Introduction to Machine Learning

ECE 580
Spring 2022

HW #3

Submission Instructions

Submit your work to the corresponding assignment in Gradescope. Although Gradescope accepts multiple file formats, they strongly recommend submitting your assignment as a single PDF file.

You are responsible for ensuring the uploaded file is: 1) the correct file, 2) complete (includes all pages), and 3) legible.

You are responsible for tagging the pages that correspond to each question. Pages may be tagged after submission, even if the submission deadline has passed. If you are submitting close to the submission deadline, submit your assignment first then immediately return to tag the pages.

When code is requested, submit a PDF print-out of your code. Submitting a URL for a cloud-based repository is insufficient.

Scoring Information

Individual questions will be scored holistically on the 9-point scale described in the syllabus. The homework assignment score will be the weighted average of the individual scores for each question. (The weight for each question is shown in parentheses to the left of the question number.)

Late Submissions

Late submissions will be accepted up to 5 days after the submission deadline, with the following point penalty applied if its late submission is not excused: [1]

1 day (0^+ to 24 hours) late: 0.2 point deduction
2 days (24^+ to 48 hours) late: 0.5 point deduction
3 days late: 0.8 point deduction
4 days late: 1.6 point deduction
5 days late: 3.2 point deduction
6 or more days late: score = 3.0 (not accepted for credit)

The late policy is designed to be minimally punitive for submissions up to 3 days late, yet encourage staying current with the coursework for our course by not allowing one assignment's late submission to overlap with the next assignment's submission.

A homework score will not drop below 3.0 as a result of applying the late penalty point deduction.

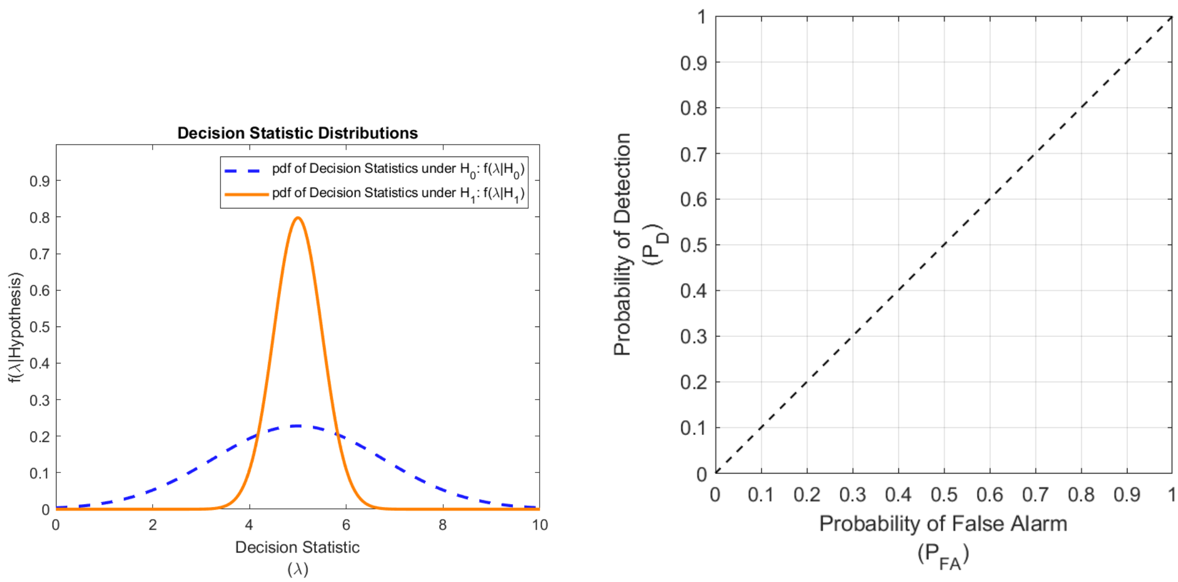
[1] One day = one 24-hour period or fraction thereof.

Decision Statistics $\leftrightarrow$ ROCs

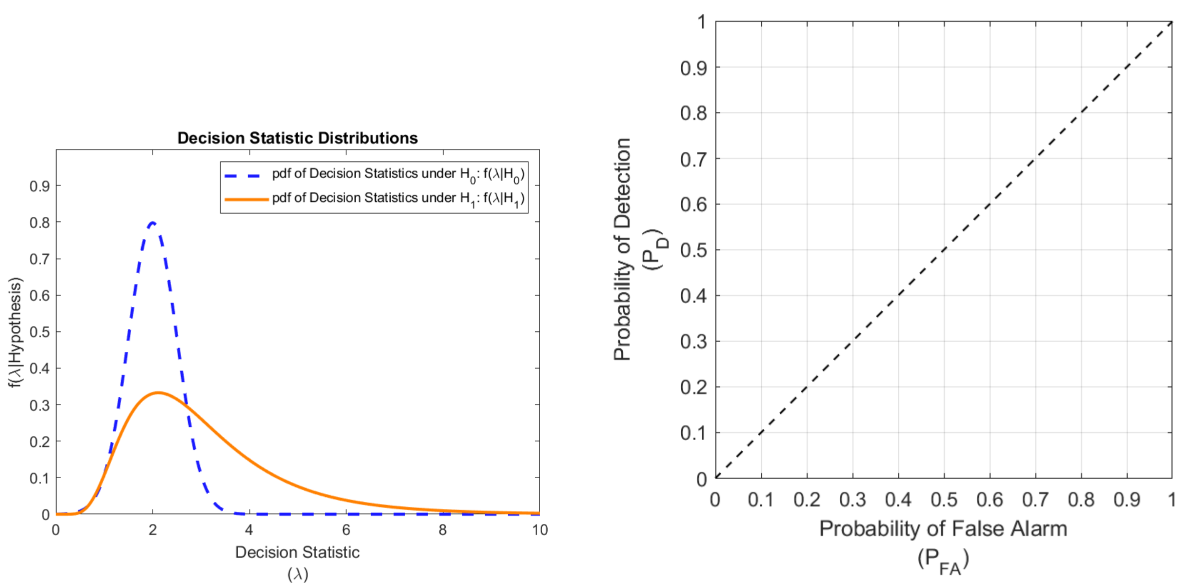
Having intuition for what the shape of the ROC reveals about the distributions of the underlying decision statistics can be quite helpful, as insight regarding the distributions of decision statistics can inform further algorithmic improvements.

(5) 1. For each set of decision statistic distributions given below, sketch by hand (qualitatively, but as accurately as you can) the corresponding ROC. Explain how you determined the shape and slope of the ROC as a function of λ, as well as how you determined key points on the ROC, such as $P_D = 0.5$ or $P_{FA} = 0.5$. We are not concerned about the precision of the ROC, but rather its general shape and relative position within the ROC axes.

(a)

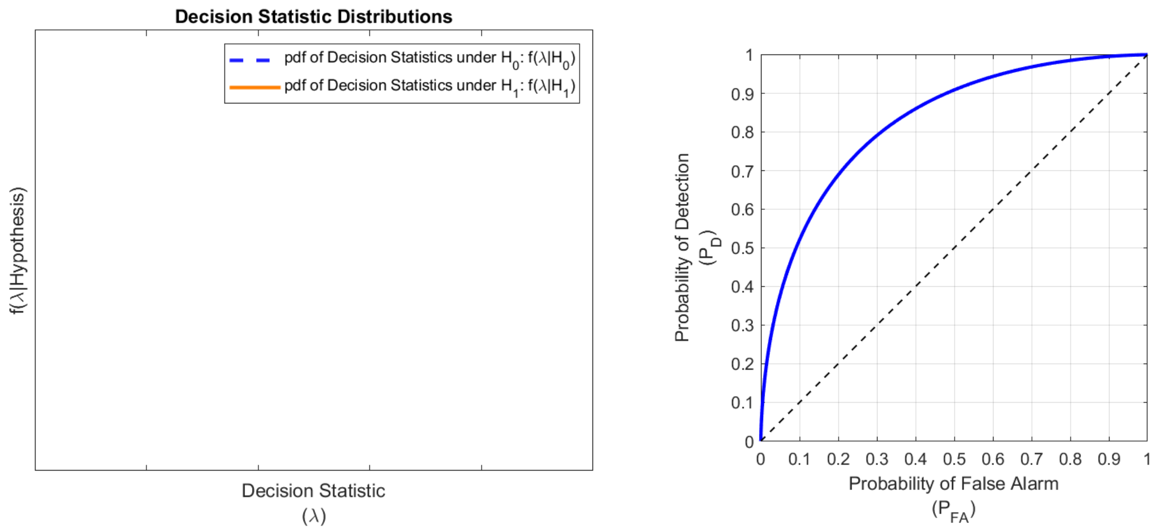


(b)

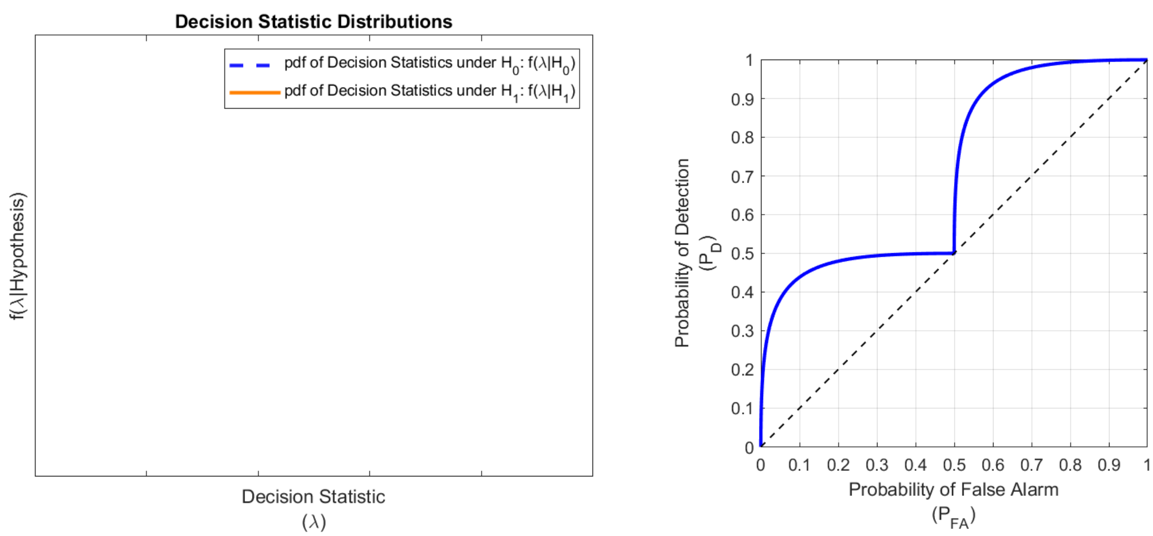


(10) 2. For each ROC given below, sketch by hand (qualitatively, but as accurately as you can) a set of decision statistic distributions that could have plausibly produced the given ROC. Explain how you determined the shapes of the decision statistic distributions as a function of λ. We are not concerned about the precision of the pdfs, but rather their general shapes and locations relative to each other.

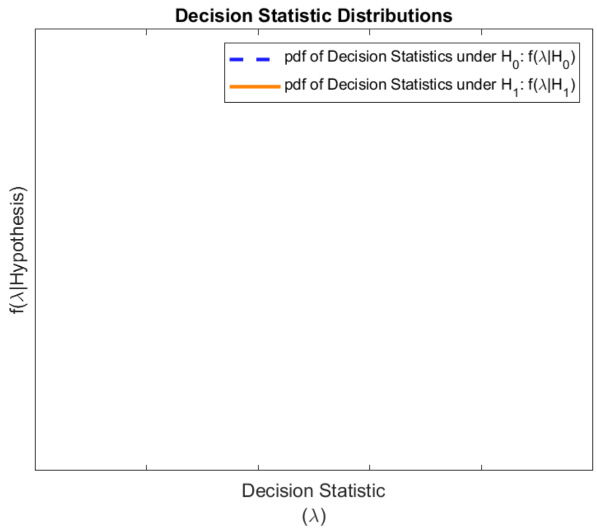
(a)

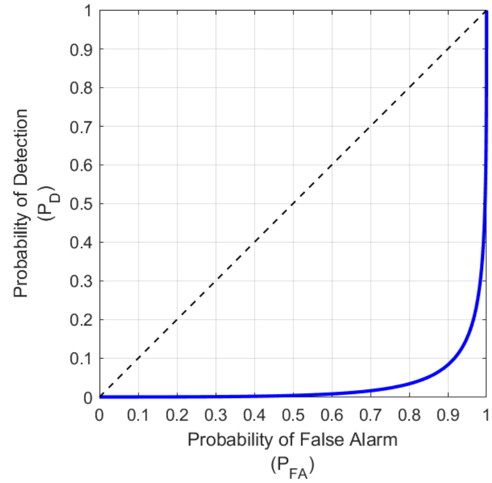


(b)

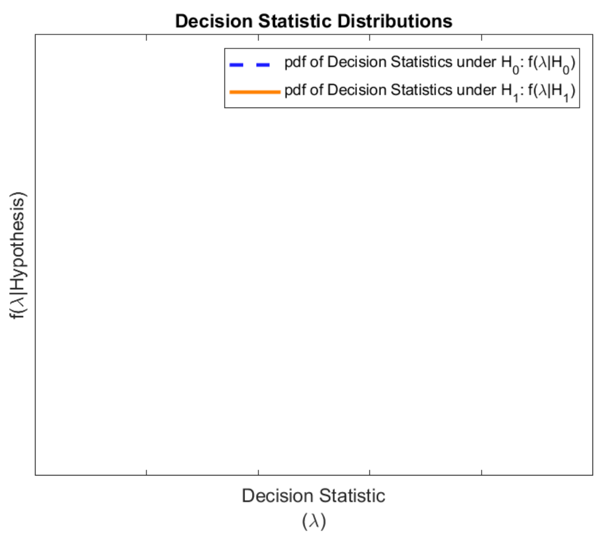


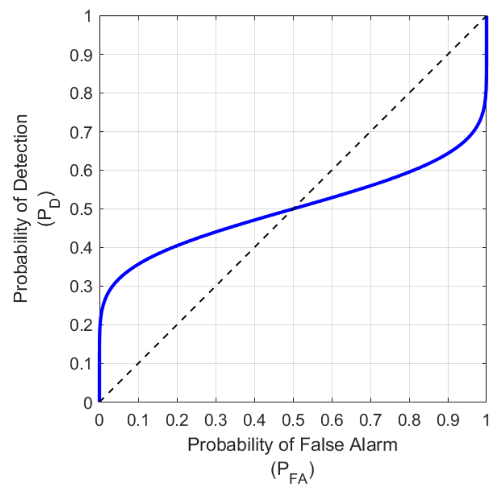
(c)


Decision Statistic Distributions
pdf of Decision Statistics under H_0: $f(\lambda|H_0)$
pdf of Decision Statistics under H_1: $f(\lambda|H_1)$
$f(\lambda|\text{Hypothesis})$
Decision Statistic
(λ)


Probability of Detection
(P_D)
Probability of False Alarm
(P_{FA})
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

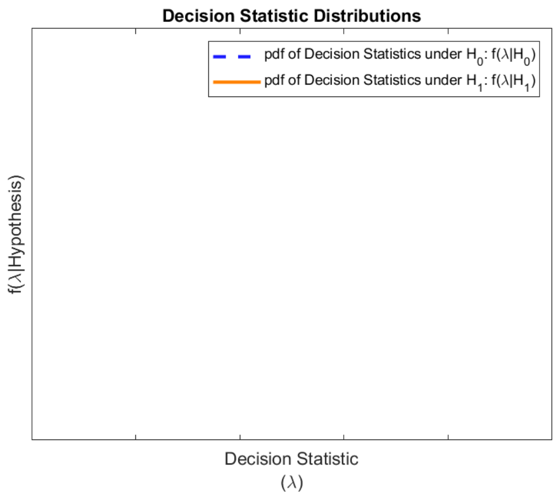
(d)

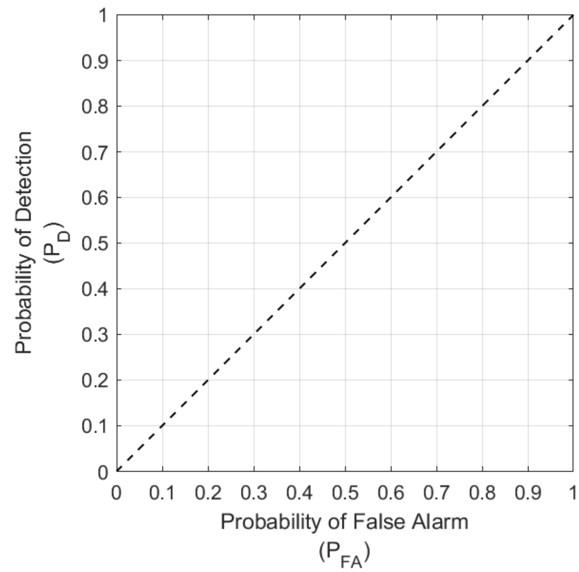

Decision Statistic Distributions
pdf of Decision Statistics under H_0: $f(\lambda|H_0)$
pdf of Decision Statistics under H_1: $f(\lambda|H_1)$
$f(\lambda|\text{Hypothesis})$
Decision Statistic
(λ)


Probability of Detection
(P_D)
Probability of False Alarm
(P_{FA})
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(10) 3. (a) What mathematical transformation could we apply to the decision statistics for question 2c so that the ROC lies entirely above the chance diagonal?

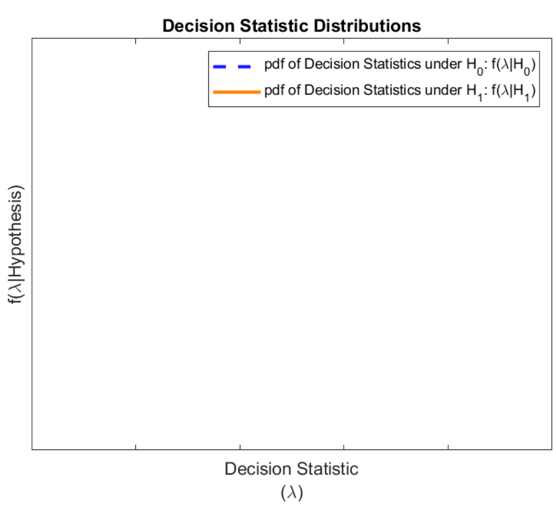
(b) Sketch by hand (qualitatively, but as accurately as you can) the new set of decision statistic distributions after applying the mathematical transformation, and the new ROC. We are not concerned about the precision of the sketches, but rather their general shapes and locations.

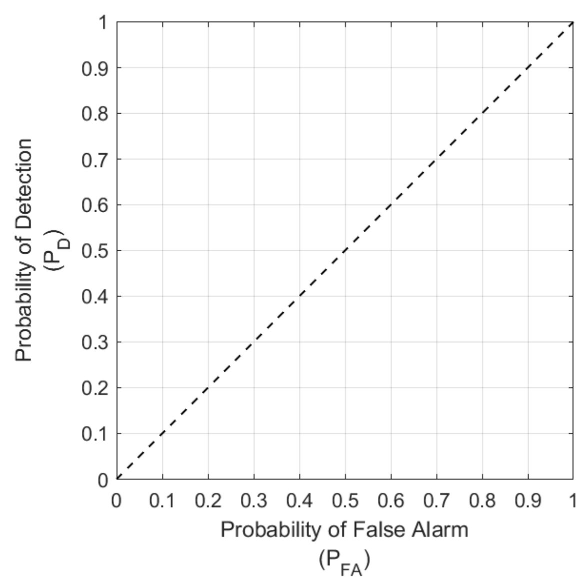




(c) What mathematical transformation could we apply to the decision statistics for question 2d so that the ROC lies entirely above the chance diagonal?

(d) Sketch by hand (qualitatively, but as accurately as you can) the new set of decision statistic distributions after applying the mathematical transformation, and the new ROC. We are not concerned about the precision of the sketches, but rather their general shapes and locations.





Generating ROCs

It is tremendously beneficial to have the ability to specify how you want an ROC to be generated, as there is no computational approach to generating ROCs that is universally better than all others under all conditions.[2]

Make sure you are able to generate an ROC by specifying the specific thresholds you want to apply and that you have flexibility to specify how those thresholds are selected – linearly spaced from $\min(\lambda)$ to $\max(\lambda)$, logarithmically spaced from $\min(\lambda)$ to $\max(\lambda)$, every n^{th} λ in the list of sorted decision statistics, every n^{th} H_0 λ in the list of sorted H_0 decision statistics, thresholds necessary to achieve a set of desired P_{FA} values, thresholds necessary to achieve a set of desired P_{D} values, etc. Even better is code that is extensible, so you can incorporate additional functionality, such as new approaches to specifying how the thresholds are selected or the ability to return other performance measures, as you encounter new use cases. You may choose to write your own function from scratch, or you may choose to leverage ROC generating functions available through standard Matlab[3] or Python packages, in which case you likely will find it helpful to write your own wrapper for these functions. If you choose to leverage existing functions, you may find it helpful to write a "wrapper" function that selects the thresholds from the list of decision statistics according to the ROC generation method you desire.

Regardless of whether you choose to write your own function or leverage functions that may be available through standard Matlab or Python packages, it is critical that you understand how the function(s) you are using work so you can effectively apply those functions to suit your needs and correctly interpret the results they provide.

The following questions concern 4 sets of decision statistics that are provided as `csv` files. The csv files are organized such that each row contains the true class (either 0 or 1) followed by the associated decision statistic. For each set of decision statistics, generate the ROC using:

1. every decision statistic as a threshold (β is $[-\infty$ {sorted list of λ's} $+\infty])$)[4]
2. thresholds selected so they linearly sample the range of decision statistics (β is 99 linearly spaced samples from $\min(\lambda)$ to $\max(\lambda)$, plus $-\infty$ and $+\infty$)
3. thresholds selected so they sample every n^{th} decision statistic, where n is chosen so there will be 99 decision statistics selected as thresholds (or $n = 1$ if there too few decision statistics to down select such that 99 decision statistics are retained), plus $-\infty$ and $+\infty$
4. every H_0 decision statistic as a threshold, plus $-\infty$ and $+\infty$
5. thresholds selected so that P_{FA} is linearly sampled from 0 to 1 at an interval of 0.01 (101 samples of P_{FA})

In the questions that follow you will compare and comment on these 5 ROCs.

For methods where you may not be able to satisfy the criteria for selecting thresholds *exactly*, select thresholds in a way that *as closely as possible* meets the desired criteria. What you should have upon completing this exercise is another "tool" for your "machine learning toolbox" that you can call on to generate an ROC for any machine learning problem you work on.

[2] It seems the "No Free Lunch" Theorem generalizes to computational goals beyond classification!

[3] If you are using Matlab, I recommend `perfcurve` (from the Statistics and Machine Learning toolbox) over `roc` (from the Neural Network toolbox) for generating an ROC curve because `roc` assumes the decision statistics fall in the range [0,1] while `perfcurve` makes no such assumptions.

[4] It is good practice to include both $-\infty$ and $+\infty$ as thresholds to ensure that the ROC spans from $(\mathrm{P}_{\mathrm{D}},\mathrm{P}_{\mathrm{FA}}) = (0,0)$ to $(\mathrm{P}_{\mathrm{D}},\mathrm{P}_{\mathrm{FA}}) = (1,1)$.

(10) 4. Compare the 5 ROCs for the decision statistics provided in the file `moderateData.csv` by plotting them on the same set of axes.

(a) Which of the 5 approaches to selecting thresholds would you consider to be appropriate for this set of decision statistics? Why?

(b) Which would you consider to be inappropriate for this set of decision statistics? Why?[5]

(10) 5. Compare the 5 ROCs for the decision statistics provided in the file `bigData.csv` by plotting them on the same set of axes.

(a) Which of the 5 approaches to selecting thresholds would you consider to be appropriate for this set of decision statistics? Why?

(b) Which would you consider to be inappropriate for this set of decision statistics? Why?

(10) 6. Compare the 5 ROCs for the decision statistics provided in the file `smallData.csv` by plotting them on the same set of axes.

(a) Which of the 5 approaches to selecting thresholds would you consider to be appropriate for this set of decision statistics? Why?

(b) Which would you consider to be inappropriate for this set of decision statistics? Why?

(10) 7. Compare the 5 ROCs for the decision statistics provided in the file `logNormalData.csv` by plotting them on the same set of axes.

(a) Which of the 5 approaches to selecting thresholds would you consider to be appropriate for this set of decision statistics? Why?

(b) Which would you consider to be inappropriate for this set of decision statistics? Why?

(10) 8. For each of the five approaches for selecting thresholds to generate an ROC considered here, explain why it is, or is not, universally applicable, meaning it will provide a good representation of the ROC without unnecessary computations. (You should provide 5 explanations – one explanation for each approach.)

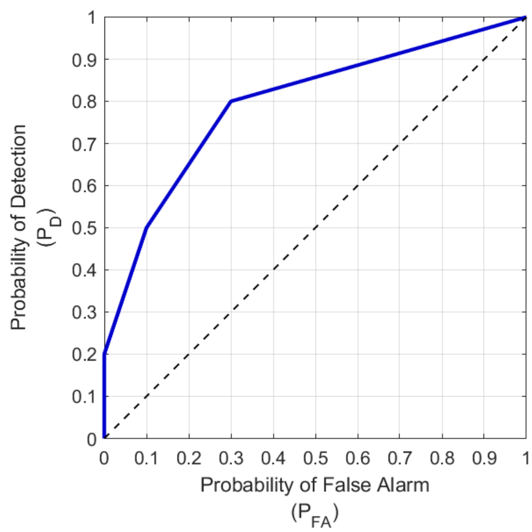
(5) 9. Submit a PDF print-out of your code for this section (Generating ROCs). (Submitting a URL for a cloud-based repository is insufficient.)

[5] Each of the 5 approaches should be categorized as either appropriate or inappropriate!

Probabilistic Decision Rules

Although ROCs for k-Nearest Neighbor (KNN) classifiers can provide (at most) only $k+2$ samples of the ROC because KNN results in one of $k+1$ possible decision statistics $\lambda(x) \in \left\{0, \frac{1}{k}, \frac{2}{k}, \ldots, \frac{k-1}{k}, 1\right\}$ (so $\beta = \left\{0, \frac{1}{k}, \ldots, 1, +\infty\right\}$ to ensure the ROC spans from $(\mathrm{P}_{FA}, \mathrm{P}_D) = (1,1)$ to $(\mathrm{P}_{FA}, \mathrm{P}_D) = (0,0)$), we are not restricted to operating at one of the $k+2$ samples of the ROC found by sweeping through all possible thresholds. It is possible to choose an operating point on the straight line connecting two samples of the ROC by using a probabilistic decision rule.

The `csv` file `knn3DecisionStatistics.csv` contains a set of decision statistics produced by a KNN classifier with $k = 3$. This csv file is organized such that each row contains the true class (either 0 or 1) followed by the associated decision statistic. When you plot the ROC curve, you should see this:



(5) 10. This ROC has operating points with $\mathrm{P_D} = 0.8$ for $\beta = 1/3$ and $\mathrm{P_D} = 1$ for $\beta = 0$. You want to operate at $\mathrm{P_D} = 0.95$. What probabilistic decision rule will allow you to operate, on average, at $\mathrm{P_D} = 0.95$?

(10) 11. Simulate your probabilistic decision rule a large number of times (say, 100) to generate many estimates of the $(\mathrm{P}_{FA}, \mathrm{P}_D)$ pair produced by your probabilistic decision rule. Each individual simulation provides an estimate of the operating point $(\hat{\mathrm{P}}_\mathrm{D}, \hat{\mathrm{P}}_\mathrm{FA})$.[6]

(a) Estimate the pdfs of $\hat{\mathrm{P}}_\mathrm{D}$ and $\hat{\mathrm{P}}_\mathrm{FA}$ via kernel density estimation and find the expected values of $\hat{\mathrm{P}}_\mathrm{D}$, $\mathbb{E}\left\{\hat{\mathrm{P}}_\mathrm{D}\right\} = \mu_{\hat{\mathrm{P}}_\mathrm{D}}$, and $\hat{\mathrm{P}}_\mathrm{FA}$, $\mathbb{E}\left\{\hat{\mathrm{P}}_\mathrm{FA}\right\} = \mu_{\hat{\mathrm{P}}_\mathrm{FA}}$, to show that although you may not achieve $\mathrm{P_D} = 0.95$ in any given simulation, the expected value of $\mathrm{P_D}$ over many simulations is 0.95.

(b) Plot the ROC with all the $(\hat{\mathrm{P}}_\mathrm{D}, \hat{\mathrm{P}}_\mathrm{FA})$ pairs produced by each individual simulation and as well as the expected operating point $(\mu_{\hat{\mathrm{P}}_\mathrm{FA}}, \mu_{\hat{\mathrm{P}}_\mathrm{D}})$ from your probabilistic decision rule superimposed on the ROC.
Do the individual operating points cluster around the ROC in the vicinity of $\mathrm{P_D} \approx 0.95$?
Does your expected operating point have $\mathrm{P_D} \approx 0.95$ and fall on (or very near to) the line connecting the $\beta = 1/3$ and $\beta = 0$ operating points?

(5) 12. Submit a PDF print-out of your code for this section (Probabilistic Decision Rules). (Submitting a URL for a cloud-based repository is insufficient.)

[6] The decision rule is *probabilistic*, so every time the decision rule is applied to the list of decision statistics a different set of decisions will be produced and the corresponding $(\hat{\mathrm{P}}_\mathrm{D}, \hat{\mathrm{P}}_\mathrm{FA})$ pair will be different.