CSE 347/447 Data Mining: Homework 4

Due on 11:59 PM, October 19, 2017

Submission Instructions

This assignment includes writeup and programming questions. The submission should include: (1) One .pdf file including your answers to Q1 $\sim$ Q4, and the writeup part of Q5.1; (2) Two .py files (DemoFit.py and DemoROC.py) which are your answers to Q5. You are required to put all files in a zipped folder named "HW4.zip".

Q1: GINI, Entropy, and Misclassification Error (4 points)

Name	Age	Salary	Donor?
Nancy	21	37,000	N
Jim	27	41,000	N
Allen	43	61,000	Y
Jane	38	55,000	N
Steve	44	30,000	N
Peter	51	56,000	Y
Sayani	53	70,000	Y
Lata	56	74,000	Y
Mary	59	25,000	N
Victor	61	68,000	Y
Dale	63	51,000	Y

Given the training data set as the table above. (a) Compute the GINI index for the entire data set with respect to the two classes in **Donor** attribute. (b) Let A be the cut value for the **Age** attribute, which divides the data set into two portions, those with age $< A$ and those with age $\geq A$. Let $A = 51$. Compute the GINI index for the two portions. Repeat this step for $A = 53$. Which splitting strategy is better? (c) Repeat the above computation with the entropy criterion. (d) Repeat the above computation with the misclassification error criterion.

Q2: Bayes Classifier (2 points)

Suppose we have d-dimensional numeric training data, in which it was known that the probability density of d-dimensional data instance X in each class i is proportional to $e^{-||X-\mu_i||_1}$, where $||\cdot||_1$ is the L_1 norm (Manhattan distance), and μ_i (d-dimensional vector) is known for each class. How would you implement the Bayes classifier in this case? How would your answer change if μ_i is unknown?

Q3: Cost Matrix (Exercise 5.10.19 of TSK, page 324) (4 points)

Q4: SVM (2 points)

Show that, irrespective of the dimensionality of the data space, a data set consisting of just two data points, one from each class, is sufficient to determine the location of the maximum-margin hyperplane.

Q5: In Vino Veritas (8 points)

We apply the nearest neighbor classifier (sklearn.neighbors.KNeighborsClassifier) to a real wine quality dataset (adapted from the dataset hosted by UCI repository). The data records 11 chemical properties (e.g., the concentrations of sugar, citric acid, alcohol, pH etc.) of thousands of red wines from northern Portugal, as well as the quality of the wines, with "0" being "bad" and "1" being "good". We intend to build a classifier that can predict a wine's quality (target) based on its chemical properties (features). Two data files are included: train.csv and test.csv. Both are of the same format: each line represents one kind of wine, with the first 11 fields as its features and the last one as its quality. We will use train.csv to train the classifier and test.csv to evaluate its performance.

Q 5.1: Underfitting/Overfitting

You are required to complete DemoFit.py. Here you vary the setting of the number of neighbors k in k-NN classifier from 1 to 128. Collect and plot the **average error** (i.e., 1 − accuracy) for each k with respect to training and test data. Run DemoFit.py. What do you observe in the plot? Can you explain the phenomena?

Q 5.2: ROC

You are required to complete DemoROC.py. Here you extract posterior probability for each test instance (refer to Page 18 of slides of Lecture 14). Collect the false positive rate and true positive rate for each distinct probability value and construct the ROC curve.