Homework 2

September 17, 2018

Due: September 25, 2018, 11:59 PM EST

Instructions

Your homework submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. If you do work with others, you must list the people you worked with. If you solve any problems by hand just digitize that page and submit it (make sure the problem is labeled).

Your programs must be written in Python 3+. All code must be able to compile and run for full credit. Comment all code following proper coding conventions. Remember, if we can't read it, we can't grade it! (For more information on python coding standards, refer to: *https://www.python.org/dev/peps/pep-0008/*)

You should submit your assignment via Github. Submit your solutions as a PDF named "hw(hw #).pdf". For example, homework 2 should be submitted as hw02.pdf. If the assignment requires coding, submit your working code as a .py file with the same name, i.e., hw02.py. ***For this assignment, you will also need to submit three .txt files as described below.***

If you have any questions address them to:

- Xiaolei Guo (TA) – suninth@ufl.edu
- Connor McCurley (TA) – cmccurley@ufl.edu
- Daniel Wells (TA) – dwells@ufl.edu

Question 1 - 10 points

In this homework, you will be implementing a *probabilistic generative classifier* and a *KNN classifier* and compare their results on the same data.

Three datasets for *training* are provided:

- 2dDataSetforTrain.txt has 2D data from two classes. You can visualize these data points in a scatter plot.
- 7dDataSetforTrain.txt has 7D data from two classes.
- HyperSpectralforTrain.txt is high dimensional data from five classes.

Class labels are given in the last column of all three provided training datasets.

Three unlabeled *testing* datasets are provided:

- 2dDataSetforTest.txt
- 7dDataSetforTest.txt
- HyperSpectralforTest.txt)

Your goal is to discriminate among the classes in each *forTrain file, then provide a classification result for the data in each *forTest file.

Complete the following tasks:

1. In your hw02.py file, submit code that implements and runs the following:

   - Implement the *probabilistic generative classifier*, under the assumption that your likelihood model $p(x|j)$ is mutivariate Gaussian and the prior probabilities $p(j)$ are dictated by the number of samples $n_j \in R$ that you have for each class. This classifier is given by comparing the posterior probability for each class j.
     First, we assume that each class j can have an arbitrary mean $\mu_j \in R^d$ and an arbitrary *full covariance* matrix $R^{d \times d}$. Both of these quantities are to be estimated from the observations in each class

   - Then, implement the *probabilistic generative classifier* under the assumption that your data is distributed according to a multi-variate Gaussian with a *diagonal covariance*

Hint: A diagonal covariance implies the variables in different dimension are independent. That reduces the problem to several univariate MLE problems. The *diagonal covariance* is given as

$$\Sigma_j = \frac{1}{n_j} \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_d \end{bmatrix}$$

$$\sigma_k = \sum_{i=1}^{n_j} \left(x_{ik}^j - \mu_{jk} \right)^2$$

where $k = \{1, 2, \cdots, d\}$ indicates dimension. x_{ik}^j is the i^{th} samples in dimension k from classes j, and μ_{jk} is the estimated mean in dimension k from classes j.

- Implement the *KNN classifier*.

2. Test your classifier implementations on the provided data set several times with different parameter settings and using *cross validation*. Provide a PDF entitled hw02.pdf that discussed the following items:
   - When training the probabilistic generative classifier, how does the *full covariance* compare to *diagonal covariance* in performance for each of the data sets? Why?
   - When training KNN classifier, what happens as you vary k from small to large? Why?
3. Determine which classifier(s) you would use for each data set and give an explanation of your reasoning. *Hint:* This should incorporate some discussion based on results from cross-validation.
4. Submit three .txt files with your predictions for the class label for each test data set named 2DforTestLabels.txt, 7DforTestLabels.txt, and HyperSpectralforTestLabels.txt in each of the three test data sets, respectively. You should use whatever method you implemented that you believe will give the best classification results. You should generate these files by running the numpy.savetxt function on a numpy array with the class labels for each test data point in the order they appear in the *forTest.txt files.