CSC 422/522: HW5

Total: 105 points

Question 1 [57 Points (25 + 32)] - Regression

In this problem we will investigate various methods for fitting a linear model for regression. Download the *regprob.zip* file from the course website.

1. Given a set of n real-valued responses y_i and a set of p predictors, we might try to model y_i as a linear combination of the p predictors. The form of this type of linear model is:

   $$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j * x_{ij}$$

   where y_i is the the value of the response for the i^{th} observation, x_{ij} is the value for the jth predictor for observation i, and β_0 is the intercept. To find 'good' values for all of the βs, one approach is to minimize the *sum of squared errors* (SSE), shown below:

   $$SSE = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j * x_{ij})^2$$

   This approach is known as regression via *ordinary least squares* (OLS). Representing this model in matrix notation, the model can be written in an equivalent form as $Y = X\beta$. Now Y is an $n \times 1$ column vector containing the response variable, X is an $n \times (p+1)$ matrix that contains the p predictors for all n observations as well as a column of all 1s to represent the intercept, and β is a $p+1$ vector. With some matrix calculus it can be shown the value of β that minimizes the SSE is given by:

   $$\hat{\beta}_{OLS} = (X^T X)^{-1} X^T Y$$

   where T indicates a matrix transpose. This formula will give a $(p+1)$ vector containing the estimated *regression coefficients*.

   Complete the following tasks:

   - Load *train.csv*
   - Compute the OLS estimates using the data in *train.csv*. Do not use a package to do this, instead compute it directly from the formula given above. There are 10 predictors in the file, so your solution should contain 11 estimated regression coefficients (1 for each predictor plus 1 for the intercept, 11 numbers in total).
   - Estimate the *mean squared error* on an unseen test set by performing 5-fold cross-validation. Recall the MSE for a set of y observations and $\hat{y}$ predictions is defined as

     $$MSE = \frac{1}{N} \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Note: To make predictions using a linear model, first compute $\hat{\beta}_{OLS}$ using the training data and next compute $X_{test}\hat{\beta}_{OLS}$, where X_{test} is a matrix of test data with $p+1$ columns.

Turn in estimates for all 11 regression coefficients and the estimated MSE obtained from 5-fold CV in addition to any code you wrote to solve this problem.

2. The term 'linear model' indicates that a model is linear with respect to β. However, we can model higher order polynomial terms by explicity computing them, including them in the X matrix, and then fit a linear model to this matrix. Perform the following tasks:

   - Load *polynomial.train.csv*
   - Plot Y as a function of X.
   - Create a new X matrix that includes a column of 1s for an intercept, a column for the original X values, and a column of polynomials for each X^i for $i \in \{2, 3, 4, 5\}$ This will create a matrix with dimensions 300×6.
   - Find the OLS solution to this using $(X^T X)^{-1} X^T Y$.
   - Overlay the fitted values (i.e. $X\hat{\beta}_{OLS}$) as a line on the plot of Y vs. X.

   Turn in the original plot of Y vs. X, the OLS regression coefficients, and the plot showing the fitted values overlayed on the original Y vs. X. Turn in any code you wrote.

Question 2 [30 Points] - Artificial Neural Network

Consider the dataset Image Segmentation Data Set from the UCI repository `http://archive.ics.uci.edu/ml/datasets/Image+Segmentation`. The dataset consists of 19 precomputed attributes of 7 outdoor images, or 7 classes. You are provided with a training set and a test set.

Unlike the problems you are seen in the past, which have all been binary classification, this problem has seven classes. With Artificial Neural Network, there are at least 2 ways to construct a multi-class classifier:

1. Direct approach, where there are 7 output nodes to a neural network
2. One-vs-All classification, where you build one binary classifier per class for each of the 7 classes. When predicting the correct class for a given instance in the test data, we choose the classifier that has the highest confidence.

Your task is to build the best possible 7-output ANN and the best possible One-vs-All classifier. You must submit the following:

- A description of how you built the classifiers including the parameters you chose and the reason behind such a choice. The parameters include epoch, momentum, learning rate, number of hidden nodes and any other parameter you think might help.

- A descriptive comparison in performance between the 7-output ANN and the One-vs-All ANN - compare the 2 models based on their predictive performance on the given test data, training time, and your judgment of which approach is better for this problem.

- Any code you have written (using Matlab, R, Weka's Java API)

Question 3 [10 Points (6 + 4)] - Multi-Class Classification

An electronic nose is a device that can "sniff" gases at various locations. One way to construct the devise is using an array of N semiconductors, each of which will have a different voltage response when in contact with certain gases. Each semiconductor responds to at least one gas (i.e., more than one gas). Let us assume that there are 3 gases A, B and C. Some locations can have either one of the gases or a mixture of gases. Thus, possible class labels are: A, B, C, AB, AC, BC, ABC.

1. If you are allowed to use only an Artificial Neural Network, which of the following configurations are possible? State why or why not.
   - 1 network with 7 output nodes
   - One-vs-All
   - 1 network with only 3 output nodes
2. Irrespective of your answer for the previous part, for each of the above configurations, comment on the complexity of the network. Comment on how you would choose the number of hidden nodes, training time and number of epochs for each of the networks.

Question 4 [8 Points] - Hyperplanes for Classification

Consider N points in a D-dimensional space, some of which are positive and some of which are negative. We all know that for $N = 2$ points in $d = 1$ dimensions, a line can separate positive and negative examples. Based on this, state whether a similar linear separator is possible for each of the following cases. If a linear separator is not possible, give an example and state conditions that must be satisfied for the existence of a linear classifier. The correct answer to this question considers all possible arrangements of the N points in the D-dimensional space. If a linear separator is not possible for even one such arrangement, your answer should state that case as an example for failure and state the conditions when a linear separator is possible.

- N=3, D=2
- N=4, D=2
- N=4, D=3
- N=5, D=3