WPI Worcester Polytechnic Institute

Computer Science

CS539 Machine Learning
Homework 2 - Spring 2017

PROF. CAROLINA RUIZ

Due Date: Thursday, March 2nd, 2017

HW Instructions

- Carefully study Chapters 6 (sections 6.1-6.8), 7 (section 7.7 is optional), and 8 of the textbook, your class notes, and any other related materials posted on the course webpage.
- Solve each of the problems and exercises assigned in this Homework.
- **Sections B, C and D are programming assignments to be completed in Matlab** (or R with professor's permission). **You must write your own code.** *No other programming languages are allowed on this project*. Make sure to consult online documentation for Matlab (or for R). Also, my miscellaneous notes on Matlab (and R) may be useful for this project.
- **You don't need to submit your homework solutions. Instead, an in-class Test will be given the day that the homework is due. This Test will evaluate your mastering of the material covered by the homework.**
- *This is meant to be an individual homework*. That is, you are expected to work on this homework on your own to make sure you know the material and know how to solve the problems, since you'll be tested individually in the Test. *Nevertheless you can discuss your questions about the homework on the Canvas' discussion forums, and consult with the professor and the TA during office hours, and with classmates if you have any trouble solving the homework problems*.

Section A: Exercises from the Textbook (90 points)

- **Chapter 6:** (Pages 157-158)
  i. Study solutions to Exercises 8, 10, 11.
  ii. (5 points each) Solve exercises 1, 5.
- **Chapter 7:** (Pages 180-182)
  i. (5 points each) Solve exercises 2, 5, 6, 7, 10, 11.
- **Chapter 8:** (Pages 208-210)
  i. (5 points each) Solve exercises 1, 2, 3, 4, 5, 6, 7 (no need to write code), 8, 9, 11.

Section B: Dimensionality Reduction (115 points)

Dataset: For this part of the project, you will use the Communities and Crime Data Set available at the UCI Machine Learning Repository. Carefully read the description provided for this dataset and familiarize yourself with the dataset as much as possible.

(5 points) Make the following modifications to the dataset:

i. Remove the "communityname" attribute (string).
ii. Replace each missing attribute value in the dataset (denoted by "?") with the attribute's mean.
iii. Use random sampling to split your dataset into 2 parts: a training set (with 60% of the data instances) and a validation set (with the remaining 40% of the data instances). Let's call this training set TS and this validation set VS.

Use this modified dataset in all the experiments below. *Note:* Remember that feature selection and feature extraction methods should be applied to the input attributes only, not to the output (target) attribute.

I. **Baseline Regression Model:**
   1. ** *Fitting a linear model:*
      (5 points) Create a multivariate linear regression model over the training set TS using the regression functionality provided in Matlab. Report the obtained regression formula and also **report the time taken to construct this regression model** (for this use timing functionality provided in Matlab).
   2. ** *Evaluating the linear model:*
      (5 points) Evaluate the regression model over the validation set VS. Report the Sum of Square Errors (SSE), the Root Mean Square Error (RMSE), the Relative Square Error (RSE), and the Coeffient of Determination (R^2) of each regression model over the validation set.

II. **Feature Selection: Sequential Subset Selection**
   Look for a function (or functions) provided in Matlab for doing feature selection. Try to find a function similar to the sequential subset selection (either forward or backward) described in Section 6.2 of the textbook.
   1. (5 points) Include the name(s) of the function(s) in the report. Briefly explain what the function does.
   2. (5 points) Apply the function to the training data TS. Include in your report the names of the attributes selected by this function.
   3. (10 points) Repeat steps 1 (** Fitting a linear model) and 2 (** Evaluating the linear model) described above, but now using just the selected subset of attributes constructed above. Remember that you need to modify the validation dataset VS so that it includes just the same exact subset of attributes selected from the training set.

III. **Feature Selection: Ranking Attributes**
   Look for a function (or functions) provided in Matlab for ranking attributes following the "Relief" approach.
   1. (5 points) Include the name(s) of the function(s) in the report. Briefly explain what the function does.
   2. (5 points) Apply the function to the training data TS. Include in your report the names of the top 50 attributes selected by this function in order of importance.
   3. (10 points) Repeat steps 1 (** Fitting a linear model) and 2 (** Evaluating the linear model) described above, but now using just the selected 50 attributes above. Remember that you need to modify the validation dataset VS so that it includes just the same exact 50 attributes selected from the training set.

IV. **Feature Extraction: Principal Components Analysis**
   Look for a function (or functions) provided in Matlab for performing PCA.
   1. (5 points) Include the name(s) of the function(s) in the report.
   2. (5 points) Apply the function to the training data TS. Describe the results of PCA. How many components were constructed? 128 or less? What is the minimum number of components needed to

capture at least 90% of the data variance? Explain.

3. (10 points) Repeat steps 1 (** Fitting a linear model) and 2 (** Evaluating the linear model) described above, but now using just the principal components needed to explain at least 90% of the data variance. Remember that you need to transform the validation dataset VS using the same exact transformation obtained from the training set.

V. **Feature Extraction: Factor Analysis (FA)**

Look for a function (or functions) provided in Matlab for performing factor analysis.

1. (5 points) Include the name(s) of the function(s) in the report.
2. (5 points) Apply the function to the training data TS. Describe the results you obtained from factor analysis.
3. (10 points) Repeat steps 1 (** Fitting a linear model) and 2 (** Evaluating the linear model) described above, but now using the obtained factors. Remember that you need to transform the validation dataset VS using the same exact transformation obtained from the training set.

VI. **Comparison of Results**

(10 points) Create a table summarizing the results of the dimensionality reduction experiments above. This table should contain a column for each the five methods used (Baseline, Sequential subset selection, Relief, PCA, and FA). Rows in the table should include the following:

- number of attributes used to construct the linear regression model,
- number of attributes appearing in the linear regression model,
- time taken contructing the linear regression model,
- Sum of Square Errors (SSE),
- Root Mean Square Error (RMSE),
- Relative Square Error (RSE), and
- Coeffient of Determination (R^2)

(10 points) Brifly analyze the results described on this table.

Section C: Clustering (70 points + 10 bonus points)

Dataset: For this part of the project, you will use the OptDigit Dataset available at the UCI Machine Learning Repository.

- Carefully read the description provided for this dataset and familiarize yourself with the dataset as much as possible.
- Use the following files:
  - optdigits.names
  - optdigits.tra: training dataset
- Remove the *class* attribute so that it is not used when clustering is performed.

I. **K-Means Clustering:**

1. (20 points) Use the k-means procedure implemented in Matlab to cluster the data in optdigits.tra (removing the *class* attribute first). Use Euclidean distance. Experiment with different initial random seeds. Systematically experiment with different values for k (= number of clusters), say between 2 and 12. Use a table to summarize your results. In this table, include runtime, k, distance metric, and SSE (*sum of squared errors*, also called *reconstruction error* in the textbook) for each experiment. Provide a brief analysis of your results.
2. (10 points) Pick the experiment that you think produced the best result. Justify your choice. Use Matlab's plotting functions to produce one or two visualizations of the resulting clusters of your chosen experiment. (e.g., consider using MultiDimensional Scaling (MSD) and Silhouette).
   (5 bonus points) Find a good way to add the *class* attribute in the visualization to see if some clusters are associated with any particular class value(s) (see for example Fig. 6.5 (p. 126) and Fig. 6.12 (p. 144) of the textbook).

3. (10 points) In this part, you will investigate methods to evaluate how well a clustering relates to values of the *class* attribute. Study the notions of *purity, normalized mutual information (NMI), and Rand index (RI)*. Calculate the purity, the NMI, and the RI of the clusters in the experiment you ran with k=10. For calculating these measures you'll need to use the *class* attribute, but after the clustering has been obtained without using *class*.

II. **EM Clustering:**

1. (20 points) Now cluster the data using Gaussian Mixture Models with the EM algorithm. Experiment with different number k of components (= clusters), with different initializations, with shared and not shared covariance matrices among components, and with diagonal and non-diagonal ("full") covariance matrices. Use a table to summarize your results. In this table, include runtime, k, initialization, type of covariance matrix, if covariance matrix is shared or not, the Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC) for each experiment. Provide a brief analysis of your results.
2. (10 points) Pick the experiment that you think produced the best result. Justify your choice. Use Matlab's plotting functions to produce a visualization of the resulting clusters of your chosen experiment. This visualization should show the shape and orientation of each component.
(5 bonus points) Find a good way to add the *class* attribute in the visualization to see if some clusters are associated with any particular class value(s).

Section D: Nonparametric Methods (90 + 10 bonus points)

Dataset: For this part of the project, you will use the OptDigit Dataset available at the UCI Machine Learning Repository.

- Carefully read the description provided for this dataset and familiarize yourself with the dataset as much as possible.
- Use the following files:
  - optdigits.names
  - optdigits.tra: training dataset
  - optdigits.tes: test dataset

I. **Univariate Density Estimation:**

Randomly generate a set of N=100 data points using a uniform distribution in the range from 0 to 50. Construct the following 6 plots, using in each case the specified density estimation function over the randomly generated dataset.

1. (5 points) Using a naive estimator with bin width h= 1 and a separate plot with h=4 (see Fig. 8.2 p. 189 of the textbook).
2. (5 points) Using a Gaussian kernel estimator with bin width h= 1 and a separate plot with h=4 (see Fig. 8.3 p. 190 of the textbook).
3. (5 points) Using a k-nearest neighbor kernel estimator with k=3 and a separate plot with k=6 (see Fig. 8.4 p. 191 of the textbook).

II. **Nonparametric Classification:**

Use the OpDigit dataset for this part. Use k-nearest classification functions in Matlab to classify the data instances in the test set optdigits.tes using optdigits.tra as the training set. Run knn with k=1, 5, 9, 11, using 3 different distance metrics: Mahalanobis, Euclidean, and cosine.

1. (20 points) Use a table to summarize your results. In this table, include runtime, k, distance metric, and classification accuracy for each experiment. Provide a brief analysis of your results.
2. (5 points) Pick the experiment that you think produced the best result. Justify your choice. Include the confusion matrix for this experiment. See what misclassifications are most common and elaborate on your observations.

III. **Outlier Detection:**

Use the optdigits.tra dataset for this part.

1. (20 points) Calculate the Local Outlier Factor (LOF) of each data instance in optdigits.tra. Describe what code you used to do this calculation.
2. (5 points) Sort the data instances in increasing order according to their LOF. Plot a graph where the horizontal axis consists of the sorted data instances and the vertical axis denotes their LOF values. Is there an "elbow" in the plot that could be a good threshold to discern between non-outliers and outliers? Explain your answer.
3. (5 bonus points) Take the 3 data instances with the highest LOF values. See if you can plot the image (digit) corresponding to each of these data instances, and see if you can tell whether or not they are abnormal/outliers.

IV. **Nonparametric Regression:**

You may find it useful to watch Matlab's nonparametric fitting video.

Use the OpDigit dataset for this part, but instead of using the class attribute as discrete (or nominal), use it as continuous.

1. (25 points) Use locally weighted regression functions in Matlab that implement techniques like loess or lowess (LOcally WEighted Scatter plot Smooth) on the optdigits.tra dataset. Use cross-validation to determine a good value for k (= number of nearest neighbors used). Summarize the results of your experiments on a table.
2. (5 bonus points) Find a good way to visualize the smoothed regression curves constructed.
