F21DL Data Mining and Machine Learning: Coursework Assignment 3
Handed Out: 04th November 2017
What must be submitted: A report of maximum 4 sides of A4 (5 sides of A4 for Level 11), in PDF format, and accompanying software.
To be 'Handed in': 15:00pm Monday, 20th of November 2017 - via Vision
Worth: 30% of the marks for the module**.**

The point: this coursework is designed to give you experience with, and hence more understanding of:

- Overfitting: finding a classifier that does very well on your training data doesn't mean it will do well on unseen (test) data.
- The relationship between overfitting and complexity of the classifier – the more degrees of freedom in your classifier, the more chances it has to overfit the training data.
- The relationship between overfitting and the size of the training set.
- Bespoke machine learning: you don't have to just use one of the standard types of classifier – the 'client' may specifically want a certain type of classifier (here, a ruleset that works in a certain way), and you can develop algorithms that try to find the best possible such classifier.

In this coursework you will work with a big 'emotion recognition' dataset, created by the research group of Pierre-Luc Carrier and Aaron Courville. Get it as http://www.macs.hw.ac.uk/~ek19/CW3-2017.zip
You will find two sets of csv files:

1. fer2017-training.csv and fer2017-testing.csv
2. fer2017-training-happy.csv and fer2017-testing-happy.csv

The training data set (of 28709 examples) and the testing data set (of 7178 examples) consist of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).
The second set of files makes only classification into happy and not happy classes.

What to do:

Before you start:

Choose the software in which to conduct the project. We strongly recommend all students to use Weka, it is a mature, well-developed tool designed specifically to facilitate mastery of machine-learning algorithms. In addition, it is supported by a comprehensive textbook: http://www.cs.waikato.ac.nz/ml/weka/book.html . Weka has a strong support for embedded Java programming, and you are welcome to use it in this assignment: it will allow you to automate many parts of this assignment. (See the chapter ``Embedded Machine learning in www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf). This will give you an experience of using Embedded Weka programming in projects involving Java. Alternatively, Weka command line interface may be embedded inside of Bash scripts, instead of Java.

Students wishing to complete the below tasks in other languages, such as R, Matlab, Python are welcome to do so, assuming they have prior knowledge of these languages.

In the below task spec, the assumption is made that the majority of the class uses Weka. Please adapt the below instructions accordingly if you use a different programming language.

---

1. Convert the above files into arff format and load them to Weka.
   Dealing with big data sets: in CW2, you were given several options how to deal with large data sets in Weka (increasing heap size for Weka GUI, using Weka command line with increased heap, wrapping Weka command line within scripts that automate the experiments, or just reducing the size of the data set using Weka methods of randomization and attribute selections). You will have to make one such decision for this coursework, too.
2. Create folders on your computer to store classifiers, screenshots and results of all your experiments, as explained below.

Your coursework will consist of two parts – in Part-1 you will work with Decision trees and in Part -2 – with Linear Classifiers and Neural Networks.
For each of the two parts, you will do the following:

3. Using the provided data sets, and Weka's facility for 10-fold cross validation, run the classifier, and note its accuracy for varying learning parameters provided by Weka. (Below you will find more instructions on those.) Record all your findings and explain them. Make sure you understand and can explain logically the meaning of the confusion matrix, as well as the information contained in the "Detailed Accuracy" field: TP Rate, FP Rate, Precision, Recall, F Measure, ROC Area.
4. Use Visualization tools to analyze and understand the results: Weka has comprehensive tools for visualization of, and manipulation with, Decision trees and Neural Networks.
5. Repeat steps 3 and 4, this time using testing data set instead of Weka's cross validation.
6. Make new training and testing sets, by moving 3000 of the instances in the testing set into the training set. Then, repeat steps 3 and 4.
7. Make new training and testing sets again, this time enlarging the training set with 6000 instances from the testing set, and again repeat steps 3 and 4.
8. Analyse your results from the point of view of the problem of classifier over-fitting.

NB: If you reduced the sizes of the training and testing data sets, then in steps 6 and 7, move 30% and 70% of the testing examples to the training set, respectively, instead of moving 3000 and 6000 instances.

Detailed technical instructions:

Part 1. Decision tree learning.

In this part, you are asked to explore the following three decision tree algorithms implemented in Weka

1. J48 Algorithm
2. User Classifier (This option allows you to construct decision trees semi-manually)
3. One other Decision tree algorithm.

You should compare their relative performance on the given data set. For this:

- Experiment with various decision tree parameters: binary splits or multiple branching, prunning, confidence threshold for pruning, and the minimal number of instances permissible per leaf.
- Experiment with their relative performance based on the output of confusion matrices as well as other metrics (TP Rate, FP Rate, Precision, Recall, F Measure, ROC Area). Note that different algorithms can perform differently on various metrics. Does it happen in your experiments? – Discuss.
- When working with User Classifier, you will learn to work with both Data and Tree Visualizers in Weka. Please reduce the number of attributes as in CW2 to prototype more efficiently in Visualizers.
- Record all the above results by going through the steps 3-8.

Part 2. Neural Networks.

In this part, you will work with the *MultilayerPerceptron* algorithm in Weka.

- Run *MultilayerPerceptron*. Experiment with various Neural Network parameters: add or remove nodes, layers and connections, vary the learning rate, epochs and momentum, and validation threshold.

- You will need to work with Weka's Neural Network Visualiser in order to perform some of the above tasks. You are allowed to use smaller data sets when working with the Visualiser.
- Experiment with relative performance of Neural Networks and changing parameters. Base your comparative study on the output of confusion matrices as well as other metrics (TP Rate, FP Rate, Precision, Recall, F Measure, ROC Area).
- Record all the above results by going through the steps 3-8.

Level 11 only (MSc students and MEng final year students):

9. Deep Learning and Deep Neural Networks have gained popularity recently. Do some research (using the www and the recommended textbook) to find out more about Deep Learning. Use algorithms and tools available in Weka or on-line. Write a one page essay comparing Neural Networks and Deep Neural Networks.

What to Submit

You will submit:

(a) All sources with the evidence of conducted experiments: data sets, scripts, tables comparing the accuracy, screenshots, etc. Give a web link to them (github, bitbucket, Dropbox, own webpage…).

(b) A report of maximum FOUR sides of A4 (11 pt font, margins 2cm on all sides) for Honours BSc students and FIVE sides of A4 (11 pt font, margins 2cm on all sides) for MSc students.

Using the results and screenshots you recorded when completing the steps 3-8, write five sections, respectively entitled:

1. "Variation in performance with size of the training and testing sets"
2. "Variation in performance with change in the learning paradigm (Decision trees versus Neural Nets)"
3. "Variation in performance with varying learning parameters in Decision Trees"
4. "Variation in performance with varying learning parameters in Neural Networks"
5. "Variation in performance according to different metrics (TP Rate, FP Rate, Precision, Recall, F Measure, ROC Area)"
6. (Level 11 students) ``Comparative analysis of Neural Networks and Deep Neural Networks"

In each of these sections you will speculate on the reasons that might underpin the performance variations that you see, considering general issues and also issues pertaining to this specific task.

You are recommended to represent all your results in one or two big tables – to which you will refer from these five specific sections.

Marking: see the detailed marking sheet on Vision.

Points possible: 100.

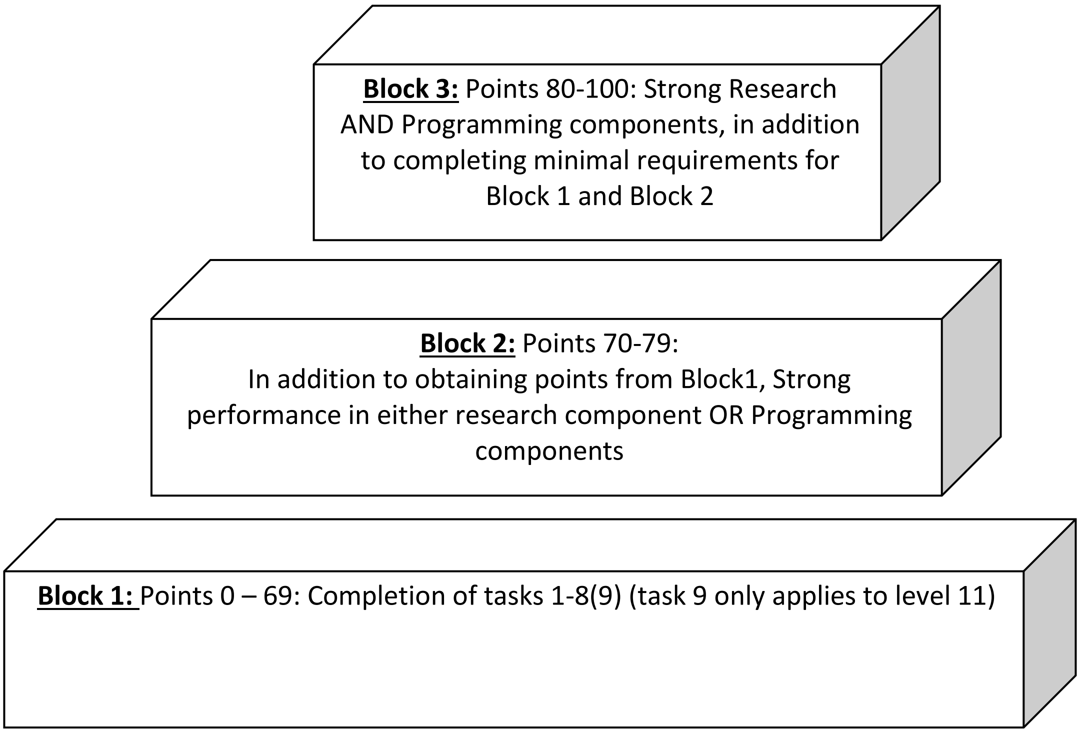
Level 10: Each Section is worth 20 points of the total 100 points.

Level 11: **Sections 1-5 are worth 17 points each, section 6 is worth 15 points.**

You will get up to 69 points (up to B1 grade) for completing the tasks 1-9 well and thoroughly (task 9 is for level 11 only) and giving a reasonable explanation of the obtained results.

In order to get an A grade (70 points and higher), you will need to do well in tasks 1-8(9) but in addition, you will need to show substantial skill in either research or programming:

- Research skills: The submission must show original thinking and give thorough, logical and technical description of the results that shows mastery of the tools and methods, and understanding of the underlying problems. The student should show an ability to ask his/her own research questions based on the CW material and successfully answer them.
- Programming skills: You will need to produce a sizeable piece of software produced to cover some tasks 1-8/9.
- The mark distribution will thus follow the below scheme:



Block 3: Points 80-100: Strong Research AND Programming components, in addition to completing minimal requirements for Block 1 and Block 2

Block 2: Points 70-79: In addition to obtaining points from Block1, Strong performance in either research component OR Programming components

Block 1: Points 0 – 69: Completion of tasks 1-8(9) (task 9 only applies to level 11)