EECE 5644 Introduction to Machine Learning and Pattern Recognition
Northeastern University, Summer I 2024

May 14, 2024
Handout #2
Due: May 24

Problem Set 1

Instructions

Please submit your solutions on Canvas in a single PDF file that includes all math, numerical/visual results, and code (as appendix or as a link to an online code repository). Only the contents of this PDF will be graded. Do not link from the pdf to external documents online where results may be presented. Any content outside the PDF will be ignored for grading purposes. External code repository is acceptable, since the purpose is for us to see your personal code. 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. Copying text, math, or code from each other are not allowed and will be considered as academic dishonesty. There cannot be any written material exchange between classmates, but verbal discussions are acceptable. Discussing with the instructor, the teaching assistant, and classmates at open office periods to get clarification or to eliminate doubts are acceptable. 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.

Problem 1.1 (30%)

The probability density function (pdf) for a 4-dimensional real-valued random vector $\boldsymbol{X}$ is as follows: $f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{X}|Y}(\boldsymbol{x} \mid 0)P_Y(0) + f_{\boldsymbol{X}|Y}(\boldsymbol{x} \mid 1)P_Y(1)$. Here Y is the true class label that indicates which class-label-conditioned pdf generates the data.

The class priors are $P_Y(0) = 0.7$ and $P_Y(1) = 0.3$. The class conditional pdfs are $f_{\boldsymbol{X}|Y}(\boldsymbol{x} \mid 0) = g(\boldsymbol{x} \mid \boldsymbol{\mu}_0, \Sigma_0)$ and $f_{\boldsymbol{X}|Y}(\boldsymbol{x} \mid 1) = g(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \Sigma_1)$, where $g(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma)$ is a multivariate Gaussian probability density function with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. The parameters of the class-conditional Gaussian pdfs are:

$$\boldsymbol{\mu}_0 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \Sigma_0 = \begin{bmatrix} 2 & -0.5 & 0.3 & 0 \\ -0.5 & 1 & -0.5 & 0 \\ 0.3 & -0.5 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \boldsymbol{\mu}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Sigma_1 = \begin{bmatrix} 1 & 0.3 & -0.2 & 0 \\ 0.3 & 2 & 0.3 & 0 \\ -0.2 & 0.3 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

For numerical results requested below, generate 10000 samples according to this data distribution, keep track of the true class labels for each sample. Save the data and use the same data set in all cases.

Part A: Expected Risk Minimization (ERM) based classification using the knowledge of true data pdf:

1. Specify the minimum expected risk classification rule in the form of a likelihood-ratio test: $\frac{f_{\boldsymbol{X}|Y}(\boldsymbol{x}|1)}{f_{\boldsymbol{X}|Y}(\boldsymbol{x}|0)} \overset{?}{>} \gamma$, where the threshold γ is a function of class priors and fixed (nonnegative) loss values for each of the four cases $D = i \mid Y = j$ where D is the decision label that is either 0 or 1, like Y.

2. Implement this classifier and apply it on the 10K samples you generated. Vary the threshold γ gradually from 0 to ∞, and for each value of the threshold compute the true positive (detection) probability $P(D = 1 \mid Y = 1; \gamma)$ and the false positive (false alarm) probability $P(D = 1 \mid Y = 0; \gamma)$. Using these paired values, trace/plot an approximation of the ROC curve of the minimum expected risk classifier. Note that at $\gamma = 0$, the ROC curve should be at $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and as γ increases it should traverse towards $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Due to the finite number of samples used to estimate probabilities, your ROC curve approximation should reach this destination value for a finite threshold value. Keep track of $P(D = 0 \mid Y = 1; \gamma)$ and $P(D = 1 \mid Y = 0; \gamma)$ values for each γ value for use in the next section.

3. Determine the threshold value that achieves minimum probability of error, and on the ROC curve, superimpose clearly (using a different color/shape marker) the true positive and false positive values attained by this minimum-P(error) classifier. Calculate and report an estimate of the minimum probability of error that is achievable for this data distribution. Note that $P(error; \gamma) = P(D = 1 \mid Y = 0; \gamma)P_Y(0) + P(D = 0 \mid Y = 1; \gamma)P_Y(1)$. How does your empirically selected γ value that minimizes $P(error)$ compare with the theoretically optimal threshold you compute from priors and loss values?

Part B: ERM classification attempt using incorrect knowledge of data distribution (Naive Bayesian Classifier, which assumes features are independent given each class label)... For this part, assume that you know the true class prior probabilities, but for some reason you think that the class conditional pdfs are both Gaussian with the true means, but (incorrectly) with covariance matrices that are diagonal (with diagonal entries equal to true variances, off-diagonal entries equal to zeros). Analyze the impact of this model mismatch by implementing the ERM classifier using this data distribution model and repeating the same steps in Part A on the same 10K sample data set you generated earlier. Report the same results, answer the same questions. Did this model mismatch negatively impact your ROC curve and minimum achievable probability of error?

Problem 1.2 (30%)

A 3-dimensional random vector $\boldsymbol{X}$ takes values from a mixture of four Gaussians. One of these Gaussians represent the class-conditional pdf for class 1, and another Gaussian represents the class-conditional pdf for class 2. Class 3 data originates from a mixture of the remaining 2 Gaussian components with equal weights. For this setting where labels $Y \in \{1, 2, 3\}$, pick your own class-conditional pdfs $f_{\boldsymbol{X}|Y}(\boldsymbol{x} \mid j)$, $j \in \{1, 2, 3\}$ as described. Try to approximately set the distances between means of pairs of Gaussians to twice the average standard deviation of the Gaussian components, so that there is some significant overlap between class-conditional pdfs. Set class priors to $0.3, 0.3, 0.4$.

Part A: Minimum probability of error classification (0-1 loss, also referred to as Bayes Decision rule or MAP classifier).

1. Generate 10000 samples from this data distribution and keep track of the true labels of each sample.

2. Specify the decision rule that achieves minimum probability of error (i.e., use 0-1 loss), implement this classifier with the true data distribution knowledge, classify the 10K samples and count the samples corresponding to each decision-label pair to empirically estimate the confusion matrix whose entries are $P(D = i \mid Y = j)$ for $i, j \in \{1, 2, 3\}$.

3. Provide a visualization of the data (scatter-plot in 3-dimensional space), and for each sample indicate the true class label with a different marker shape (dot, circle, triangle, square) and whether it was correctly (green) or incorrectly (red) classified with a different marker color as indicated in parentheses.

Part B: Repeat the exercise for the ERM classification rule with the following loss matrices which respectively care 10 times or 100 times more about not making mistakes when $Y = 3$:

$$\Lambda_{10} = \begin{bmatrix} 0 & 1 & 10 \\ 1 & 0 & 10 \\ 1 & 1 & 0 \end{bmatrix} \text{ and } \Lambda_{100} = \begin{bmatrix} 0 & 1 & 100 \\ 1 & 0 & 100 \\ 1 & 1 & 0 \end{bmatrix}$$

Note that, the $(i, j)^{th}$ entry of the loss matrix indicates the loss incurred by deciding on class i when the true label is j. For this part, using the 10K samples, estimate the minimum expected risk that this optimal ERM classification rule will achieve. Present your results with visual and numerical reprentations. Briefly discuss interesting insights, if any.

Problem 1.3 (40%)

Download the following datasets...

- Wine Quality dataset located at `https://archive.ics.uci.edu/ml/datasets/Wine+Quality` consists of 11 features, and class labels from 0 to 10 indicating wine quality scores. There are 4898 samples.
- Human Activity Recognition dataset located at `https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones` consists of 561 features, and 6 activity labels. There are 10299 samples.

Implement minimum-probability-of-error classifiers for these problems, assuming that the class conditional pdf of features for each class you encounter in these examples is a Gaussian. Using all available samples from a class, with sample averages, estimate mean vectors and covariance matrices. Using sample counts, also estimate class priors. In case your sample estimates of covariance matrices are ill-conditioned, consider adding a regularization term to your covariance estimate as in: $\Sigma_{\text{Regularized}} = \Sigma_{\text{SampleAverage}} + \lambda I$ where $\lambda > 0$ is a small regularization parameter that ensures the regularized covariance matrix $\Sigma_{\text{Regularized}}$ has all eigenvalues larger than this parameter. With these estimated (trained) Gaussian class conditional pdfs and class priors, apply the minimum-P(error) classification rule on all (training) samples, count the errors, and report the error probability estimate you obtain for each problem. Also report the confusion matrices for both datasets, for this classification rule. Visualize the datasets in various 2 or 3 dimensional projections (either subsets of features, or if you know how to do it, using principal component analyses). Discuss if Gaussian class conditional models are appropriate for these datasets and how your model choice might have influenced the confusion matrix and probability of error values you obtained in the experiments conducted above. Make sure you explain in rigorous detail what your modeling assumptions are, how you estimated/selected necessary parameters for your model and classification rule, and describe your analyses in mathematical terms supplemented by numerical and visual results in a way that conveys your understanding of what you have accomplished and demonstrated.

Hint: Later in the course, we will talk about how to select regularization/hyper-parameters. For now, you may consider using a value on the order of arithmetic average of sample covariance matrix estimate non-zero eigenvalues $\lambda = \alpha \operatorname{trace} \Sigma_{SampleAverage} / \operatorname{rank}(\Sigma_{SampleAverage})$ or geometric average of sample covariance matrix estimate non-zero eigenvalues, where $0 < \alpha < 1$ is a small real number. This makes your regularization term proportional to the eigenvalues observed in the sample covariance estimate.