EECE5644 Fall 2019 – Exam 1

Submit: Monday, 2019-October-21 before 09:00ET

Please submit your solutions on Blackboard in a PDF file that includes all math and numerical results. Also include your code in one of the following ways: (Acceptable) upload an accompanying ZIP file containing all code files, (Preferred) keep your code in an online version control repository (such as GitHub) and provide a link to the relevant online repository in your PDF file.

Make sure that you cite all resources you benefit from (books, papers, software packages). This is a graded assignment and the entirety of your submission must contain only your own work. You may benefit from literature including software (as allowed by specific restrictions in questions), as long as these sources are properly acknowledged in your submission.

Question 1 (30%)

Design a classifier that achieves minimum probability of error for a three-class problem where the class priors are respectively $P(L=1)=0.15, P(L=2)=0.35, P(L=3)=0.5$ and the class-conditional data distributions are all Gaussians for two-dimensional data vectors:

$\mathscr{N}([\begin{smallmatrix}-1\\0\end{smallmatrix}],[\begin{smallmatrix}1&-0.4\\-0.4&0.5\end{smallmatrix}]), \mathscr{N}([\begin{smallmatrix}1\\0\end{smallmatrix}],[\begin{smallmatrix}0.5&0\\0&0.2\end{smallmatrix}]), \mathscr{N}([\begin{smallmatrix}0\\1\end{smallmatrix}],[\begin{smallmatrix}0.1&0\\0&0.1\end{smallmatrix}])$.

Generate 10000 samples according to this data distribution, keep track of the true class labels for each sample. Apply your optimal classifier designed as described above to this dataset and obtain decision labels for each sample. Report the following:

- actual number of samples that were generated from each class;
- the confusion matrix for your classifier consisting of number of samples decided as class $r \in \{1,2,3\}$ when their true labels were class $c \in \{1,2,3\}$, using r, c as row/column indices;
- the total number of samples misclassified by your classifier;
- an estimate of the probability of error your classifier will achieves, based on these samples;
- a visualization of the data as a 2-dimensional scatter plot, with true labels and decision labels indicated using two separate visulization cues, such as marker shape and marker color;
- a clear but brief description of the results presented as described above.

Note: See the attached generateData_Exam1Question1.m Matlab script for data generation.

Might have to update for “3” equidistant points

Question 2 (35%)

An object 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 object 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 object:

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

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 object 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 object 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 object 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.

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.

Suggestion: For σ_x and σ_y consider values around 0.25, and for the noise variance values σ_i^2 consider values around 0.1 for posterior functions that are illustrative; you may choose different values than what is suggested here, so make sure to specify what your values are in the numerical results presented.

Note: The additive Gaussian distributed noise used in this question is actually not appropriate, since it could lead to negative measurements, which are not legitimate for a proper distance sensor. However, in this question, we will ignore this issue and proceeding with this noise model for the sake of illustration. In practice, a multiplicative log-normal distributed noise may be more appropriate than an additive normal distributed noise.

Question 3 (35%)

We have two dimensional real-valued data (x,y) that is generated by the following procedure, where all polynomial coefficients are real-valued and $v \sim \mathcal{N}(0,\sigma^2)$:

$$y = ax^3 + bx^2 + cx + d + v \tag{2}$$

Let $\mathbf{w} = [a,b,c,d]^{\mathrm{T}}$ be the parameter vector for this polynomial relationship. Given the knowledge of σ and that the relationship between x and y is a cubic polynomial corrupted by additive noise as shown above, iid samples $D = (x_1,y_1),\ldots,(x_N,y_N)$ generated by the procedure using the true value of the parameters (say $\mathbf{w}_{true}$), and a Gaussian prior $\mathbf{w} \sim \mathcal{N}(\mathbf{0},\gamma^2\mathbf{I})$, where $\mathbf{I}$ is the 4×4 identity matrix, determine the MAP estimate for the parameter vector.

Write code to generate $N = 10$ samples according to the model, draw iid $x \sim Uniform[-1,1]$ and choose the true parameters to place the real roots (for simplicity) for the polynomial in the interval $[-1,1]$. Pick a value for σ (that makes the noise level sufficiently large), and keep it constant for the experiments. Repeat the following for different values of γ (note that as γ increases the MAP estimates approach the ML estimate).

Generate samples of x and v, then determine the corrsponding values of y. Given this particular realization of the dataset D, for each value of γ, find the MAP estimate of the parameter vector and calculate the squared L_2 distance between the true parameter vector and this estimate.

For each value of γ perform at least 100 experiments, where the data is independently generated according to the procedure, while keeping the true parameters fixed. Report the minimum, 25%, median, 75%, and maximum values of these squared-error values, $\|\mathbf{w}_{true} - \mathbf{w}_{MAP}\|_2^2$, for the MAP estimator for each value of γ in a single plot. How do these curves behave as this parameter for the prior changes?

Note: Make sure to change gamma to cover a sufficiently broad range to see its effects at multiple scales. To achieve this, you might want to select values for this hyperparameter as power of 10 linearly spaced from $-B$ to $+B$, so that you cover the interval $[10^{-B},10^{B}]$ logarithmically. Choose $B > 0$ well.