IS 621: Business Analytics and Data Mining

Assignment 2: Learning versus Design

Instructions:

- Soft deadline for this assignment (recommended due date): **Monday, February 23, 2015 at 11:59 p.m. EST**.
- Hard deadline for this assignment (penalties apply if late): **Monday, March 2, 2015 at 11:59 p.m. EST**.
- Late assignments will receive 50% of credit earned and can be submitted until 4:00 p.m. Thursday, March 5, 2015. Full sample solutions are posted in the course at that time and we will discuss at our meetup, so assignments submitted after 4:00 p.m. will receive no credit.
- Solutions should be typed. Your submission should be electronic and through Blackboard. Multiple files, clearly named, are acceptable.
- Assignments that either cannot be opened correctly or are illegible will receive no credit. If you have any concerns about this, please ask me before the deadline.
- The quality of your solution presentation is as important as the correct answer. For full credit you must show your steps and give clear, thorough answers. In code, this means good commenting.

Background

You are implementing two solutions to a problem in this assignment. The assignment is simple – you are implementing a "vending machine" that must distinguish between different types of coins. For the learning solution, you will build a perceptron. For the design solution, you will build a multivariate normal solution. Details are below.

Data

Data for the assignment are attached in Blackboard as CSV fileshese include both the training data and a pair of test data sets. The public test data set (test-public.csv) has answers. The second test data set (test-private.csv) is the one you will score for a grade and does not have answers.

Restrictions

No outside packages should be necessary in R. Please do not use someone else's perceptron algorithm. That defeats the point of the lesson!

Tasks to Complete:

1. (Learning) A perceptron assigns a class to each observation by using the function

$$h(\boldsymbol{x}) = sign(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x})$$

where x is the vector of inputs (including a 1 in the first entry and then each of the numeric inputs mass, thickness, and diameter in our coin example) and w is the vector of weights that define the hyperplane used to separate the two classes. (In two dimensions, the set of weights define a straight line that separates the two types of object. In three dimensions, the weights define a plane.)

The perceptron algorithm updates the set of weights by considering one of the misclassified examples from the training data set. For instance, after iteration t, you will have a set of weights $w(t)$. The rule for updating the set of weights is to create a new set of weights $w(t+1)$ as follows:

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + y(t)\boldsymbol{x}(t)$$

where $y(t)$ is the correct classification for one of the misclassified observations (chosen however you like) and $\boldsymbol{x}(t)$ is the set of inputs for that same observation.

Thus the algorithm is as follows:

Step 1: Assign an arbitrary set of weights to begin (typically the zero vector).

Step 2: Test the set of weights to see if all observations are correct.

Step 3: If all observations are correctly classified, output the weights and end the algorithm.

Step 4: If all observations are not correctly classified, update the weights using the rule above.

Repeat steps 2-4 until the algorithm is ended in step 3 with all correct classifications.

Program in R a perceptron algorithm that correctly separates the cents from the dimes in the training data set. Then, run your perceptron against the test data sets to make predictions.

For this task, you should submit

- Your R code that implements the perceptron and runs it
- A CSV file with your classifications for the private test data set

2. (Design) For the design solution, we will apply a likelihood method using a multivariate normal distribution for each coin type. This method is based loosely on Bayes's theorem.
    a. The multivariate normal function that we will use is written as follows:

$$f(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{E}) = \frac{1}{\left(\sqrt{2\pi}\right)^{d}\sqrt{|\boldsymbol{E}|}} exp\left\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^{T}\boldsymbol{E}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right\}$$

    We are going to use the following two covariance matrices for this exercise (with the variables ordered mass, diameter, and thickness in the rows/columns of the matrices):

$$E_{cent} = \begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.1452 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix}$$

$$E_{dime} = \begin{bmatrix} 0.0021 & 0 & 0 \\ 0 & 0.1283 & 0 \\ 0 & 0 & 0.0007 \end{bmatrix}$$

    Note that all zeros off the diagonals means we are assuming independence for this data set. Do you think this is a reasonable assumption for this coin project? Explain.

    b. The means for the different variables for each coin are given in the following table:

Coin	Mass	Diameter	Thickness
Cent	2.500	19.05	1.52
Dime	2.268	17.91	1.35

    Implement a likelihood function (a function that calculates $f(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{E})$ as above) for each type of coin in R.

    c. Apply your functions to the public test data and classify each coin according to which likelihood is higher. (Note that the likelihoods are not, strictly speaking, probabilities. That's okay! It's the relative size that matters.) Comment on how well you did in classifying observations with this method.
    d. Apply your functions to the private test data set and classify each observation according to which likelihood is higher.

For this task, you should submit the following items:

- Your answer to part (a)
- Your R code for part (b)
- Your comments for part (c)
- Your classification data set for part (d)