ECE/MSIM 607 Machine Learning I

Fall 2022, 4:20pm - 7:00pm T

INSTRUCTOR: Jiang Li, Professor, ECE Department.

Office: 231D KH, Phone: 683-6748, Email: JLi@odu.edu

Office Hours: 9:00-10:30am MW

Project 1

**Due on Oct. 17th. Do not use build-in functions in the package you are using to directly implement a classifier or a regressor. You need to implement those functions by your own. Other built-in functions like cov() inv() or mean() are allowed. Please follow the sample codes (Matlab) or pseudo codes for your reference. You can use any computer language.**

Part 1:

You will design a Bayes classifier, a Naive Bayes classifier and a *k*-nearest neighbor classifier in this part. Attached are two data sets. One was generated (“generated\_train.csv” and “generated\_test.csv”) and another one is for zip code recognition (“zipcode\_train.csv”, “zipcode\_test.csv”), which also contains training and testing sets. For the generated data, it is a 400x3 matrix for training and testing, respectively. Each row in the data sets is one observation. All observations belong to one of the two classes (class 0 and class 1). The first two columns are 2 dimensional observations and the third column contains class IDs. For the zip code datasets, there are 16 inputs (features) and the last column are the class IDs, which are from ‘1’ to ‘10’.

**Task 1:** For the generated datasets, scatter plot the training and testing data sets (generated\_train.csv and generated\_test.csv). Show class 0 in red and class 1 in blue.

**Task 2:** For both the generated data and the zip code data (zipcode\_train.csv and zipcode\_test.csv), design a Bayes classifier assuming that the data follows a Gaussian distribution. Estimate corresponding parameters from the training data (parametric estimation). Apply your Bayes classifier to the training and testing data sets, respectively. Report training and testing classification accuracies. (For the zip code dataset, if the inverse of any covariance matrix does not exist (this can be done by checking if the determinant of the covariance matrix is zero), then add a small positive value to all the diagonal components and report the value you added in your project report). (Each row is an data vector and the end column is the class ID for the data vector.)

**Task 3:** Repeat Task 2 by designing a naive Bayes classifier. (A simple way to design a naïve Bayes classifier is to make those off-diagonal elements in the estimated covariance matrix zero.)

**Task 4**: Repeat Task 2. Utilize a nonparametric estimation technique to estimate the conditional distribution *p*(**x**|*C*i), using a Gaussian kernel. Try different kernel sizes *h* and report the best testing accuracies and the corresponding *h* values.

**Task 5**: For both the generated data and the zip code data, design a *k*-nearest neighbor classifier. Try different *k* values (Keep in mind that *k* must be odd) and report the best testing accuracies and the corresponding *k* values. Turn in all of your codes and results.

**Task 6**: For the generated data set, design a method to display the decision boundaries produced by each of the above classifiers. Show them separately.

Part 2:

You will design a linear model for regression and classification in this part by using whatever computer languages you prefer. However, you are not allowed to use existing APIs from Pytorch, Keras, Matlab or other libraries.

**Task 1**: Design your own regression program (pseudo inverse). Train a linear model using the training data set, “regression\_train.csv”. This data set contains 8 inputs and 7 outputs (One row contains 8 inputs followed by 7 outputs). Report the training error. Test your trained model on the testing data set, “regression\_test.csv”. This data set contains 8 inputs and 7 outputs as well. Report the testing error.

**Task 2**: Modify your linear regression model such that it can do linear classification. Run the modified codes on the generated data sets you have in part 1, which has two classes. Report the training classification accuracy and testing classification accuracy and compare with the results you obtained in part for this data set.

**Task 3**: Run the classification codes on the zip code data set you have in part 1. If you have difficulty to inverse the autocorrelation matrix, you can address this issue by utilizing the regularization technique. Repeat this task six times by using six different regularization coefficients of 0.01, 0.1, 0.5, 1, 5, and 10. Report the testing classification accuracies and compare the results with those in part 1.

Write a report and turn in your report and all of your codes and results. There is no specific format for the report but you need to organize your results in a readable manner.