EECE5644 2023 Spring – Assignment 2

Submit: Before Tuesday, 2023-March-14, 08:00ET

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 (preferred) or include the code as an appendix in 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 (20%)

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 $P(L=0) = 0.6$ and $P(L=1) = 0.4$. The class class-conditional pdfs are $p(\mathbf{x}|L=0) = w_{01}g(\mathbf{x}|\mathbf{m}_{01}, \mathbf{C}_{01}) + w_{02}g(\mathbf{x}|\mathbf{m}_{02}, \mathbf{C}_{02})$ and $p(\mathbf{x}|L=1) = w_{11}g(\mathbf{x}|\mathbf{m}_{11}, \mathbf{C}_{11}) + w_{12}g(\mathbf{x}|\mathbf{m}_{12}, \mathbf{C}_{12})$, 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: $w_{i1} = w_{i2} = 1/2$ for $i \in \{1,2\}$, and

$$\mathbf{m}_{01} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad \mathbf{m}_{02} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{m}_{11} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \mathbf{m}_{12} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \mathbf{C}_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ for all } \{ij\} \text{ pairs.}$$

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

Part 1: (6%) 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 $D^{10K}_{validate}$. From the decision results and true labels for this validation set, estimate and plot the ROC curve for a corresponding discriminant score for this 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 $D^{10K}_{validate}$. 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: (12%) (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^{20}_{train}, D^{200}_{train}, D^{2000}_{train}. 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 $D^{10K}_{validate}$, 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.

Discussion: (2%) 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 (20%)

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 三次多项式 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^{-m} to 10^{n}. 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 and discussion.

Question 3 (20%)

A vehicle at true position $[x_T, y_T]^{\mathrm{T}}$ in 2-dimensional space is to be localized using distance (range) measurements to K reference (landmark) coordinates $\{[x_1, y_1]^{\mathrm{T}}, \ldots, [x_i, y_i]^{\mathrm{T}}, \ldots, [x_K, y_K]^{\mathrm{T}}\}$. These range measurements are $r_i = d_{Ti} + n_i$ for $i \in \{1, \ldots, K\}$, where $d_{Ti} = \|[x_T, y_T]^{\mathrm{T}} - [x_i, y_i]^{\mathrm{T}}\|$ is the true distance between the vehicle and the i^{th} reference point, and n_i is a zero mean Gaussian distributed measurement noise with known variance σ_i^2. The noise in each measurement is independent from the others.

Assume that we have the following prior knowledge regarding the position of the vehicle:

$$p\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = (2\pi\sigma_x\sigma_y)^{-1} e^{-\frac{1}{2}\begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}^{-1}\begin{bmatrix} x \\ y \end{bmatrix}} \tag{1}$$

where $[x, y]^{\mathrm{T}}$ indicates a candidate position under consideration.

Express the optimization problem that needs to be solved to determine the MAP estimate of the vehicle position. Simplify the objective function so that the exponentials and additive/multiplicative terms that do not impact the determination of the MAP estimate $[x_{MAP}, y_{MAP}]^{\mathrm{T}}$ are removed appropriately from the objective function for computational savings when evaluating the objective.

Implement the following as computer code: Set the true vehicle location to be inside the circle with unit radious centered at the origin. For each $K \in \{1, 2, 3, 4\}$ repeat the following.

Place evenly spaced K landmarks on a circle with unit radius centered at the origin. Set measurement noise standard deviation to 0.3 for all range measurements. Generate K range measurements according to the model specified above (if a range measurement turns out to be negative, reject it and resample; all range measurements need to be nonnegative).

Plot the equilevel contours of the MAP estimation objective for the range of horizontal and vertical coordinates from -2 to 2; superimpose the true location of the vehicle on these equilevel contours (e.g. use a $+$ mark), as well as the landmark locations (e.g. use a o mark for each one).

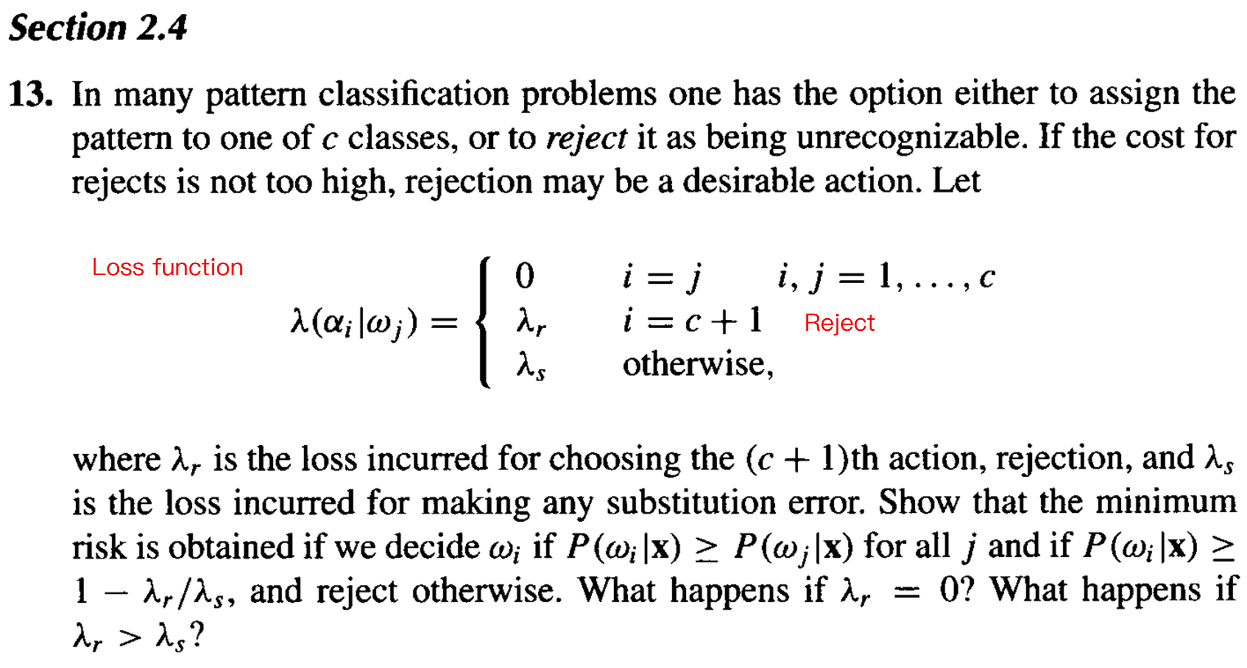
Provide plots of the MAP objective function contours for each value of K. When preparing your final contour plots for different K values, make sure to plot contours at the same function value across each of the different contour plots for easy visual comparison of the MAP objective landscapes. *Suggestion:* For values of σ_x and σ_y, you could use values around 0.25 and perhaps make them equal to each other. Note that your choice of these indicates how confident the prior is about the origin as the location.

Supplement your plots with a brief description of how your code works. Comment on the behavior of the MAP estimate of position (visually assessed from the contour plots; roughly center of the innermost contour) relative to the true position. Does the MAP estimate get closer to the true position as K increases? Doe is get more certain? Explain how your contours justify your conclusions.

Note: The additive Gaussian distributed noise used in this question is likely not appropriate for a proper distance sensor, since it could lead to negative measurements. However, in this question, we will ignore this issue and proceed with this noise model for illustration. In practice, a multiplicative log-normal distributed noise may be more appropriate than an additive normal distributed noise depending on the measurement mechanism.

Question 4 (20%)

Problem 2.13 from Duda-Hart-Stork textbook:



Section 2.4

13. In many pattern classification problems one has the option either to assign the pattern to one of c classes, or to *reject* it as being unrecognizable. If the cost for rejects is not too high, rejection may be a desirable action. Let

Loss function

$$\lambda(\alpha_i|\omega_j) = \begin{cases} 0 & i = j \quad i, j = 1, \ldots, c \\ \lambda_r & i = c + 1 \\ \lambda_s & \text{otherwise,} \end{cases}$$

Reject

where λ_r is the loss incurred for choosing the $(c+1)$th action, rejection, and λ_s is the loss incurred for making any substitution error. Show that the minimum risk is obtained if we decide ω_i if $P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x})$ for all j and if $P(\omega_i|\mathbf{x}) \geq 1 - \lambda_r/\lambda_s$, and reject otherwise. What happens if $\lambda_r = 0$? What happens if $\lambda_r > \lambda_s$?

Question 5 (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

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

通常作为先验分布