EECE5644 Spring 2022 – Assignment 4

Please submit in Canvas a single PDF file showing all results, and include code as appendix, separate ZIP file, or as an external repository link in the PDF file. Cite all appropriate sources you benefit from. Directly exchanging code/results between classmates is not acceptable, but you may talk to each other in classes, office periods, benefit from each others' questions and answers for those.

Question 1 (60%)

Train and test Support Vector Machine (SVM) and Multi-layer Perceptron (MLP) classifiers that aim for minimum probability of classification error (i.e. we are using 0-1 loss; all error instances are equally bad). You may use a trusted implementation of training, validation, and testing in your choice of programming language. The SVM should use a Gaussian (sometimes called radial-basis) kernel. The MLP should be a single-hidden layer model with your choice of activation functions for all perceptrons.

Generate 1000 independent and identically distributed (iid) samples for training and 10000 iid samples for testing. All data for class $l \in \{-1, +1\}$ should be generated as follows:

$$\mathbf{x} = r_l \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} + \mathbf{n} \tag{1}$$

where $\theta \sim Uniform[-\pi, \pi]$ and $\mathbf{n} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. Use $r_{-1} = 2, r_{+1} = 4, \sigma = 1$.

Note: The two class sample sets will be highly overlapping two concentric disks, and due to angular symmetry, we anticipate the best classification boundary to be a circle between the two disks. Your SVM and MLP models will try to approximate it. Since the optimal boundary is expected to be a quadratic curve, quadratic polynomial activation functions in the hidden layer of the MLP may be considered as to be an appropriate modeling choice. If you have time (optional, not needed for assignment), experiment with different activation function selections to see the effect of this choice.

Use the training data with 10-fold cross-validation to determine the best hyperparameters (box constraints parameter and Gaussian kernel width for the SVM, number of perceptrons in the hidden layer for the MLP). Once these hyperparameters are set, train your final SVM and MLP classifier using the entire training data set. Apply your trained SVM and MLP classifiers to the test data set and estimate the probability of error from this data set.

Report the following: (1) visual and numerical demonstrations of the K-fold cross-validation process indicating how the hyperparameters for SVM and MLP classifiers are set; (2) visual and numerical demonstrations of the performance of your SVM and MLP classifiers on the test data set. It is your responsibility to figure out how to present your results in a convincing fashion to indicate the quality of training procedure execution, and the test performance estimate.

Hint: For hyperparameter selection, you may show the performance estimates for various choices and indicate where the best result is achieved. For test performance, you may show the data and classification boundary superimposed, along with an estimated probability of error from the samples. Modify and supplement these ideas as you see appropriate.

Question 2 (40%)

In this question, you will use GMM-based clustering to segment a color image. Pick your color image from this dataset: `https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images.html`.

As preprocessing, for each pixel, generate a 5-dimensional feature vector as follows: (1) append row index, column index, red value, green value, blue value for each pixel into a raw feature vector; (2) normalize each feature entry individually to the interval $[0, 1]$, so that all of the feature vectors representing every pixel in an image fit into the 5-dimensional unit-hypercube.

Fit a Gaussian Mixture Model to these normalized feature vectors representing the pixels of the image. To fit the GMM, use maximum likelihood parameter estimation and 10-fold cross-validation (with maximum average validation-log-likelihood as the objective) for model order selection.

Once you have identified the *best* GMM for your feature ectors, assign the most likely component label to each pixel by evaluating component label posterior probabilities for each feature vector according to your GMM. Present the original image and your GMM-based segmentation labels assigned to each pixel side by side for easy visual assessment of your segmentation outcome. If using grayscale values as segment/component labels, please uniformly distribute them between min/max grayscale values to have good contrast in the label image.

Hint: If the image has too many pixels for your available computational power, you may downsample the image to reduce overall computational needs).