University of Victoria
Examinations April 2020 [On-line Exam]

**ECE 485:**
**Data Analysis and Pattern Recognition A01**

(CRN 20935)

Name: Jordan Carlson

Student No. V0 0714886

Instructor: Stephen W. Neville

Section: A01
Duration: 3.0 Hours

TO BE ANSWERED ON THE EXAM PAPER

STUDENTS MUST COUNT THE NUMBER OF PAGES IN THIS EXAMINATION PAPER BEFORE BEGINNING TO WRITE, AND REPORT ANY DISCREPANCY IMMEDIATELY TO THE INVIGILATOR.

THIS QUESTION PAPER HAS 8 PAGES.

TOTAL MARKS: 100

ATTEMPT ALL QUESTIONS.

ALL FIVE (5) QUESTIONS WILL BE COUNTED FOR YOUR GRADE.

EACH QUESTION IS WORTH 20 MARKS.

# On-line Exam Academic Integrity Statement:

All students writing this on-line exam **must** abide by UVic academic regulations and observe standards of 'scholarly integrity,' (no plagiarism or cheating).

As such, this online exam *must* be taken individually and not with a friend, classmate, or group and while only accessing the allowed information sources, as listed below. You are also prohibited from sharing any information about the exam with others.

PLEASE ENTER YOUR NAME AND V0 _ NUMBER INTO THE PLEDGE TEXT BELOW AND SIGN AND DATE THE STATEMENT.

*I,* Jordan Carlson *, V0* 0714886 *affirm and confirm that I will (and have) completed this exam independently, all provided answers will be (and are) solely my own work, and that I will not make (and have not made) use of any unauthorized materials in my completion of this exam.*

Signature: Date: April 24, 2020

Note: All exams submitted without a signed academic integrity statement will not be marked and will received a zero grade.

## *Exam Notes*:

- On-line mathematical resources such as Matlab, Wolfram Alpha, MS Execl, etc. *can* be used to answer exam questions, *but* the resulting code, Excel sheet, etc. *must* be submitted along with your answer. Answers submitted without this supporting material, if used, will receive a ZERO grade.
- No other aids permitted. Written answers may be handwritten or word processed.
- All work must be shown. Answers without worked solutions will receive a ZERO grade.
- All submitted exam materials *must* have your student name and number clearly denoted on them, i.e, on all submitted code, on each and all scanned exam pages, etc.
- All uploaded exam materials *must* be either in .pdf or .zip formats, with the pdfs readable by standard pdf readers, i.e., Acrobat Read. Any materials uploaded in any other formats will NOT be graded. Matlab LiveScripts *must* be converted into .zip files before uploading.
- The academic integrity statement must be signed for the exam to be eligible to be marked. Exams with unsigned statements will not be marked and will receive a ZERO grade.
- The exam includes the exam paper as well as the associated data files.
- Students are solely responsible for ensuring the legibility and accessibility of any and all uploaded materials. Illegible answers or answers inaccessible using standard technologies will not be marked and will receive a ZERO grade.
- Security permissions and passwords must not be set on uploaded .zip or .pdf files. Any uploaded materials with set security features and/or passwords will not be marked and will receive a ZERO grade.
- *Do not* change or rename the provided data files as the original files will be the files that your Matlab LiveScripts will be executed and marked against.
- Note: Basic Matlab functions, such as mean(.), cov(.), eig(.), goodness-of-fit tests, plot(.), contours, etc. can be used. But, functions which effectively solve the major bulk of what is being asked cannot be used, e.g. using a pca(.) function to do principal component analysis. To receive marks you need to show that you are able to do what is being asked, not that Matlab has a function that someone else has already created to do what is being asked.
- Submitted code may be checked, including via automated tools, for similarity to other student code or Internet available code, with any code determined to be highly (overly similar) similar receiving a ZERO grade.

- All submitted LiveScript solution *must* begin with the command ”*clear all*” as the first executable statement in the LiveScript.

1. Develop a Matlab LiveScript to:

    (a) Load the data file Data Ques 1, which contains a 25 dimensional 1000 sample data set as the variable *Data*, where is data sample is a row, and a 1000 × 1 vector *Classes*, denoting the class labels for each data point with Class 1 denoted by a 1 and Class 2 by a 2 within this vector.

    (b) Apply Fisher Discriminant analysis to reduce this data to a 2-dimensional data set that best linearlyseparates the two classes.

    (c) Produce a scatter plot of this generated 2-dimensional data, with Class 1 plotted in blue and Class 2 plotted in red.

    (d) Formally determine whether the reduced data represents Gaussian distributed classes and, if, so whatare the $p(x|\omega_1)$ and $p(x|\omega_2)$ distributions?

    (e) Determine the decision boundary between Class 1 and Class 2 in the reduced feature space.

    (f) Discuss whether (or not) the application of Fisher Discriminant analysis will *always* improve classifier performance.

    *Note:* Your LiveScript *must* be appropriately commented using Text sections such that it is clear what each subsequent Code section is intended to do (and does). You will submit your LiveScript code as your answer to this question. This LiveScript should be named ”firstname lastname ques1 ans.mlx”, using your first and last names.

Answer for (F)

Fisher Discriminant Analysis is useful for supervised learning and for finding classifications in which class separation can be found. Fisher discriminant analysis is not always useful in that it assumes Gaussian distributions and equal class covariances. If the distribution is not Gaussian or the class covariance are highly varying, the analysis will not yield appropriate results

2. Discuss the implications of the No Free Lunch and Ugly Duckling theorems on modern data analysis andpattern recognition techniques such as Deep learning, neural networks, etc.

   Additionally, discuss these issues with respect to the particular problem of seeking to develop a machine learning image recognition solution to be used at full production-scales for self-driving cars, i.e., where the image space would be on the order of $10^{15M}$, the available data would be on the order of $10^{16}$, and full production-scale would mean tens to hundreds of millions of self-driving vehicles on the road.

The No Free Lunch Theorem is the proven idea that there is "No Free Lunch" in algorithms. This means that any given algorithm that is best for situations will lose efficacy when applied to other algorithms. Clearly this is an important baseline in pattern recognition and deep learning, as many algorithms are needed to both begin a classification approach, and to fine-tune pattern-classification algorithms to form useful tools for use. For use in self-driving cars, the big data involved will need to be uniquely classified with distinct algorithms, which proves the necessity of pattern recognition expertise. Self-driving cars for example will need to detect new types of road infrastructures that are created, so awareness of algorithms in detecting unseen objects will need to be monitored to maximize safety.

The no free lunch theorem at the largest scale proves the need for fine-tuning and changing of algorithms, and proves the need for independence of algorithms and the motivation for creating effective algorithms.

The No Free Lunch Theorem is complemented by the Ugly Duckling Theorem, which states that biases are the foundation of classification and extends to explain how biases can be attributed significantly but finitely to all objects.

The Ugly Duckling Theorem proves the motivation to try new approaches to algorithms, and especially to find appropriate amounts of similarities in pattern recognition problems to maximize algorithm efficacy. In self-driving cars, for example, lines on the road will have to be recognized as different, such as forks in the road of all types, but similar enough to have the driver cross them sometimes while maintaining safe distance at others.

3. Two approaches commonly arising within current machine learning-based (ML-based) data analysis are:

   - Training a ML classifier on a random selection of 90% of the available data and then testing (assessing) the classifier's performance on the remaining 10% of the data, i.e., assessing the ML performance based on the data not seen during training.
   - Reporting the classifier's performance in terms of a Receiver-Operator Curve (ROC), which denotes the hit probability versus the false alarm probability, that the trained ML classifier achieved over its test data.

   (a) Discuss the issues with a 90/10 training regime and the conditions where it will and will not lead to atrained ML classifier that correctly and properly generalizes.

   (b) Discuss the issues with such uses of ROC curves and whether and when such an approach would andwould not be appropriate and informative.

(a)

The 90/10 training regime, as expressed in the No Free Lunch Theorem and Ugly Duckly Theorem, has its uses but pitfalls. For example, when using a small dataset, the 90/10 regime alone, along with many other ratios, has a higher chance of leading to an inaccurate Maximum Likelihood classifier. The 90/10 regime may prove as the best ratio if the CPU resources are limited, or the data has a low variance, and especially when combined with other validation techniques such as decision trees and k-fold cross validation.

(b)

ROC curves are useful for determining how well two classes are discriminated by a model. The effectiveness of the ROC curve depends on if the ROC curve itself can be calibrated accurately before classifying data. The ROC curve can determine the amount of false negatives and positives, which may be useful for further MLE determination and verification of which data is more useful and accurate and in which respect, such as when needing to determine if an estimate may either an overestimation, or underestimation of results (such as when leaving a buffer on a high-risk investment).

The ROC curve fails when the data has varying weights of error magnitude. The ROC curve might tell you how much data is correct and incorrect in which way, but not take into account the severity of the data's miscalculation if the severity is embedded in the data aside from the curve. The Area Under the Curve and the ROC magnitudes may be balanced if there is enough prior information to determine some data magnitude.

4. Develop a Matlab LiveScript to:

   (a) Load the data file Data Ques 4, which contains 10,000 samples taken from a time series stochastic process as the variable *Data* where the first column is the time stamp of each data sample and the second column is the measured data.

   (b) Apply statistical hypothesis testing to determine whether this time series data has a stationary meanand standard deviation.

   (c) Determine whether this time series data has a quasi-stationary mean and/or standard deviation.

   (d) Determine whether the data is wide sense stationary.

   (e) Determine whether the data can be modeled using a Gaussian distribution.

   *Note:* Your LiveScript *must* be appropriately commented using Text sections such that it is clear what each subsequent Code section is intended to do (and does). You will submit your LiveScript code as your answer to this question. This LiveScript should be named ”firstname lastname ques4 ans.mlx”, using your first and last names.

5. The three classes $\omega_1$, $\omega_2$, and $\omega_3$ are given by the sufficient statistics:

$$ {}_{1} = \begin{bmatrix}-10\\10\\10\end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix}6\\14\\-4\end{bmatrix}, \boldsymbol{\mu}_3 = \begin{bmatrix}0\\-2\\12\end{bmatrix},$$

$\mu$

$\boldsymbol{\Sigma}_1 = \begin{bmatrix}24.1321 & 1\\ 1.3489 & 47\\ -3.5186 & 7\end{bmatrix}$ 1.3489 −3.5186⍰ ⍰ 6.2852 −1.1446 −4.5746⍰

47.8791 7.9578 ⍰, $\boldsymbol{\Sigma}_2$ = ⍰−1.1446 9.5775 1.5270 ⍰, and

7.9578 6.0794 −4.5746 1.5270 13.2511

⍰19.3188 −5.6360 0.7915 ⍰

$\boldsymbol{\Sigma}_3$ = ⍰−5.6360 27.9177 −6.2955⍰

0.7915 −6.2955 30.0063

Develop a Matlab LiveScript to:

(a) Load the data file Data Ques 5, which contains three training data sets of 100 samples each from each of $\omega_1$, $\omega_2$, and $\omega_3$ denoted by the 3×100 Matlab variables *Class1*, *Class2*, and *Class3*, where the Matlab variables *m1*, *m2*, *m3*, *S1*, *S2*, and *S3* are the above per-class means and covariances, i.e., you do *not* need to hand enter these per-class sufficient statistics.

(b) Apply a Bayes classifier with equal a prior probabilities to classify the following ground-truthed points:

**Class 1 Points:** $\mathcal{D}_{\omega_1} = \left\{ \begin{bmatrix}-5.6783\\0.3313\\-4.6218\end{bmatrix}, \begin{bmatrix}5.4838\\20.6387\\-0.6436\end{bmatrix}, \begin{bmatrix}10.0027\\3.9438\\6.5321\end{bmatrix} \right\}$

**Class 2 Points:** $\mathcal{D}_{\omega_2} = \left\{ \begin{bmatrix}3.1000\\-0.4120\\-1.9054\end{bmatrix}, \begin{bmatrix}19.7638\\13.1648\\12.0513\end{bmatrix}, \begin{bmatrix}-6.6313\\5.8846\\3.2152\end{bmatrix} \right\}$

**Class 3 Points:** $\mathcal{D}_{\omega_3} = \left\{ \begin{bmatrix}-2.5014\\-7.3041\\2.1365\end{bmatrix}, \begin{bmatrix}-6.3075\\-7.1580\\9.9063\end{bmatrix}, \begin{bmatrix}18.9199\\7.4855\\4.3541\end{bmatrix} \right\}$

**Note**: These points are in the *Data Ques _5* file as the row-wise variables *D w1* , *D w2* , and *D w3*, i.e, the points above are the rows within the respective variables.

(c) Apply a 1-nearest neighbor classifier to classify the points of (b) based on the training data sets containedwith the variables *Class1*, *Class2*, and *Class3* in the *Data Ques _5* file where each row-wise variable contains 100 data samples from the given class.

(d) Explain the classification differences that occur between the (b) and (c) classifiers' performance from theperspective of the known ground-truths for these points and the minimization of Bayes risk. For this discussion also provide a clear table, for all of the points above, delineating each point's actual class, its class as assigned by the Bayes classifier, and its class as assigned by the 1-nearest neighbor classifier.

*Note:* Your LiveScript *must* be appropriately commented using Text sections such that it is clear what each subsequent Code section is intended to do (and does). You will submit your LiveScript code as your answer to this question. This LiveScript should be named ”firstname lastname ques5 ans.mlx”, using your first and last names.

**END**