Note that "kernel function" and "window function" are used synonymously in this assignment.

1. **Convergence of Kernel Density Estimation (KDE)**

   In lecture we gave 4 conditions to ensure convergence of the KDE estimate (Lecture 23, page 4 (Jenkins' section), and Lecture 23, slide 12 (Chugg's section)).

   In this KDE problem $\underline{x}$ and $\underline{u}$ have dimension *D*. You are given that the unnormalized window function is:

$$\Phi(\underline{u}) = \exp\left(-\frac{1}{2}||\underline{u}||^2\right).$$

   (a) Show that condition (1) is satisfied. (**Tip:** very short)

   (b) Show that a volume given by:

$$V_n = \frac{b}{\sqrt{n}}, \quad b = \text{constant}$$

   satisfies conditions (3) and (4). (**Tip:** very short.)

   (c) First answer: what is the volume V_n of $\Phi\left(\frac{x}{h_n}\right)$?

   Hint: compare $\Phi\left(\frac{x}{h_n}\right)$ to a multivariate normal density function; what is the integral of the normal density function?

   Then answer: give a formula for h_n as some function of *n*, that will give the schedule for V_n stated in (b).

   (d) **Extra credit**. Show that condition (2) is satisfied. For this condition, you may assume:

$$\underline{u} = \begin{pmatrix} a_1 v \\ a_2 v \\ \vdots \\ a_D v \end{pmatrix}$$

   in which a_i are constants, and v is taken to infinity; you can also let and assume:

$$\mathrm{A} = \prod_{d=1}^{D} a_d\,, \quad A \neq 0$$
$$a = \| \underline{a} \|, \quad a \neq 0$$
$$v \neq 0$$

   Hints for showing condition (2):

   (H1) You can drop the ½ without loss of generality

   (H2) For what *u* is $u^2 > u$?

   (H3) For what *u* is $e^{u^2} > e^{u}$?

   (H4) Set up the expression as a ratio, and use L-Hopital's rule

2. **2-class minimum-error classification based on KDE estimates**

In this problem you may use NumPy, sklearn, and matplotlib.

This problem is 2-dimensional (2 features).

Let the class-conditional density functions for a 2-class problem be:

$$p(\underline{x}|S_1) = \alpha_1 p_1(\underline{x}) + \alpha_2 p_2(\underline{x})$$

$$p_1(\underline{x}) = \mathrm{N}\left(\underline{x}\middle|\underline{m}_1, \underline{\underline{\Sigma}}_1\right), \quad p_2(\underline{x}) = \mathrm{U}_{x_1}(0,2)\mathrm{U}_{x_2}(-1,1)$$

$$\underline{m}_1 = \begin{pmatrix}-4\\0\end{pmatrix}, \quad \underline{\underline{\Sigma}}_1 = \begin{pmatrix}4 & 0\\0 & 1\end{pmatrix}$$

$$p(\underline{x}|S_2) = \mathrm{N}\left(\underline{x}\middle|\underline{m}_2, \underline{\underline{\Sigma}}_2\right)$$

$$\underline{m}_2 = \begin{pmatrix}-0.5\\0\end{pmatrix}, \quad \underline{\underline{\Sigma}}_2 = \begin{pmatrix}0.16 & 0\\0 & 9\end{pmatrix}$$

with $\alpha_1 = 0.7$, $\alpha_2 = 0.3$, and the priors are $P(S_1) = P(S_2) = 0.5$.

Note: For all feature-space plots in this problem, use the following ranges for your axes: $x_1 \in [-8,4]$, $x_2 \in [-6,6]$ for consistency.

Tip: for plots of decision boundaries and regions in this problem, you might find it easiest to use the method we have used before – defining a grid of points in 2D feature space, and applying the decision rule to each point on the grid. Suggestion: try an interval of 0.05 between points on the grid (in each dimension); modify if needed for sufficient resolution with reasonable computation time.

(a) *True Bayes minimum error classifier*

Give an expression for the Bayes minimum error decision rule algebraically, in terms of $d_M\left(\underline{x}, \underline{m}_i\right)$, $\underline{\underline{\Sigma}}_i$, $P(S_i)$, $i = 1,2$. Suggestion: use the indicator function for the uniform density. Plot (by computer is probably easier) in 2D feature space, the decision boundary and regions, as well as the means of the 3 densities (2 means in S_1, 1 mean in S_2). (1 final expression and 1 plot)

(b) *Dataset generation.*

Draw and store $N_T = 20{,}000$ data points from $p(\underline{x}, S_k)$, in a $20{,}000 \times 3$ matrix (the 3 columns are x_1, x_2, k). This is the "full training dataset", and your training datasets below will come from this. (No need to report anything.)

Separately, draw and store $N_{Test} = 10{,}000$ data points from $p(\underline{x}, S_k)$, in a $10{,}000 \times 3$ matrix. This will be your testing set. (No need to report anything.)

Tips:

For the normal densities, use `np.random.multivariate_normal.`

For $p(\underline{x}, S_k)$, you can draw each data point by first drawing randomly between S_1 and S_2 according to $P(S_1)$, then draw from $p(\underline{x}|S_1)$ or $p(\underline{x}|S_2)$.

Similarly, to draw from $p(\underline{x}|S_1)$ you can first draw randomly a value of 1 (with probability 0.7) or 2 (with probability 0.3) (biased coin flip). If 1 was drawn, then draw x from $p_1(x)$; if 2 was drawn, then draw x from $p_2(x)$.

To visualize the data, produce a scatter plot in 2D feature space of the first 2000 points in the full training set, with a different symbol or color for each class. (Report 1 plot total.)

(c) *Ideal accuracy.*

Compute the classification accuracy of your classifier in (a) on the testing set.

(d) *Minimum-error classifier based on estimates from the training data.*

In this part your code will learn from the data without knowledge of the probabilities given above.

Repeat this part for a training set $\mathcal{D}_n$ that uses the first n data points in the full training dataset, for $n = 200, 2000, 20000$ (e.g., n=200 will result in 200 data points for $\mathcal{D}_{200}$, some of which will be labeled S_1 and some of which will be labeled S_2).

Use KDE to get estimates $\hat{p}_{\mathrm{n}}(\underline{x}|S_1)$ and $\hat{p}_n(\underline{x}|S_2)$ of the class-conditional densities from $\mathcal{D}_n$. Use a Gaussian window function:

$$\Phi(\underline{u}) = \exp\left(-\frac{1}{2}||\underline{u}||^2\right) \quad \text{with } \underline{u} = \frac{\underline{x}}{h_n}$$

and kernel width (bandwidth) $h_n = \left(\frac{100}{n}\right)^{\frac{1}{4}}$. (No need to report anything.)

(i) Use frequency of occurrence to estimate get estimates $\hat{P}(S_1)$, $\hat{\mathrm{P}}(\mathrm{S}_2)$ of the class priors from $\mathcal{D}_n$. (Report 2 values for each value of n)

(ii) Plot in 2D feature space the decision boundaries and decision regions for a Bayes minimum-error classifier based on your KDE and prior estimates from $\mathcal{D}_n$. (1 plot for each value of n)

(iii) Compute and report the classification accuracy on the testing set for the classifier based on $\mathcal{D}_n$. (1 accuracy for each value of n)

(e) *Comparison.*

Compare results of classifiers (based on probability estimates) in (d), with each other and with the classifier based on the actual probabilities in (a), (c), as follows.

(i) Compare the classification accuracies to each other, and compare the plots to each other. Explain your observations.

(ii) Do the error rates seem consistent with the plots? Explain why or why not.

