

SENG474 Assignment 2. Fall 2017 (50pts)

Student Name:

Student Number:

Instructor George Tzanetakis

Question	SENG474	CSC578D
1	10	10
2	20	20
3	20	20
4	10	10
Total	50	50

Overview

The goal of this assignment is to explore classification accuracy, ROC curves, maximum-likelihood estimation and the perceptron algorithm. the Naive Bayes classifier. Different marking schemes will be used for undergrads (SENG474) and graduate students. Undergraduate students do not have to answer the grad questions.

All code questions use Python and the scikit-Learn library. You may install it on your own computer or alternatively use the computers in the lab. Submissions in other programming languages or environments are welcome but you will need to find the appropriate libraries yourself or write the code from scratch. Don't hesitate to contact the instructor via email or utilize the mattermost channels for any questions/clarifications you might need.

Submit a PDF for questions 1-2 and code for question 3 through Connex

1 Classifier Accuracy (SEng 474 and CSc 578D: 10 points)

Assume you were given a dataset built from random data, where attributes values have been randomly generated with no consideration to the class labels. The dataset has three classes: rectangle", triangle and circle.

You were asked to build a classifier for this dataset, and told that 50% of the data will be used for training, and 50% for testing.

The testing set is balanced, so you can assume it has the same distribution as the training set. Because you are smart, you will start by establishing a theoretical baseline for your classifiers performance.

- **1.1 (2 points)**

Assume the data is equally split between the three classes (33.3% rectangle, 33.3%, triangle, 33.3%, "circle") and your classifier systematically predicts rectangle for every test instances, what is the expected error rate of your classifier? (Show your work)

- **1.2 (3 points)**

What if instead of always predicting rectangle, the classifier predicted rectangle with a probability of 0.7, and triangle with a probability of 0.3. What is the expected error rate of the classifier in this case? (Show your work)

- **1.3 (2 points)**

Now lets assume that the data is not split equally, but has half ($1/2$) of its data labeled rectangle, one-fourth ($1/4$) labeled as triangle, and one-fourth ($1/4$) labeled as circle . What is the expected error rate of the classifier if, as in question a), the prediction is rectangle for every test instances.

- **1.4 (3 points)** With this dataset (half ($1/2$) labeled rectangle, one-fourth ($1/4$) labeled triangle, and one-fourth ($1/4$) labeled circle) What is the expected error rate of the classifier if, as in question b), it predicted rectangle with a probability of 0.7, and triangle with a probability of 0.3. (Show your work)

2 ROC Curves (SEng 474 and CSc 578D: 20 points)

2.1 Background

You have been introduced to different measures of performance for classifiers: accuracy, precision, recall, F-measure etc. It is important to look at them all, as one measure can hide important information. For example, a 60% accuracy obtained by a classifier dealing with 20 classes (labels) is doing much better than one obtaining a 60% accuracy on 2 classes. ROC graphs are two-dimensional graphs plotting the True Positive rate (TP rate) on the y-axis, against the False Positive rate (FP rate) on the x-axis (See the paper in [1] for definitions). A discrete classifier produces TP and FP rates, represented by a single point in the ROC graph. The point at (0,0) represents a classifier that always predicts negative. This strategy assures that the classifier never generates a false positive. However, this strategy would also never generate a true positive. The upper left corner (0,1) represents a classifier that yields a perfect classification. Finally, the upper right corner (1,1) represents a classifier that would always issue a positive classification, generating true positives and false positives equal to the total number of positive and negative instances in the test set (respectively). Thus, the TP rate and FP rate are both 1.

Classifiers such as Nave Bayes and Neural Networks typically issue an instance probability or score along with each of their predictions. For example, the Nave Bayes classification question on your last assignment chose the class with the highest probability. Imagine that instead of choosing the class with the highest probability, we choose a class when its probability is greater than some threshold t . ROC graphs are often used to visualize the performance of such classifiers at different score thresholds. Given a list of instances, their classes and their scores, we can plot the ROC graph. Scores are plotted one by one in descending order. Each instance is a step moving up if positive, and right if negative. The length of the steps going up depends on the total number of positive instances, and the length of the steps moving right depends on the number of negative instances. For example, if the list has 10 positive instances, it will move up by 0.1 each time. If it has 5 negative instances, it will move right by 0.2. Figure 3 is given as an example of a ROC graph built to analyze the performance of a classifier at different

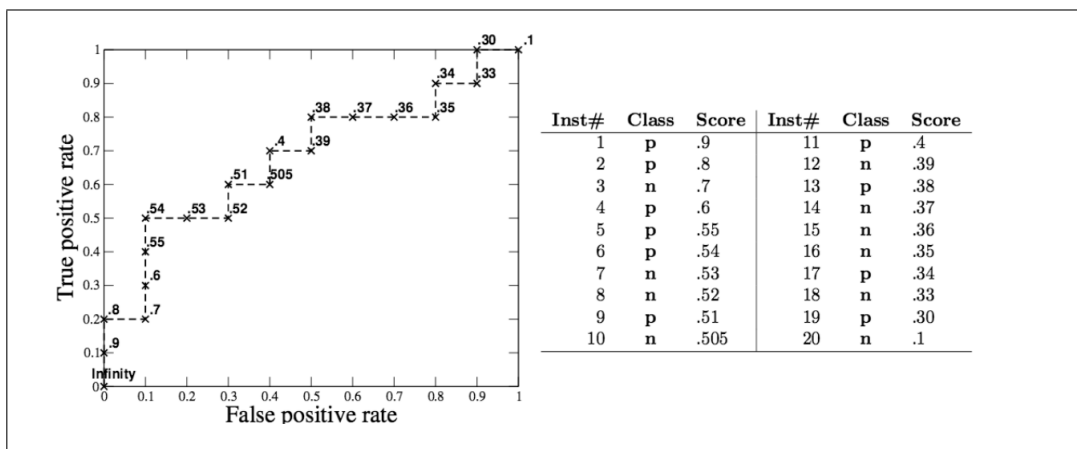


Figure 1: Example for ROC curve

threshold.

Looking at Figure 1, we can see that if we chose to use a classifier with a threshold of 0.8, we get a precision of 100% (2/2), but a recall of 20% (2/10). With a threshold of 0.54, we lost in precision with 83.3% (5/6), but the recall has increased to 50% (5/10).

Instance	True Class	Classifier A	Classifier B
1	P	0.73	0.61
2	P	0.69	0.03
3	N	0.44	0.68
4	P	0.55	0.31
5	N	0.67	0.45
6	N	0.47	0.09
7	P	0.08	0.38
8	N	0.15	0.05
9	N	0.45	0.01
10	P	0.35	0.04

2.2 Problem

You are asked to evaluate the performance of two classifiers, A and B. The following table shows the ranking obtained by applying the classifiers to a test set of 10 instances.

- **2.1 (SENG 474 10/CSC578D 5 points)** Plot the ROC graphs for both A and B on the same graph, as in Fig 1.
- **2.2 (5 points)** For classifier A, suppose you choose the cutoff threshold to be $t = 0.5$. In other words, any test instances whose ranking is greater than t will be classified as a positive example. Compute the precision, recall, and F-measure for the classifier at this threshold value.
- **2.3 (5 points)** Plot the curve of an unbiased random classifier (equal probability of predicting positive or negative) on the graph in a). At what threshold does classifier A performs better than a random classifier? At what threshold does classifier B perform better than a random classifier?
- **2.4 (Only CSC578D 5 points)**

Based on the Tom Fawcetts paper [1]: ROC graphs: Notes and practical considerations for researchers: Consider two discrete classifiers whose performances have been placed on ROC graph. Classifier As coordinate are (0.3, 0.7) and classifier B is positioned at (0.8, 0.1). Which of the two classifiers would you choose and why?

3 The Perceptron algorithm

In this question you will implement a simple 2D perceptron classifier, and visualize how it learns the decision boundary based on some hand-crafted two dimensional data. The training data (Xtrain) is sampled from two linearly separable gaussian distribution with same variance and different means. Training data is normalized to have zero mean and unit variance and it is organized in matrix (Xtrain) form where each row is a training sample. The label of each training sample is determined by the corresponding generating distribution, and it is also organized in matrix form (Ytrain).

To answer this question you are provided with some skeleton code and you are asked to implement the missing functions (do not modify the skeleton code!) and submit your modified python file. In particular, you have to implement the stochastic gradient descent algorithm for the perceptron classifier. This function has to go through the entire dataset (called training epoch) and for each training sample update the parameters according to the perceptron update described in the slides). Please note that the skeleton of this function automatically adds a column of ones to the training data in order to take into account the intercept parameter. After each epoch, we can plot the learned decision boundary by using the provided function.

- **3.1 (SENG474 10 points, CSC578D 5 points)** Set `nepochs=25`, learning rate = 0.001. and plot the first 5 iterations as well as the last decision boundary. Explain what happens.
- **3.2 (10 points)** Set `nepoch=25` and experiment with different learning rates (0.1, 0.01, 0.0001). Explain what happens.
- **3.3 (only CSC578D 5 points)** Using the same training set plot the decision boundary computed by the perceptron algorithm in scikit-learn as well as the decision boundary computed by a linear support vector machine (SVM) also using the algorithm in scikit-learn. Show the resulting boundaries on top of the final decision boundary of your algorithm. Comment on what you observe.

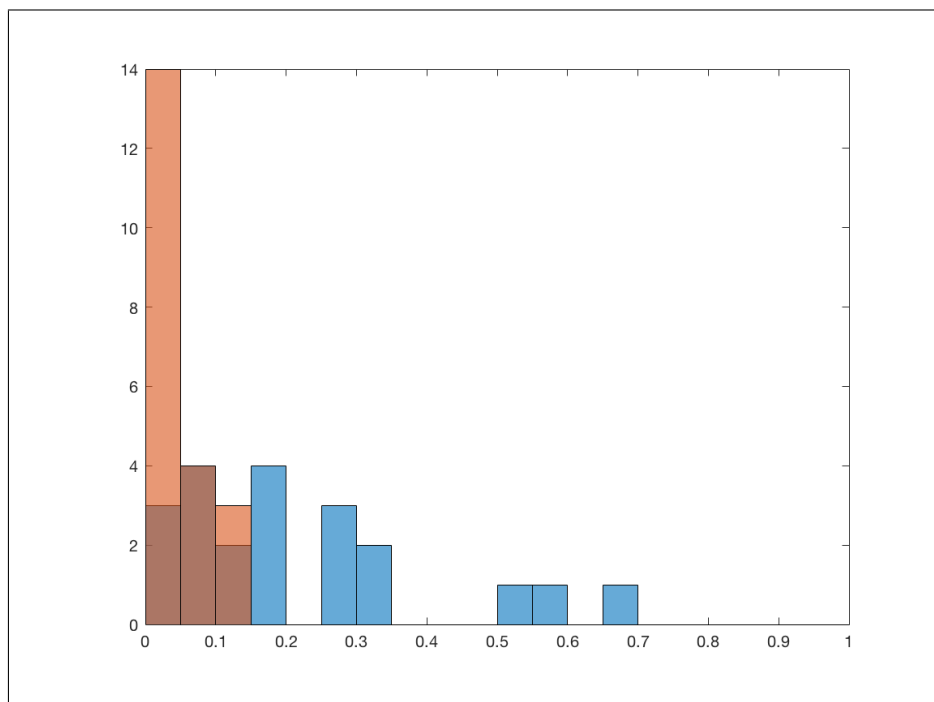


Figure 2: Example for ROC curve

4 Maximum Likelihood Estimation

- **4.1 (5 points)**

Let $\theta = P(X = T)$. Calculate the **MLE** for θ for the following dataset by finding the maximum of $P(D|\theta)$. $D = T, T, T, T, T, T, T, F, F, F$.

- **4.2 (5 points)** Consider the two histograms shown in Figure 2 consisting of sets of data corresponding to two classes. Assume that you can treat these histograms as a crude empirical approximation to an underlying probability density function. Let's call the left histogram A and the right histogram B. Note that the shaded rectangles belong to B (the Blue histogram) and are shown that way as they overlap with the histogram of A. Consider the following data: $D = 0.1, 0.5, 0.6, 0.1, 0.15, 0.55, 0.68, 0.15, 0.12$. Which of the two models (A and B) is more likely to explain the data. Don't just provide the answer but explain. Be precise about how you calculate the likelihood and what is your reasoning.

References

- [1] T. Fawcett, “Roc graphs: Notes and practical considerations for researchers,” *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.