Introduction to Data Science, WMCS16002, semester 1a 2020

1 Assignment 2

Due September 24, 2020 09:00AM CET
Submit by creating a pull request on GitHub

Your submission should consist of

- A brief written report (preferably PDF (latex template available on Nestor) or dynamic report formats (RMarkdown, etc.)
- Source code files which generate the solution, tables, visualizations used in the report. The code can contain much more than what is finally used in the report — please use comments to structure the source code and provide a README.txt file informing how to run it.
- For Sections 1.2.3 and 1.2.4 where you are provided incomplete Matlab implementation you are expected to submit the full source that includes the given script and the statements that you are expected to fill in. Then, in the report you only write the functions where you are expected to add code.

Note that you are free to use whatever programming language you want if not stated otherwise in the assignment. If you choose languages other than Matlab then you would need to rewrite the code that we provide you as well. Be aware that the difficulty of an exercise might scale differently depending on what language you use!

You have been allocated to work in groups of three to five students to mix different backgrounds. This is an interdisciplinary course therefore we suggest you take advantage of your complementary background (including Mathematics, Astronomy, Engineering and Computing Science). You can indicate in the last section in case you did not contribute equally. If each of you use the GitHub rigorously we could in principle follow your actions in every detail in case of problematic issues.

Remember that you have to pass every homework assignment (but one) to pass the course. Furthermore, note that **plagiarism is fraud** and we take it serious. If we find it in your submissions you risk being expelled.

Submission

- Make a new folder for every assignment (this is the first one). Every assignment's folder should have clear instructions on how to run your code.
- Do not commit massive data files to the repository, leave clear instructions on what and where.

2 Part A

2.1 Exercises: (Total 3 points)

2.1.1 Eigen values and Eigen vectors (1.5 points)

Compute the Eigen values and the L2-normalized Eigen vectors of the matrices A and B. Please show all working (not programming) step by step with clarity.

$$A = \begin{bmatrix} 3 & 4 \\ 5 & 8 \end{bmatrix}, \qquad B = \begin{bmatrix} 4 & 2 \\ 3 & 1 \end{bmatrix}$$

2.1.2 ANOVA (1.5 points)

We split the Introduction to Data Science course of last year into three categories: Attended more than half of the lectures (High), half of the lectures (Medium), and less than half of the lectures (Low). Random samples of students with different sizes were drawn from the three categories, and their final grades are reported in the following table.

High	9, 7, 6.5, 8, 7.5, 7, 9.5, 8, 6.5
Med	7.5, 8, 6, 7, 6.5, 7.5
Low	8, 6, 6, 6.5, 6.5

Please compute the following one-way ANOVA table and for each value show all working step by step with clarity. The abbreviations SS, df, MS, and F stand for sum of squares, degrees of freedom, mean of squares, and F-statistic, respectively.

Source	SS	df	MS	F
Between				
Within				NA
Total			NA	NA

2.2 Implementation (Total 8 points)

2.2.1 Implementation of the F-statistic (2 points)

Write a function in Matlab with the following signature and that computes the F-statistic of a given set of independent and dependent variables.

$$F = \text{myOneWayANOVA}(IV, DV)$$

where the input variables IV and DV are vectors of independent and dependent variables, respectively and the output variable F is the F-statistic. Both IV and DV must have the same size. If DV has only one unique value, then you must raise an error. You can use the data given in Section 1.1.2 and check that the F-value that this function returns is the one that you computed above. NOTE: You MUST NOT use in-built functions that compute the F-statistic directly.

$$IV = \{9,\ 7,\ 6.5,\ 8,\ 7.5,\ 7,\ 9.5,\ 8,\ 6.5,\ 7.5,\ 8,\ 6,\ 7,\ 6.5,\ 7.5,\ 8,\ 6,\ 6,\ 6.5,\ 6.5\}$$
$$DV = \{H,\ H,\ H,\ H,\ H,\ H,\ H,\ H,\ H,\ M,\ M,\ M,\ M,\ M,\ M,\ L,\ L,\ L,\ L,\ L\}$$

2.2.2 Principal Component Analysis (2 points)

Write a function in Matlab with the following signature and that performs principal component analysis using the scalable approach shown in class.

$$[pc, eigenvalues] = \text{mypca}(A)$$

where the input variable A is an $M \times N$ matrix with the rows of A corresponding to observations and columns to variables. The output variables *pc* and *eigenvalues* correspond to the L2-normalized principal components and the respective eigenvalues. Both output variables must be sorted in descending order of the eigenvalues by using the following code:

```
[mx,srtidx] = sort(eigenvalues,'descend');
eigenvalues = eigenvalues(srtidx);
pc = pc(:,srtidx);
```

2.2.3 Application: Face Recognition (4 points)

Download the zip file http://www.cs.rug.nl/~george/IDS/FaceRecognition.zip. It contains a data directory with a data set of face images and the following three Matlab scripts:

a) myEigenFaces.m: It contains an incomplete implementation of Face Recognition using principal component analysis. You **MUST** fill out the missing statements between lines 50 and 55. Use the inline comments for assistance (2 points).
b) myFeatureSelectionwithANOVA.m: It contains an implementation that uses the script myOneWayANOVA.m (mentioned above) and performs feature selection based on ANOVA and it is applied to the same problem as well.

Use the above two scripts to compare the outcome of the two approaches and highlight the advantages and disadvantages of each technique (2 points).

3 Part B

3.1 Exercises: (Total 2 points)

3.1.1 Classification metrics and unbalanced classes (1.5 points)

A binary classification has the following confusion matrix.

		predicted	
		positive	negative
true	positive	15	20
	negative	0	1965

a) Compute the following quantities from the confusion matrix: TPR (true positive rate), TNR (True Negative Rate), F_1 score, recall, precision, specificity, classification accuracy, and Cohen's Kappa. Before computing the numerical values, express all the quantities in terms of TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative), or in terms of other quantities from this list, if applicable.
b) Calculate the Imbalance Ratio (IR) of the dataset. The Imbalance Ratio is the ratio between the number of majority instances to the number of minority instances.
c) Which metric(s) is most suitable to evaluate the performance of the classification represented by the given confusion matrix? Explain your answer.

3.1.2 Feature selection with Information Gain (1.5 points)

Table 1 shows four features that describe a number of days. The last column (output) represents whether a particular cyclist went for a ride or not on those given days. Using Information Gain (Mutual Information), rank the features in terms of their relevance in determining whether the cyclist went on a ride or not. Rank the features from most relevant to least relevant. Show in detail your Information Gain and Entropy calculations.

Table 1: Factors for determining whether or not to cycle.

Sky condition	Temperature	Humidity	Windy	Cycle
Sunny	High	High	False	No
Sunny	High	High	True	No
Cloudy	High	High	False	Yes
Rain	Mid	High	False	Yes
Rain	Low	Low	False	Yes
Rain	Low	Low	True	No
Cloudy	Low	Low	True	Yes
Sunny	Mid	High	False	No
Sunny	Low	Low	False	Yes
Rain	Mid	Low	False	Yes
Sunny	Mid	Low	**True**	Yes
Cloudy	Mid	High	True	Yes
Cloudy	High	Low	False	Yes
Rain	Mid	High	True	No

3.2 Implementation (Total 8 points)

3.2.1 Implementation of single imputation (2 points)

Write a Matlab function with the following signature which performs single imputations:

$$X_{full} = \text{MyImpute}(X_{missing}, S, Options)$$

The function should function as follows:

a) The function takes an input matrix $X_{missing}$ of size $n \times p$, which contains missing values, and returns X_{full} of the same size, which has the missing values imputed.
b) S is a parameter that specifics the type of each feature (continuous or categorical).
c) If a feature is continuous, the function should impute its missing values with the mean. If it is categorical, the function should impute its value with the mode.
d) The function should accept missing values encoded as NaN (or equivalent), and detect them automatically.

3.2.2 Relief feature selection (4 points)

Write a Matlab function that implements the Relief feature selection algorithm for categorical features only. The function should have the following signature and specifications:

$$W = \text{MyRelief}(X, Y, m)$$

a) X contains the input data, while Y contains the output labels (the function needs only to handle binary output labels). W is the feature weight vector.
b) m is the number of random instances that are sampled from the training instances.
c) For each target instance, update the weights of all features whose values differ between the target instance and the nearest hit or nearest miss (i.e. do not subsample the features).

Use the function you created to rank the features in Table 1. Set $n = 14$ so that every instance contributes to updating the weights. Compare the resulting ranking with the ranking you obtained in question 2.1.2.