

## Assignment 1

# Classification and Genetic Algorithm Optimisation

For this assessment, you are asked to perform classification using k-nearest neighbour on a real-world dataset and optimise the features using a binary genetic algorithm (GA). The assessment will be marked by machine<sup>1</sup>. It is therefore very important that you follow the instructions given to you very carefully. Failure to do so will result in substantial loss of marks, **even possibly zero**.

To help you complete the assessment, you are given 3 pieces of shell program codes called `kNN.java`, `kNN_test.java` and `kNN_GA.java`. The first function performs classification of alcoholics vs controls using brain data (EEG) as in the paper<sup>2</sup> but using k-Nearest Neighbour (kNN) classifier; the second will be used in the automated marking. The third function is a template that you could use to develop further for implementing GA to improve the classification performance using fewer selected features. Look at the codes carefully to understand how these work. See below for more details on what is required for submission.

### Problem 1

The first problem is a classification problem. The EEG data was collected from 61 active channels (electrodes, see the reference paper for more information but this is not required to complete this submission) where each channel was used to compute a feature. There are 800 patterns from 40 subjects (roughly similar number of alcoholics and controls). The 800 pattern is divided into 3 sets:

- Training set: 400 patterns (given in `train_data.txt` file)
- Validation set: 200 patterns (given in `val_data.txt` file)
- Test set: 200 patterns (given in `test_data.txt` file)

Each row in the file consists of 61 feature values representing either an alcoholic or control subject data. The class labels for the patterns i.e. either alcoholic or control is given as 0 and 1, respectively but only for training and validation data sets. Class labels for the test data set are not given but will be used in the assessment marking. *Each of you will have **different data sets**, as such submission outputs will be different (except by chance).*

You will find these three datasets in: `/courses/comp5280/xyz` (or equivalent `\\raptor.kent.ac.uk\exports\courses\comp5280`) for COMP5280 students and `/courses/comp8250/xyz` or `\\raptor.kent.ac.uk\exports\courses\comp8250` for COMP8250 students on raptor where xyz is to be replaced by your own personal login. If you do not find a folder with your login or the folder does not have any files, then you need to inform me as soon as possible so that the problem can be rectified.

`kNN.java`, `kNN_test.java` and `kNN_GA.java` files can be found on Moodle page under the section Assignment 1.

---

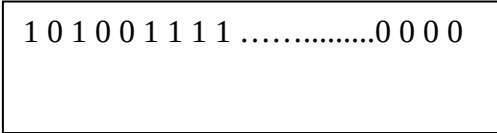
<sup>1</sup> Although I will be doing random checks to ensure there are no issues such as plagiarism.

<sup>2</sup> R. Palaniappan, P. Raveendran and S. Omatu, "VEP optimal channel selection using genetic algorithm for neural network classification of alcoholics," IEEE Transactions on Neural Networks, pp.486-491, vol. 13, issue 2, 2002.

## Submission

Your task for problem 1 is to obtain the optimal  $k$  (i.e. number of neighbours) in the nearest neighbour algorithm that maximises the classification performance. You will need to do the following:

- Using `kNN.java` and modifying as necessary to obtain the best/optimal  $k$  value (using the training and validation data sets). After obtaining this  $k$  value, include this value in the `kNN_test.java`. **Do not modify** any other line in `kNN_test.java`.
- I will compile and run this `kNN_test.java` which will generate the predicted labels in the `kNN_output.txt` file in the manner prescribed below. In the first line, there will be one line of 200 values in the file with either 0 or 1 representing the predicted class of the test data with a single space between the predicted labels. There is no need to include this `kNN_output.txt` file in the submission folder but you may wish to compile and run `kNN_test.java` to ensure that it runs correctly on raptor or to ensure that you have not made any inadvertent change. Using the predicted labels for the test data, I will obtain the classification accuracy using your best  $k$  value.
- You need to place your solution file (i.e. `kNN_test.java`) in: `/proj/co528c/ga/xyz` (or `\\raptor.kent.ac.uk\exports\proj\comp5280\ga\xyz`) for COMP5280 students and `/proj/co528m/ga/xyz` (`\\raptor.kent.ac.uk\exports\proj\comp8250\ga\xyz`) for COMP8250 students on raptor where `xyz` is to be replaced by your own personal login. Permissions have been set so that only `xyz` can access files in the directory `xyz`. You will lose write permission at 23:55 on the day of the deadline.



```
1 0 1 0 0 1 1 1 1 .....0 0 0 0
```

Figure 1: `kNN_output.txt` file

**Marking scheme** (5 marks) for this problem is as follows: Your test data output labels will be matched for accuracy computation. Five marks will be allocated if your accuracy is above 80% (which will be the case when an optimal  $k$  value is used). Accuracy below 80% (which will likely be the case for the default  $k=1$ ) will not get you any marks. If you do not submit `kNN_test.java` file, you will not get any marks for this problem. If `kNN_test.java` does not compile and run on raptor, you will not get any marks.

## Problem 2

The second problem is an extension of problem 1. In this problem, you will need to use GA to find the appropriate channel (features) to use in the classification that will improve the classification accuracy. There are some additional constraints to be met:

- Performance (accuracy)  $\geq 85\%$
- Maximum number of features to use  $\leq 40$
- Both these constraints must be met, otherwise you will get a mark of zero

## Submission

Your task for problem 2 is to obtain the best features to use. You will need to do the following:

- Use the code template `kNN_GA.java` and include additional codes (within the file) as necessary to implement selection, crossover and mutation operations over many generations. You should include all codes within this file. The fitness will be generated using kNN fitness (using training and validation datasets) and number of features, where the former needs to be maximised and the latter minimised. Once the optimal set of features are obtained, these will be included in another file, `kNN_GA_output.txt` in the manner prescribed below. In the first line, there will be one line of 61 values in the file with either 0 or 1 where 1 represents features from that are selected and 0 denotes features that are not used. Therefore, you need to ensure that you do not modify the `kNN_GA_output.txt` file generation code.
- You will need to ensure that `kNN_GA.java` runs correctly on raptor and within 5 minutes (it is best not to increase the population size or maximum number of generations for this reason). If the program does not compile or terminate within 5 minutes on raptor, I will manually terminate the program and no marks will be given.
- Using the obtained best features from `kNN_GA_output.txt`, I will obtain the classification accuracy using the test data.
- You need to place your `kNN_GA.java` file in: `/proj/co528c/ga/xyz` (or `\\raptor.kent.ac.uk\exports\proj\comp5280\ga\xyz`) for COMP5280 students and `/proj/co528m/ga/xyz` (`\\raptor.kent.ac.uk\exports\proj\comp8250\ga\xyz`) for COMP8250 students on raptor where xyz is to be replaced by your own personal login. Permissions have been set so that only xyz can access files in the directory xyz. You will lose write permission at 23:55 on the day of the deadline.



1 0 1 0 1 .....0 1 0 0

Figure 2: `kNN_GA_output.txt` file

**Marking scheme** (20 marks) for this problem is as follows: The selected features in the `kNN_GA_output.txt` file will be used to obtain prediction of test data classes and labels will be matched for accuracy computation. Note that although you will be finding the optimal features to use with validation data, if the features are indeed optimal for validation dataset, these will also be likely somewhat optimal for test data set.

Marks =  $(x - 85) * 10 + (61 - \text{cnt}) / 30 * 10$  where `cnt` is the number of features and `x` is the accuracy obtained by the machine running the **test** dataset but subject to accuracy  $\geq 85\%$  and feature count  $\leq 40$ . If `x`  $< 85\%$  or `cnt`  $> 40$ , then no mark will be awarded. Note that the mark will be capped at 20 for this problem, so `x`  $> 95\%$  or `cnt`  $< 31$  will not improve the mark any further.

## To check

If you have completed the tasks correctly, you must have these files in the root directory:

kNN\_test.java

kNN\_GA.java

If you compiled and tested the files by running them on raptor, then you could have these files in the root directory:

kNN.java

kNN\_test.java

kNN\_GA.java

kNN\_GA\_output.txt

kNN\_output.txt

test\_data.txt

train\_data.txt

train\_data\_label.txt

val\_data.txt

val\_data\_label.txt

and any necessary files generated during Java compilation. ***Please do not create any sub-folders or include additional files.***

To summarise, use kNN.java to get the optimal k value for kNN, include this k value in kNN\_test.java and submit kNN\_test.java, modify kNN\_GA.java to implement GA and submit kNN\_GA.java.

If you do not submit the required files, you will not get any marks.

Total marks for problems 1 and 2 is 25 marks.

## Deadline

The deadline for submission is **5 November 2021, 23.55 pm.**

Section 2.2.1.1 of Annex 9 of the Credit Framework states that, "Academic staff may not accept coursework submitted after the applicable deadline except in concessionary circumstances". For extensions/late coursework submission requests, refer to <https://moodle.kent.ac.uk/2021/course/view.php?id=612>

A Frequently Asked Questions document on Plagiarism and Collaboration is available at: [www.cs.kent.ac.uk/teaching/student/assessment/plagiarism.local](http://www.cs.kent.ac.uk/teaching/student/assessment/plagiarism.local). The work you submit must be your own, except where its original author is clearly referenced. Random checks will be run on submitted work in an effort to identify possible plagiarism, and take disciplinary action against anyone found to have committed plagiarism. When you use other peoples' material, you must clearly indicate the source of the material.