 2D for your visualization and analysis.
- A function `a5q1` that you will modify for Question #1. This function will receive the reduced college data as input and will call an internal function that performs the Perceptron Rule of machine learning. You will modify this code to compute accuracies and output your results for Question #1.
- A function `sepbinary` that you will modify for Question #1. This will compute the Perceptron Rule using a "learning rate", which in the notes is a value η and in the code is a variable `eta`. The code will manage the loop and convergence criteria. The code will return an augmented weight vector that represents a separating hyperplane.
- A function `a5q2` that you will modify for Question #2. This function will receive the Iris data as input and call an internal function that computes a Gram matrix for kernel PCA. You will modify this function to: map the 4D Iris data to 2D; cluster the 2D version; and plot the data using the labels and the cluster indexes.
- A function `gramgauss` that you will modify to compute a Gram matrix for a Gaussian kernel function. It will receive as input a data matrix and, optionally, the variance to be used; it will return a Gram matrix.
- A function `logreg` that you will *not* modify. This function computes a logistic regression for a data matrix and a label vector.
- A function `plotline` that you will *not* modify. This function adds a separating hyperplane to a 2D plot.

Question 1: Perceptron Rule For Machine Learning 10% of Final Grade

For this question you will need to modify the functions `a5q1` and `sepbinary`.

The conceptual problem for this question is: how well can a single artificial neuron perform in learning how to recognize whether a US college is private or public? The reference for this is logistic regression, which the code will perform for you. Your coding task is to implement the Perceptron Rule for machine learning, including a learning rate, and to produce plots that address the conceptual problem.

In `sepbinary`, the starter code will: set up the problem; manage the loop; and use convergence criteria to exit the loop. You will need to modify this code to perform two tasks: compute the residual error, which is a vector in the variable `rvec`; and implement the Perceptron Rule to update the current estimate of the augmented vector for the hyperplane, which is a vector in the variable `v_est`. These can be done in very few lines of code by vectorization, or with a few more lines of code by looping.

In a5q1, you will compute the accuracy of your artificial neuron and of logistic regression. Optionally, you may want to find the threshold values that produce the maximum accuracy; these may not be the threshold values that are returned by sepbinary and logreg. You will display the results to the command line and include the results in your table.

Also in a5q1, you will plot: the ROC curves for the two methods; and the dimensionally reduced data plus the separating hyperplane for each method. The data can be plotted by using the MATLAB function gscatter. The hyperplane can be plotted by using the plotline function in the starter code. This function should produce 4 plots: 2 ROC curves and two data+hyperplane plots.

In your discussion, you may want to comment on the meanings of the threshold values that are varied in MATLAB to produce the ROC curves.

Question 2: Kernel PCA and K-Means Clustering 5% of Final Grade

For this question you will need to modify the functions a5q2 and gramgauss.

The conceptual problem for this question is: how well can we classify Fisher's Iris data as the species *I. setosa*? To do this, you will use a kernel PCA method to reduce the data to 2D for k-means clustering. Your coding tasks are to compute a Gram matrix for a Gaussian kernel function and to implement kernel PCA as described in the notes.

In a5q2, you will compute the kernel PCA. The starter code will call gramgauss to find the Gram matrix of the data. Your coding tasks are finding and sorting the spectral decomposition of the Gram matrix, and then using the Gram matrix and its first two eigenvectors to project the data to 2D; these can be accomplished in a few lines of code. The starter code then performs k-means clustering and returns the indexes as 0 and 1.

Also in a5q2, you will use gscatter to plot the Iris data in two separate figures. First, use the labels; encode 0 as "red" and 1 as "blue". Second, use the cluster indexes; encode 0 as "magenta" and 1 as "cyan". These colors will be used by the graders to evaluate your results.

In gramgauss, you will compute the Gram matrix for the data in Xmat. This can be done in a couple of lines of code if you use the MATLAB function pdist2, and in a few lines of code if you use looping.

3: Grading Guide

We will test your code by invoking the functions 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; or 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:

$4/30$ points: all and only the numerical values that are produced by the code and that are presented in the results

$6/30$ 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

$20/30$ points: quality of the report, especially including the figures and descriptions; clarity may be assessed, in part, by the written introduction, verbal defense of choices, and the discussion of results

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, `a5_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 tables and the figure.
- Your function must take no arguments, return the specified 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 any plot that is in your report. 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 (`a5_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.
- You may assume that the file `collegenum.csv` is in the current directory when a grader tests your code.
- The assignment must be submitted using the Queen's "onQ" software.

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 k-fold cross-validation.
- Code that causes `MATLAB` to produce an error or warning will result in a failing grade.
- You may assume that the file `collegenum.csv` is in the current directory when a grader tests your code.

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.*