EECE5644 2022 Spring – Assignment 2

Submit: Before Tuesday, 2022-March-15, 23:59ET

Please submit your solutions at the assignments page in Canvas in the form of a single PDF file that includes all math, numerical and visual results. Also, for verification of the existence of your own computer implementation, include a link to your online code repository or include the code as an appendix / attachment in a ZIP file along with the PDF. The code is not graded, but helps verify your results are feasible as claimed. Only results and discussion presented in the PDF will be graded, so do not link to an external location where further results may be presented.

This is a graded assignment and the entirety of your submission must contain only your own work. You may benefit from publicly available literature including software (not from classmates), as long as these sources are properly acknowledged in your submission. All discussions and materials shared during office periods are also acceptable resources and these tend to be very useful, so participate in office periods or take a look at their recordings. Cite your sources as appropriate. Discussing verbally with classmates are acceptable, but there can not be any written material exchange.

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 citations to resources and allowed otherwise as described.

Question 1 (40%)

The probability density function (pdf) for a 2-dimensional real-valued random vector $\mathbf{X}$ is as follows: $p(\mathbf{x}) = P(L=0)p(\mathbf{x}|L=0) + P(L=1)p(\mathbf{x}|L=1)$. Here L is the true class label that indicates which class-label-conditioned pdf generates the data.

The class priors are $P(L=0) = 0.65$ and $P(L=1) = 0.35$. The class class-conditional pdfs are $p(\mathbf{x}|L=0) = w_1 g(\mathbf{x}|\mathbf{m}_{01}, \mathbf{C}_{01}) + w_2 g(\mathbf{x}|\mathbf{m}_{02}, \mathbf{C}_{02})$ and $p(\mathbf{x}|L=1) = g(\mathbf{x}|\mathbf{m}_1, \mathbf{C}_1)$, where $g(\mathbf{x}|\mathbf{m}, \mathbf{C})$ is a multivariate Gaussian probability density function with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{C}$. The parameters of the class-conditional Gaussian pdfs are: $w_1 = w_2 = 1/2$, and

$$\mathbf{m}_{01} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad \mathbf{C}_{01} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{m}_{02} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \quad \mathbf{C}_{02} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad \mathbf{m}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

For numerical results requested below, generate the following independent datasets each consisting of iid samples from the specified data distribution, and in each dataset make sure to include the true class label for each sample.

- D_{train}^{20} consists of 20 samples and their labels for training;
- D_{train}^{200} consists of 200 samples and their labels for training;
- D_{train}^{2000} consists of 2000 samples and their labels for training;
- $D_{validate}^{10K}$ consists of 10000 samples and their labels for validation;

Part 1: (10%) Determine the theoretically optimal classifier that achieves minimum probability of error using the knowledge of the true pdf. Specify the classifier mathematically and implement it; then apply it to all samples in $D_{validate}^{10K}$. From the decision results and true labels for this validation set, estimate and plot the ROC curve of this min-P(error) classifier, and on the ROC curve indicate, with a special marker, the location of the min-P(error) classifier. Also report an estimate of the min-P(error) achievable, based on counts of decision-truth label pairs on $D_{validate}^{10K}$. Optional: As supplementary visualization, generate a plot of the decision boundary of this classification rule overlaid on the validation dataset. This establishes an aspirational performance level on this data for the following approximations.

Part 2: (30%) (a) Using the maximum likelihood parameter estimation technique train three separate logistic-linear-function-based approximations of class label posterior functions given a sample. For each approximation use one of the three training datasets D_{train}^{20}, D_{train}^{200}, D_{train}^{2000}. When optimizing the parameters, specify the optimization problem as minimization of the negative-log-likelihood of the training dataset, and use your favorite numerical optimization approach, such as gradient descent or Matlab's fminsearch. Determine how to use these class-label-posterior approximations to classify a sample in order to approximate the minimum-P(error) classification rule; apply these three approximations of the class label posterior function on samples in $D_{validate}^{10K}$, and estimate the probability of error that these three classification rules will attain (using counts of decisions on the validation set). Optional: As supplementary visualization, generate plots of the decision boundaries of these trained classifiers superimposed on their respective training datasets and the validation dataset. (b) Repeat the process described in Part (2a) using a logistic-quadratic-function-based approximation of class label posterior functions given a sample. How does the performance of your classifiers trained in this part compare to each other considering differences in number of training samples and function form? How do they compare to the theoretically optimal classifier from Part 1? Briefly discuss results and insights.

Note 1: With $\mathbf{x}$ representing the input sample vector and $\mathbf{w}$ denoting the model parameter vector, logistic-linear-function refers to $h(\mathbf{x},\mathbf{w}) = 1/(1+e^{-\mathbf{w}^{\mathrm{T}}\mathbf{z}(\mathbf{x})})$, where $\mathbf{z}(\mathbf{x}) = [1,\mathbf{x}^{\mathrm{T}}]^{\mathrm{T}}$; and logistic-quadratic-function refers to $h(\mathbf{x},\mathbf{w}) = 1/(1+e^{-\mathbf{w}^{\mathrm{T}}\mathbf{z}(\mathbf{x})})$, where $\mathbf{z}(\mathbf{x}) = [1,x_1,x_2,x_1^2,x_1x_2,x_2^2]^{\mathrm{T}}$.

Question 2 (40%)

Assume that scalar-real y and two-dimensional real vector $\mathbf{x}$ are related to each other according to $y = c(\mathbf{x},\mathbf{w}) + v$, where $c(.,\mathbf{w})$ is a cubic polynomial in $\mathbf{x}$ with coefficients $\mathbf{w}$ and v is a random Gaussian random scalar with mean zero and σ^2-variance.

Given a dataset $D = (\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_N,y_N)$ with N samples of $(\mathbf{x},y)$ pairs, with the assumption that these samples are independent and identically distributed according to the model, derive two estimators for $\mathbf{w}$ using maximum-likelihood (ML) and maximum-a-posteriori (MAP) parameter estimation approaches as a function of these data samples. For the MAP estimator, assume that $\mathbf{w}$ has a zero-mean Gaussian prior with covariance matrix $\gamma\mathbf{I}$.

Having derived the estimator expressions, implement them in code and apply to the dataset generated by the attached Matlab script. Using the *training dataset*, obtain the ML estimator and the MAP estimator for a variety of γ values ranging from 10^{-4} to 10^4. Evaluate each *trained* model by calculating the average-squared error between the y values in the *validation samples* and model estimates of these using $c(.,\mathbf{w}_{trained})$. How does your MAP-trained model perform on the validation set as γ is varied? How is the MAP estimate related to the ML estimate? Describe your experiments, visualize and quantify your analyses (e.g. average squared error on validation dataset as a function of hyperparameter γ) with data from these experiments.

Note: Point split will be 20% for ML and 20% for MAP estimator results.

Question 3 (20%)

Let Z be drawn from a categorical distribution (takes discrete values) with K possible outcomes/states and parameter θ, represented by $Cat(\Theta)$. Describe the value/state using a 1-of-K scheme for $\mathbf{z} = [z_1,\ldots,z_K]^T$ where $z_k = 1$ if variable is in state k and $z_k = 0$ otherwise. Let the parameter vector for the pdf be $\Theta = [\theta_1,\ldots,\theta_K]^T$, where $P(z_k = 1) = \theta_k$, for $k \in \{1,\ldots,K\}$.

Given $D\{\mathbf{z}_1,\ldots,\mathbf{z}_N\}$ with iid samples $\mathbf{z}_n \sim Cat(\Theta)$ for $n \in \{1,\ldots,N\}$:

- What is the ML estimator for Θ?
- Assuming that the prior $p(\Theta)$ for the parameters is a Dirichlet distribution with hyperparameter α, what is the MAP estimator for Θ?

Hint: The Dirichlet distribution with parameter α is

$$p(\Theta|\alpha) = \frac{1}{B(\alpha)}\prod_{k=1}^{K}\theta_k^{\alpha_k-1} \quad \text{where the normalization constant is} \quad B(\alpha) = \frac{\prod_{k=1}^{K}\Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K}\alpha_k)}$$