Emma LaPorte

Pattern Classification and Recognition
ECE 681

Spring 2019

Homework #1: Performance Evaluation (ROC Curves)

Due: 5:00 PM, Tuesday, January 29, 2019

Grace Period Concludes: 11:30 PM, Friday, February 1, 2019

This homework assignment is worth **200 points**.

Each problem is worth some multiple of 10 points, and will be scored on the below letter scale.

The letter grades B through D may be modified by + (+3%) and A through D may be modified by a - (-3%).

- A+ = 100%: Exceeds expectations, and no issues identified
- A = 95%: Meets expectations, and (perhaps) minor/subtle issues
- B = 85%: Issues that need to be addressed
- C = 75%: Significant issues that must be addressed
- D = 65%: Major issues, but with noticeable perceived effort
- F = 50%: Major issues, and insufficient perceived effort
- Z = 30%: Minimal perceived effort
- N = 0%: Missing, or no (or virtually no) perceived effort

Your homework is not considered submitted until both components (**self-contained pdf file** and your code) have been submitted. Please do not include a print-out of your code in the pdf file.

The majority of homework assignments this semester will involve coding. For example, in this homework you will be writing code to generate and plot ROCs. I strongly suggest making use of data structures so the input/output argument lists for your functions cannot become unwieldy. For example, instead of `[pd,pfa,auc,...] = generateROC(decisionStatistics,truth,thresholds,...)`, I suggest something like[1] `[roc] = generateROC(classifierOutput,rocOptions)`, where `roc` is a structure with fields `pd`, `pfa`, `auc`, and is extensible to also contain any other information about the ROC, such as the maximum probability of correct decision, `classifierOutput` is a structure with fields `decisionStatistics`, `truth`, and optionally may also contain other information regarding the classifier and its output even if that other information is not needed to generate the ROC, and `rocOptions` is a structure with fields that contain the information necessary to select the thresholds (method for selecting the thresholds and parameter(s) the threshold selection method may need).

I also suggest thinking about how to modularize your code. For example, instead of calculating performance metrics (e.g, AUC, maximum probability of correct decision) within your function that generates the ROC, consider having separate functions (that can be called by the function that generates your ROC) to calculate these performance metrics. If the functions that calculate the performance metrics are separate from the function that generates the ROC, then you will have flexibility to find these metrics for separately from generating an ROC.

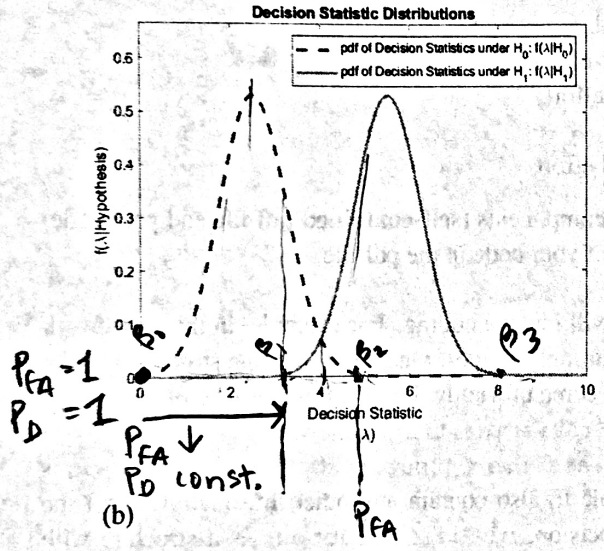
[1] Note I am not specifying the syntax for the code you write or use. I am suggesting you think about how you can structure and organize your code so it can be extended to additional use cases "easily".

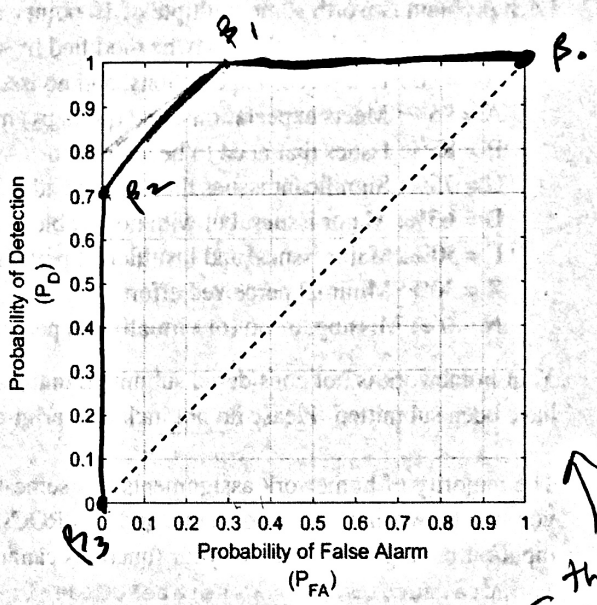
Decision Statistics ↔ ROCs

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.

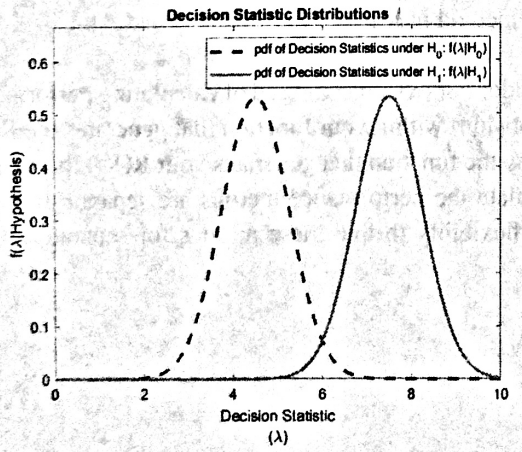
(20) 1. For each set of decision statistic distributions given below, sketch (qualitatively, but as accurately as you can) the corresponding ROC. We are not concerned about the precision of the ROC, but rather its general shape and relative position within the ROC axes.

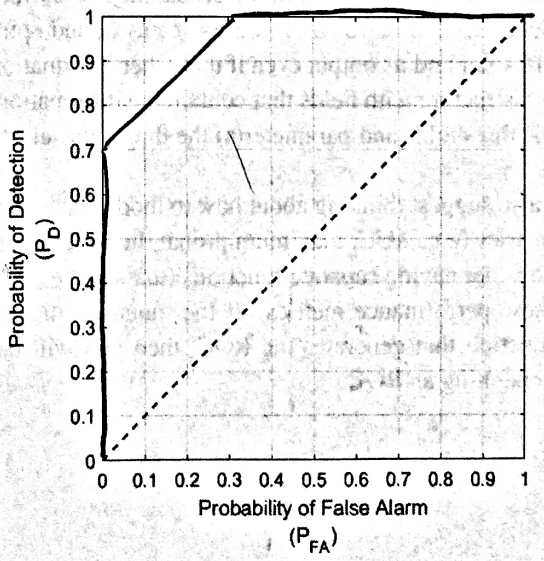
(a)





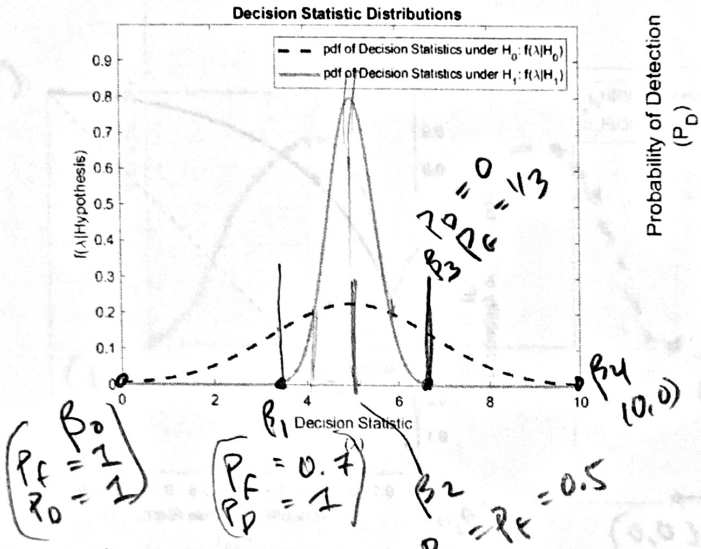
(b)

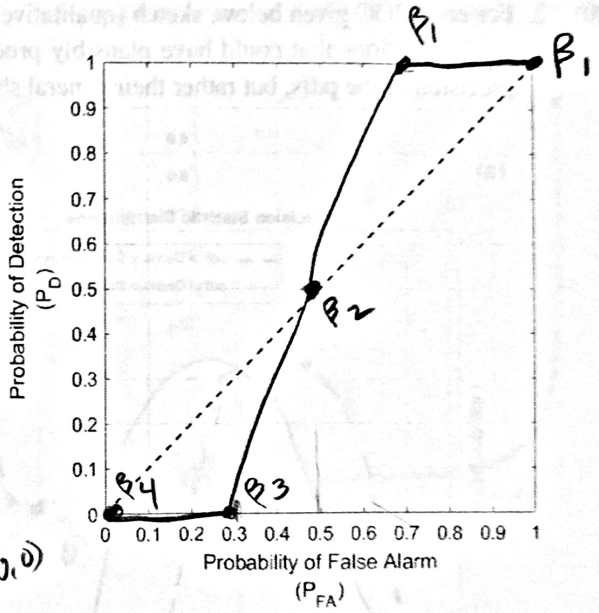




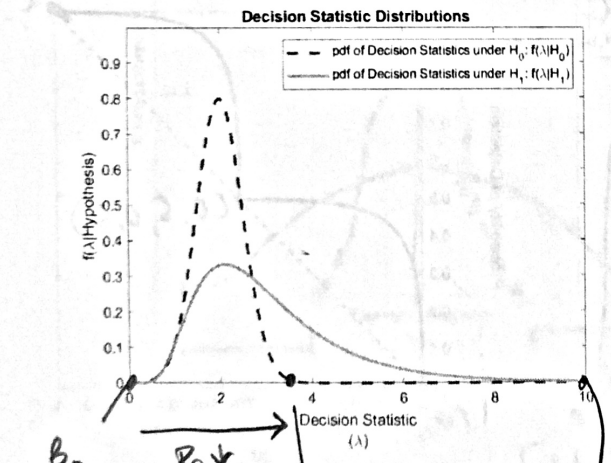
these 2 are the same because all that matters is λ relative to β, not the actual λ values.

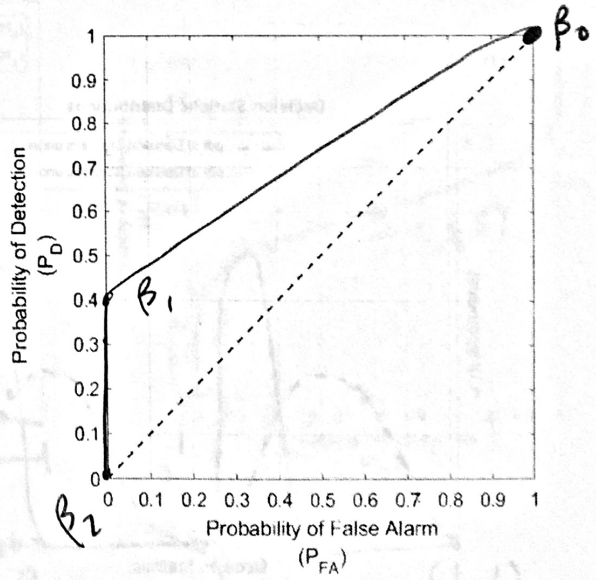
(c)


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(λ|Hypothesis)
Decision Statistic
(λ)


Probability of Detection
(P_D)
Probability of False Alarm
(P_{FA})

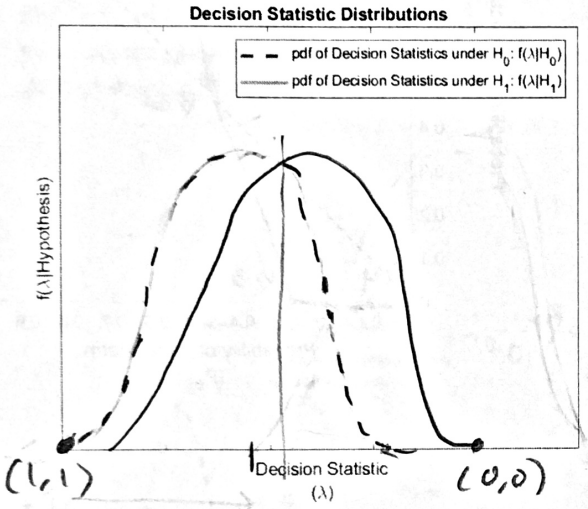
(d)

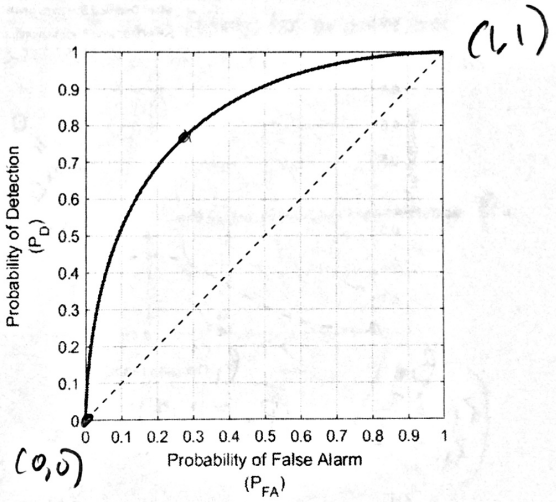

Decision Statistic Distributions
pdf of Decision Statistics under H_0: $f(\lambda|H_0)$
pdf of Decision Statistics under H_1: $f(\lambda|H_1)$
f(λ|Hypothesis)
Decision Statistic
(λ)


Probability of Detection
(P_D)
Probability of False Alarm
(P_{FA})

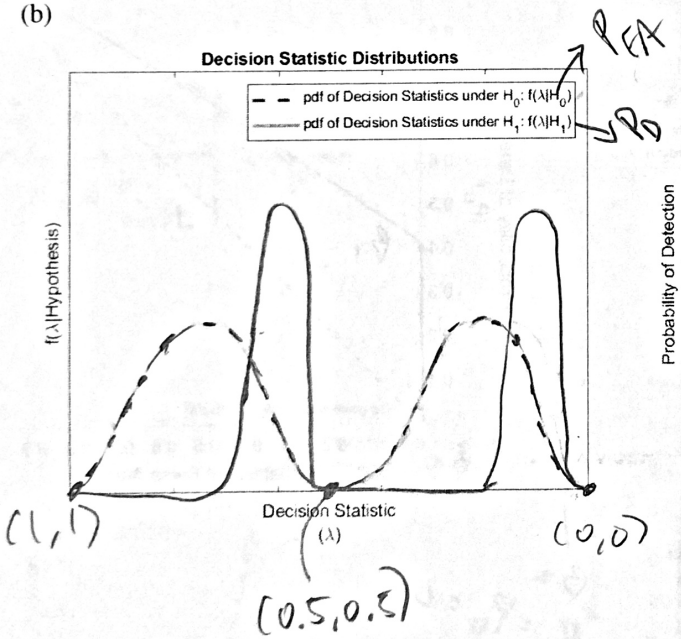
(20) 2. For each ROC given below, sketch (qualitatively, but as accurately as you can) a set of decision statistic distributions that could have plausibly produced the given ROC. We are not concerned about the precision of the pdfs, but rather their general shapes and locations relative to each other.

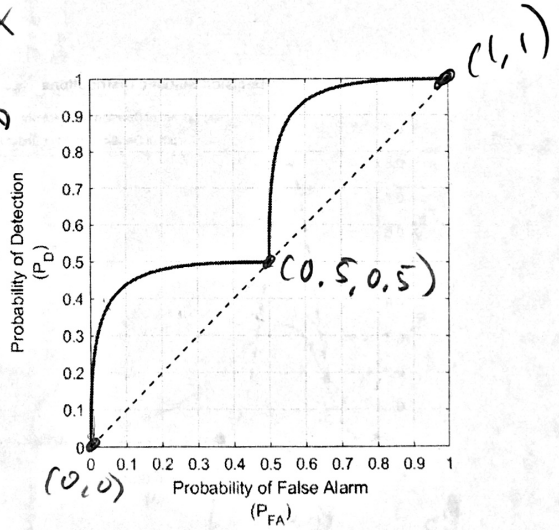
(a)



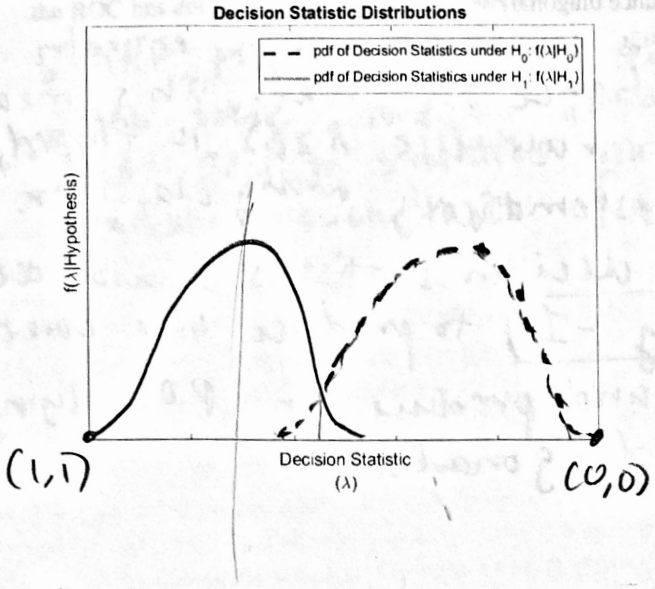


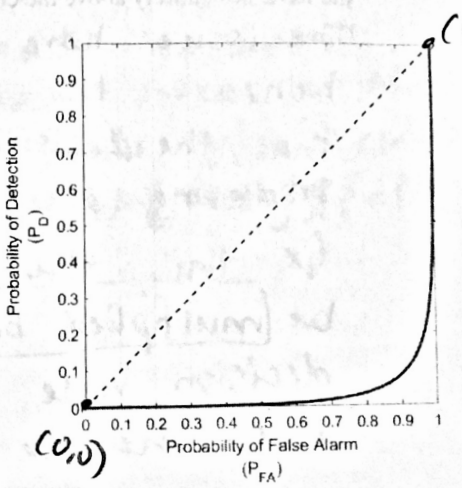
(b)





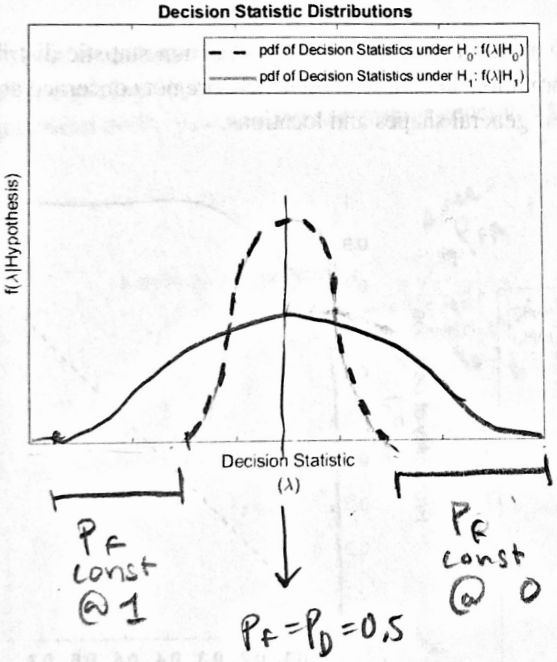
(c)

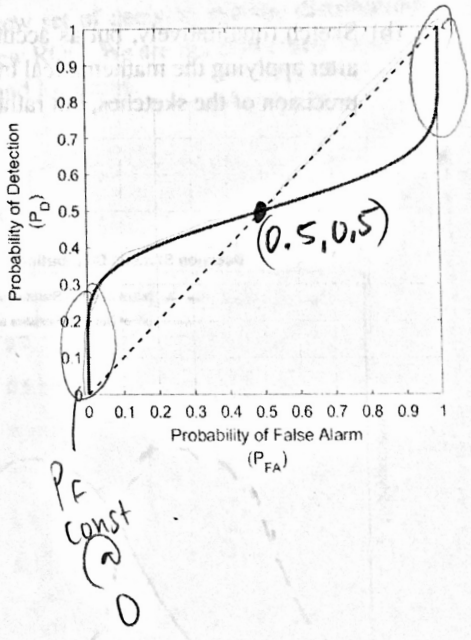




the decision rule needs to be switched so that this reflects over chance diagonal.

(d)

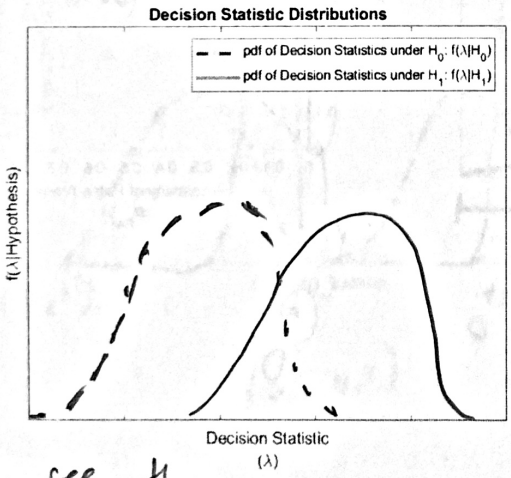




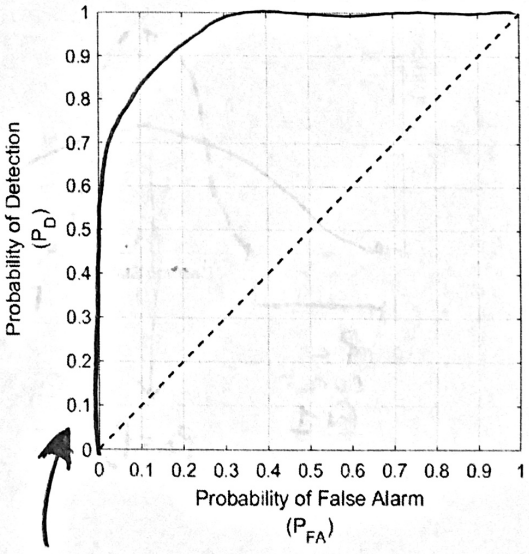
(10) 3. (a) What mathematical transformation could we apply to the decision statistics for question 2c so that the ROC lies entirely above the chance diagonal?

The issue here is that the ROC is entirely beneath the chance diagonal. This means that the decision rule (ie $\lambda \geq \beta$) is flipped, producing a systematically wrong classifier. To fix this, the decision statistics should all be multiplied by -1 to produce the correct decision rule and produce an ROC lying above the chance diagonal.

(b) Sketch (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.



As you can see, the new decision stat distributions are flipped relatively given the act of multiplying λ's by -1.

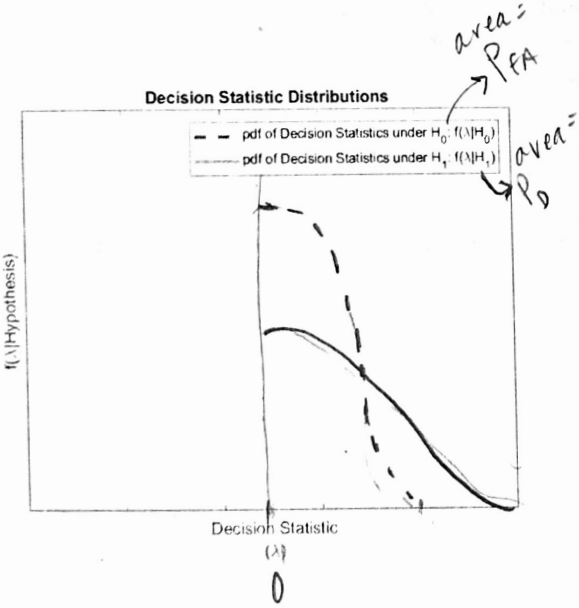


The new ROC is reflected over the chance diagonal.

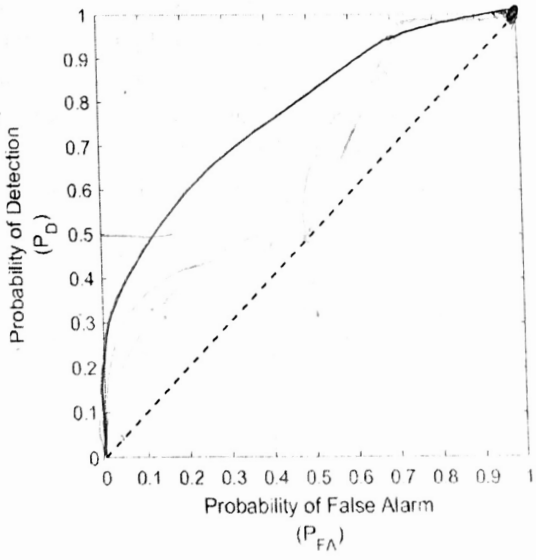
(10) 4. (a) What mathematical transformation could we apply to the decision statistics for question 2d so that the ROC lies entirely above the chance diagonal?

Center the pdf's such that the mean is 0, then take the absolute value of all the decision statistics and scale the decision statistics such that $\int f(\lambda|H_0) = 1$ and $\int f(\lambda|H_1) = 1$

(b) Sketch (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.



make sure area is still = 1



abs(λ) - assuming $\eta = 0$.

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.

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] → inefficient for large data sets → fine for small datasets
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 $+\infty$) → you miss important info for certain distributions
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 $+\infty$ → will work best for
4. every H_0 decision statistic as a threshold, plus $-\infty$ and $+\infty$ → could be weird if # λ's is small
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}) → a little better than 3

with only 99 pieces can't generalize.

similar to 3. can't catch detailed info

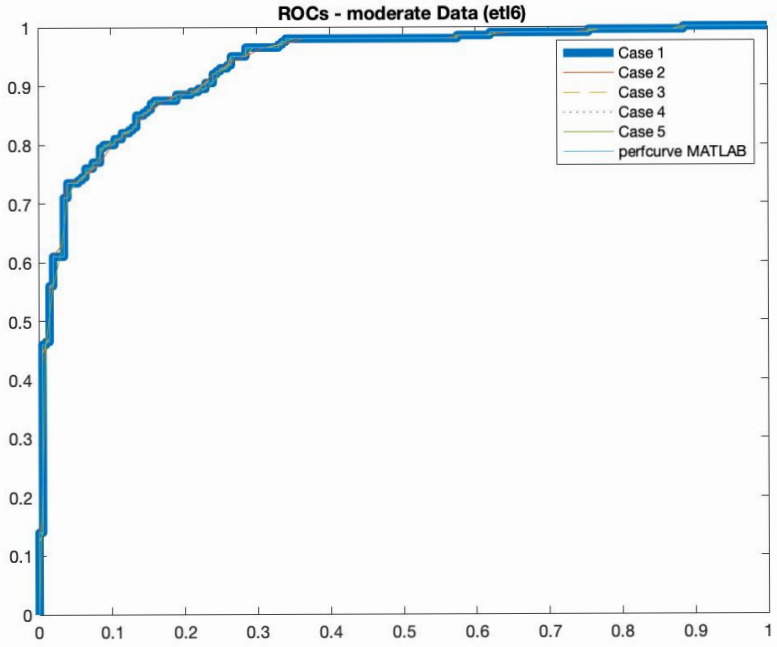
In the questions that follow you will compare and comment on these 5 ROCs.

[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 $(P_D,P_{FA}) = (0,0)$ to $(P_D,P_{FA}) = (1,1)$.

(20) 5. 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?

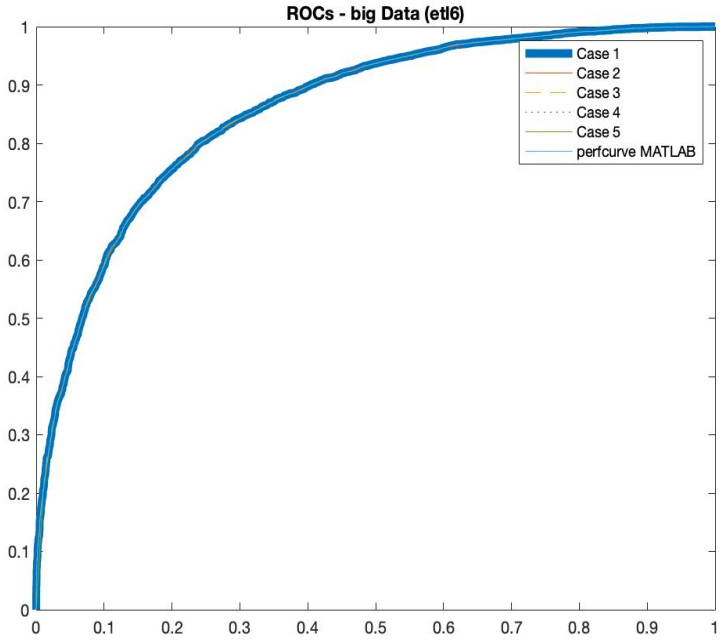
Approaches 2, 3, 4, and 5. These approaches appropriately sample the the data creating ROC curves using 100 - 200 thresholds. These ROC curves are robust and provide the relevant information to understand how the classifier is performing.

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

Approach 1 inappropriately oversamples that data, as it requires 2-4x more threshold values to produce very similar information to the other approaches. Interestingly, MATLAB's perfcurve function also oversamples the data with the same number of thresholds as Approach 1.

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

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



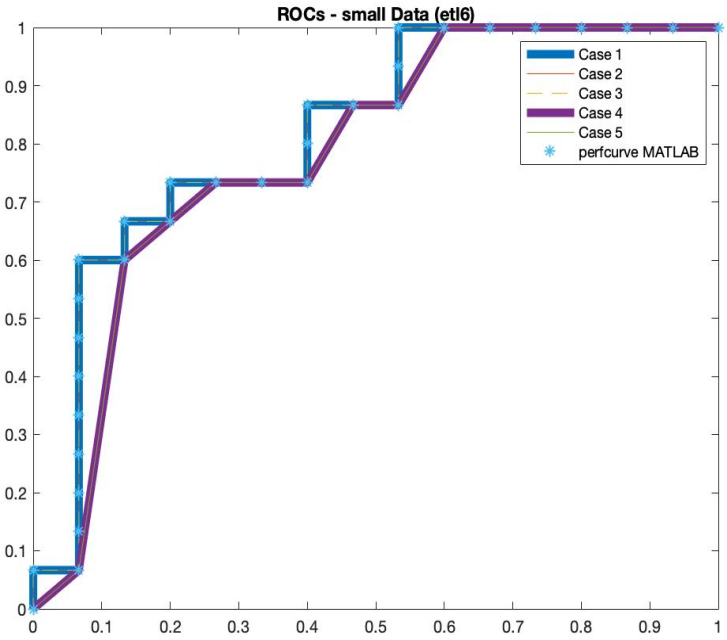
(a) Which of the 5 approaches to selecting thresholds would you consider to be appropriate for this set of decision statistics? Why?

Approaches 2, 3, 5 are appropriate for this set of decision stats. They provide the relevant info in a succint # of thresholds.

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

Approaches 1, 4 oversample the decision stats and are inefficiently computing values that do not add value to the info presented on the ROC curve.
Interestingly, MATLAB's perfcurve function also seems to oversample – using 9,519 thresholds

(20) 7. 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?

Approaches 1,3 appropriatly sample the data, making use of the limited data. They display all relevant info in a succinct # of thresholds.

The MATLAB command perfcurve also defaults to using same number of thresholds as approaches 1,3. and thus appropriately samples the data.

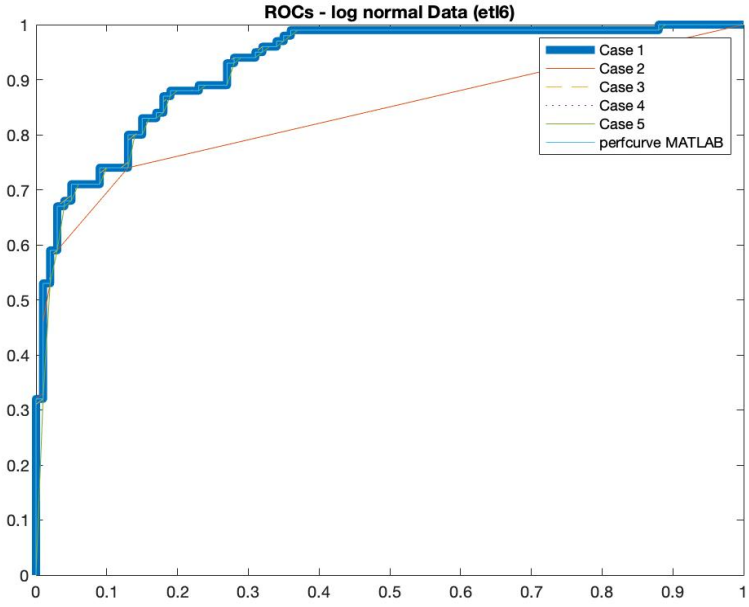
(b) Which would you consider to be inappropriate for this set of decision statistics? Why?

Approach 2 produces same curve as Approaches 1 and 3, but uses larger # of thresholds inefficiently.

Approach 4 seems to undersample + not provide all relevant info. Same with Approach 5.

It is important to note here that with such a small amount of data it is hard to trust the consistency + the

(20) 8. 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?

Approaches 1, 3, 4, 5 are appropriate. They all robustly describe the data's ROC curve in a succint, reasonable # of threshold values. MATLAB's perfcurve defaults to an appropriate sampling of the data as well, at 201 values.

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

Approach 2 is inappropriate for this data distribution. When the threshold values are simply linearly spaced btwn min(λ) and max(λ) you can lose important info for certain distributions, as exemplified above.

(20) 9. 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.

There is no method of selecting threshold values that is universally applicable in every situation. There are advantages and disadvantages to the threshold selection process that mostly depend on the decision statistics of some data set and classification problem. In some cases an approach will provide a good representation of an ROC without inefficiencies and unnecessary computations, while for a different data set it will not.

Approach 1 is very inefficient for large datasets. Especially in a typical machine learning classification problem, a lot of data is required to train the algorithm. Producing ROCs using every single decision statistic for each data point is an incredibly inefficient solution for a large dataset.

Approach 2 is more computationally efficient for some datasets as it uses just 99 threshold values. However, this few of thresholds can be hugely under sampling for a larger dataset, and doesn't capture all the relevant information to accurately depict the ROC. Also, by linearly sampling the decision statistics other distributions of decision statistics won't be represented fully enough through the ROC. For example, the log normal data was not represented well by this approach.

Approach 3, similar to Approach 2, is more computationally efficient for some datasets as it uses just 99 threshold values. However, this few of thresholds can be hugely under sampling for a larger dataset, and doesn't capture all the relevant information to accurately depict the ROC. This won't generalize well to larger datasets.

Approach 4 is convenient as it is a way of having P_{FA} linearly sampled along the x-axis of the ROC curve. However, it depends on the number of H_{0} truths. This number could be either under sampling or over sampling a dataset, depending on their relative sizes.

Approach 5 is similar to Approaches 2 and 3 in that it reduces the number of threshold values to make creating the ROC more efficient, however can hugely under sample the data if the data set is large. Additionally, it is similar to Approach 4 in that it linearly samples P_{FA}. This is a good "go-to" approach, but is not universal to every situation.

ROCs ↔ Summary Performance Measures

It is helpful to be able to compare ROCs more quantitatively than visually evaluating which is closer to the upper left corner of the ROC graph. Two summary performance measures that are commonly used are AUC (the area under the ROC curve) and the mmaximum probability of correct decision, $\max P_{cd}$, (or, equivalently, the minimum probability of error, $\min P_e$).

Make sure you have a way to find summary performance measures for an ROC. You should be able to find AUC, $\max P_{cd}$, and $\min P_e$.

(10) 10. How are $\max P_{cd}$ and $\min P_e$ related?

$$P_{cd} = P_D P(H_1) + \underbrace{(1-P_{FA})}_{P_{CR}} P(H_0) \xrightarrow{\text{maximize}} \uparrow P_D, \downarrow P_{FA}$$

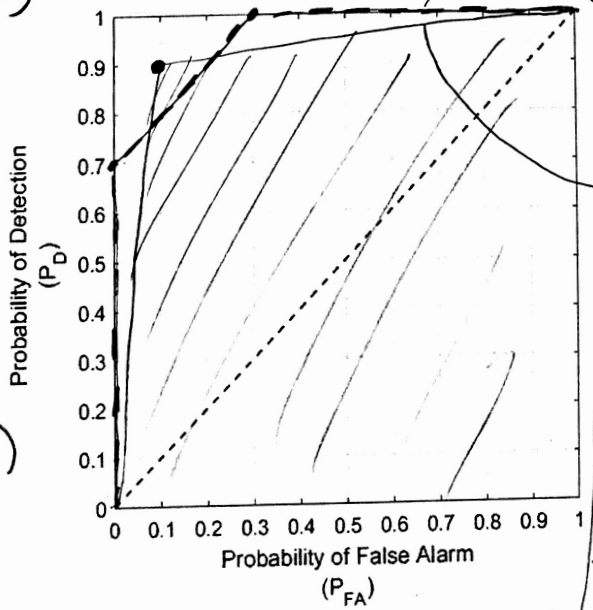
$$P_E = \underbrace{P_M}_{1-P_D} P(H_1) + P_{FA} P(H_0) \xrightarrow{\text{minimize}} \uparrow P_D, \downarrow P_{FA}$$

As seen by the equations above, $\max P_{CD}$ requires $\uparrow P_D, \downarrow P_{FA}$ and $\min P_E$ requires $\uparrow P_D, \downarrow P_{FA}$. This mathematically implies that $\max P_{cd}$ and $\min P_E$ happen at the same operating point of an ROC curve — or at the same threshold value giving some (P_D, P_{FA}) pair. Also, $\boxed{P_E + P_{CD} = 1.}$

(10) 11. Assume the priors on the two hypotheses are equal ($p(H_0) = p(H_1) = 0.5$), and sketch two ROCs where the ROC with the higher AUC has lower $\max P_{cd}$.

$$P_{cd} = P_D \cdot P(H_1) + \underbrace{P_{CR}}_{1-P_{FA}} \cdot P(H_0)$$

max P_{cd} happens when $P_D \uparrow$, $P_{FA} \downarrow$



AUC - area under curve

Curve A
→ larger AUC
→ max P_{cd} = 0.85

Curve B
→ smaller AUC
→ max P_{CD} = 0.9

Curve A
- larger AUC (seen visually)
- $P_{cd} = 0.5(0.7) + 0.5(1)$ (max) = $\boxed{0.85}$

Curve B
- smaller AUC (seen visually)
- $P_{cd} = 0.5(0.9) + 0.5(0.9) = \boxed{0.9}$ (max)

This shows one example in which performance metrics choose different classifiers as "better." This speaks to the importance of knowing the context and choosing what constitutes "best performance" from that.

$P(H_1) + P(H_0) = 1$ always

$p(H_0) + 2p(H_0) = 1$
$3p(H_0) = 1 \Rightarrow$

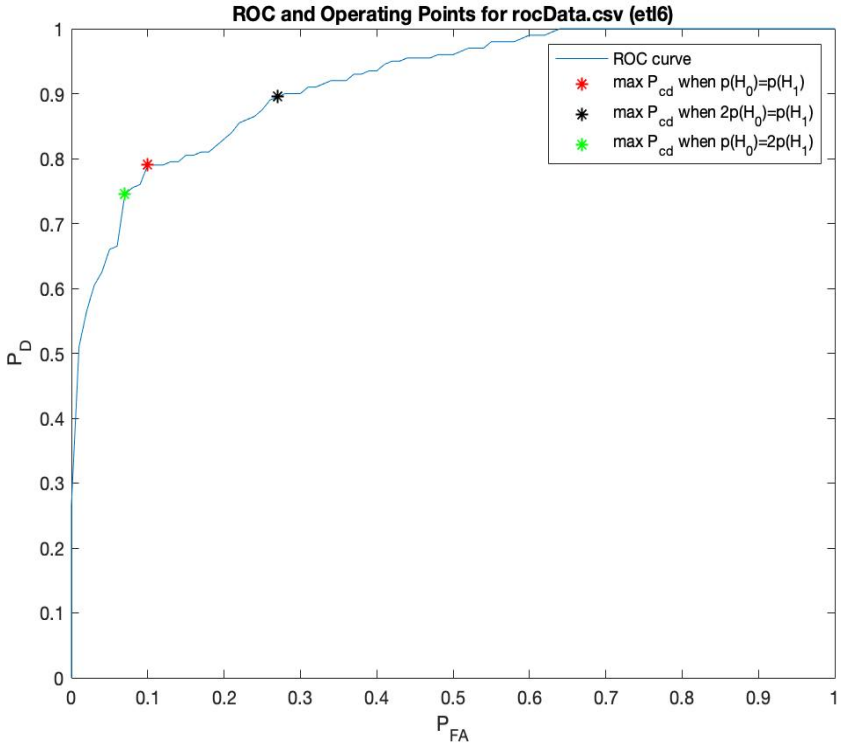
(20) 12. The csv file `rocData.csv` contains (P_{FA}, P_D) pairs for an ROC curve, and is organized such that each row contains P_{FA} followed by P_D.

$P_{cd} = P_D P(H_1) + (1 - P_{FA}) P(H_0)$

(a) On a single set of axes, plot:

1. the ROC curve
2. the operating point corresponding to max P_{cd} when $p(H_0) = p(H_1) = 0.5$
3. the operating point corresponding to max P_{cd} when $2p(H_0) = p(H_1) \Rightarrow P(H_0) = 1/3, \; P(H_1) = 2/3$
4. the operating point corresponding to max P_{cd} when $p(H_0) = 2p(H_1) \Rightarrow P(H_0) = 2/3, \; P(H_1) = 1/3$

(b) Provide the AUC for the ROC and the maximum P_{cd} for each of the three operating points. These performance metrics can be provided either in the legend of the graph, or included as separate text below the graph.



AUC = 0.9166

max P_{cd} when...

$p(H_0) = p(H_1)$: 0.8450

$2p(H_0) = p(H_1)$: 0.84

$p(H_0) = 2p(H_1)$: 0.8683