Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 6

Due: November 14, 2023(100 points)

YOUR NAME
YOUR GIT USERNAME
YOUR E-MAIL

Evaluating Your Classifier Performance

In lecture, you will recall that you can calculate the ROC curve by computing the true positive rate (TPR = Probability of Detection, i.e. P_D) and false positive rate (FPR = Probability of False Alarm, i.e P_{FA}) at every threshold.

In this homework, our function will take in predicted classifier scores (`scores` from Livermore Laboratory sensors) and the true labels (`labels` of whether or not laser tubes are calibrated). We will implement a special function that efficiently plots the ROC curve and calculates the AUC *for a specified range* of FPR. Where your function differs from traditional ROC curve calculations is that it will compute only the P_D from the range min-P_{FA} (defaulted to 0) to max-P_D (required parameter). It will then return all the P_{FA} and P_D values that can create a ROC curve in the specified P_{FA} range, as well as the area under it. For example, I can specify max-$P_{FA} = 0.4$, and our function will give us all the ROC scores in the range $P_{FA} = [0, 0.4]$, inclusive of the end values. While there may be a way to leverage existing libraries (e.g., Scikit-Learn), please refrain from doing so (though you can feel free to check your answers with them.) The function signature looks like the following:

```
def roc_auc_pfa(scores, labels, maxpfa, minpfa = 0):
  return pfa_in_range, pd_in_range, auc_in_range
```

We will check your answers against our data, which you can find here at the course website. To read this data, feel free to use this code:

```
import numpy as np
data = np.load("assignment6.npz")

# The data that you will read in
```

```
scores_small = data['scores_small']
scores_large = data['scores_large']
labels_small = data['labels_small']
labels_large = data['labels_large']
```

This unpacks the data, which has some small test data that you can use to try out your algorithm before running the analysis on the larger set of data. If interested, this data has been drafted from National Ignition Facility readouts. Please make your code readable for any data with the above generic signature.

Homework Questions

1. Plot the ROC curve and calculate the AUC for the following ranges:
   a) $P_{FA} \in [0, 1.0]$, the full range of thresholds
   b) $P_{FA} \in [0, 0.4]$
   c) $P_{FA} \in [0, 0.75]$
   d) $P_{FA} \in [0.25, 0.75]$
2. Your implementation notes:
   a) Describe your implementation. How would you sweep your thresholds? For each threshold, how would you calculate the PFA and PD? What is the runtime in big-$\mathcal{O}$ notation?
   b) Determine the runtime of your implementation in big-$\mathcal{O}$.
   c) Can you make your implementation run in $\mathcal{O}(N \log N)$?
3. What thresholds provide a *precision* of 0.9?
4. At this threshold, what is the *accuracy* of the classifier?

Submission Instructions

Submit your work to Gradescope. There, you will need to upload a PDF file with the written answers (including the plots) to all the questions above. As well, there will be a link to upload your Python code. Make sure that the signature matches the above signature as we will check it against other types of data.