EECE5644 Spring 2020 – Take Home Exam 3

Submit: Monday, 2020-March-23 before 11:45ET

Please submit your solutions on Blackboard in a single PDF file that includes all math and numerical results. Only the contents of this PDF will be graded.

For control purposes, please also provide any code you write to solve the questions in one of the following ways:

1. Include all of your code in the PDF as an appendix.
2. In the PDF provide a link to your online code repo where the code resides.

This is a graded assignment and the entirety of your submission must contain only your own work. Do not make your code repo public until after the submission deadline. If people find your code and use it to solve questions, we will consider it cheating for all parties involved. You may benefit from publicly available literature including software (that is not from classmates), as long as these sources are properly acknowledged in your submission.

Copying math or code from each other is clearly not allowed and will be considered as academic dishonesty. Discussing with the instructor and the teaching assistant to get clarification, or to eliminate doubts is acceptable.

By submitting a PDF file in response to this take home assignment you are declaring that the contents of your submission, and the associated code is your own work, except as noted in your acknowledgement citations to resources.

Start early and use the office periods well. The office periods are on gcal and contain video links for remote participation. You may call 1-617-3733021 to reach Deniz if you have difficulty using the video links for the office periods.

It is your responsibility to provide the necessary and sufficient evidence in the form of explanations, math, visualizations, and numerical values to earn scores. Do not provide a minimal set of what you interpret is being precisely requested; you may be providing too little evidence of your understanding. On the other hand, also please do not dump in unnecessarily large amounts of *things*; you may be creating the perception that you do not know what you are doing. Be intentional in what you present and how you present it. You are trying to convince the reader (grader) that your mathematical design, code implementations, and generated results are legitimate and correct/accurate.

Question 1 (50%)

In this exercise, you will train many multilayer perceptrons (MLP) to approximate the class label posterior, using maximum likelihood parameter estimation (equivalently, with minimum average cross-entropy loss).

Pick your own C value (number of classes) from this set: $\{3,4,5\}$. Pick two activation functions for the perceptrons; one sigmoid, one soft-ReLu type differentiable function. Your MLP structure will consist of 2 fully-connected layers of adaptive weights, followed by a softmax layer to ensure your model output conforms to the requirements of a probability distribution over discrete class labels. The dataset you will classify consists of 2-dimensional real vectors. The number of perceptrons in the first layer of the MLP will be determined using cross-validation procedures (e.g., from the set $\{1,...,6\}$).

Using the Matlab function generateMultiringDataset.m generate two sets of data. D_{train} will have 100 iid samples and their class labels. D_{test} will have 10000 iid samples and their class labels. Do NOT use D_{test} in making any training or model selection choices in the process. You will use it only for final performance evaluation.

Using 10-fold cross-validation with D_{train}, and minimum classification error probability as the objective, select the best combination of *number of perceptrons in the first layer* and *the activation function*.

When you determine the best combination of model order and activation nonlinearity, train an MLP with these specifications using D_{train}. This is your final trained MLP model for class-label posteriors. Throughout this process, make sure that you train the MLS from multiple initial conditions and select the best solution encountered, to avoid the possibility of relying on training attempts that may get trapped in a local minimum.

Using MAP classification rule and your final trained MLP model for class label posteriors, classify the samples in D_{test} and estimate the probability of error your classifier would achieve.

Report/explain the entire process you went through including your modeling and algorithm design choices, present your results and discuss any insights/observations you made.

Repeat the entire process with new training datasets that have 500 and 1000 training samples, respectively. Use the same test dataset for final performance evaluation. Discuss the impact of number of training samples on your final modeling choices and test performance.

Note: You may use software packages for MLP weight optimization with a given training set. However, you must implement your own 10-fold cross-validation procedure.

Question 2 (50%)

In this excercise, you will train an alternative approximate MAP classifier for the same datasets used in the previous question, but this time using Gaussian Mixture Models for each class conditional pdf. You will compare results with those obtained by the MLP models.

Pick one of your training datasets (e.g., the one with 100 samples). From the labels estimate the class prior probabilities. Then, for each class, using only the samples with that class label, train a GMM for the class-conditional pdf of that class.

For each class, to select the number of Gaussian components for the associated GMM, use 10-fold cross validation with maximum validation-log-likelihood as the objective.

Once you determine the best model order (number of components), use all the data from the training data for the class label under consideration to fit one final GMM using the entire training dataset.

Use the EM algorithm for all GMM-optimization procedures (and make sure to use multiple random initializations for each attempt and select the best training-log-likelihood solution to help mitigate the impact of local minima that may trap some of the EM training attempts).

Once you have these final trained GMMs for each class label, construct an approximate MAP classifier using these GMMs and estimated class priors, and apply this decision rule to the test dataset to assess the probability of error achieved by this classifier.

Repeat the process for all three training sets and report/explain your entire process of modeling this dataset towards approximating the MAP classifier using GMMs for class conditional models. Discuss your results, including the impact of the number of training samples on your final modeling choices and test performance.

Note: You may use software packages for EM-based GMM optimization with a given training set. However, you must implement your own 10-fold cross-validation procedure.