F20/21DL Data Mining and Machine Learning: Coursework Assignment 2

Handed Out: 11th November 2019
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.
Submission deadline: 11:55pm Monday 2nd December 2019 -- via Vision
Worth: 25% of the marks for the module**.**

The point: this coursework is designed to give you experience with, and hence improve your 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 application may specifically require a certain type of classifier, and you can develop algorithms that find the best possible such classifier.

The data set:

The data set for the coursework is the same as in Coursework 1, please refer to Coursework 1 specification for general description and motivation. For this coursework, we additionally provide the testing data sets. They can be downloaded here: http://www.macs.hw.ac.uk/~ek19/data/CW2/. The naming convention is as follows:

1. The main training data set: as in coursework 1.
2. The main test data set:

- [x_**test_gr_smpl.csv**] contains test features (i.e. the images).

3. Class labels for the main test data set:

- **[y_test_smpl.csv]** contains labels for the test file, ranging from 0 to 9. As before, the labels stand for the following street signs:

0. **speed limit 60**
1. **speed limit 80**
2. **speed limit 80 lifted**
3. **right of way at crossing**
4. **right of way in general**
5. **give way**
6. **stop**
7. **no speed limit general**
8. **turn right down**
9. **turn left down**

- [y_**test_smpl_<label>.csv**] Test labels for one-vs-rest classification. Images that were originally in class <label> are marked by 0 and all other images -- by 1. For example, if <label> is 6, then all images in the train set displaying a stop sign are given the label 0, labels of all other images are set to 1.
- *Note 1: these ``one-vs-rest" data sets can be used as reserve data sets, for testing various hypotheses you may come up with in the coursework. Also they may give better accuracies, and thus may be handy for some experiments. Please use them to enrich your research hypotheses and experiments.*

What to do:

Before you start: Choose the software in which to conduct the project. We strongly recommend all students use Weka. Weka is a mature, well-developed tool designed to facilitate mastery of machine-learning algorithms. It is supported by a comprehensive textbook: http://www.cs.waikato.ac.nz/ml/weka/book.html . Weka supports embedded Java programming, and you are welcome to use embedded programming in this assignment as it will allow you to automate parts of this assignment. (See the chapter ``Embedded Machine learning in www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf). Alternatively, the Weka command line interface may be embedded inside Bash (shell) 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. **For students who already know Python, there will be tutorials and labs covering some of the Python libraries for machine learning.** Recommended implementation of the multiplayer perceptron is in the library *keras*[1].

In the below task specification, the assumption is made that you are using 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 CW1, 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 training data sets, and the 10-fold cross validation, run the classifier, and note its accuracy for varying learning parameters provided by Weka (or your other tool of choice). 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 training and testing data sets instead of the cross validation. That is, build the classifier using the training data set, and test the classifier using the test data set. Note the accuracy.
6. Make new training and testing sets, by moving 4000 of the instances from the original training set into the testing set. Then, repeat step 5.
7. Make new training and testing sets again, this time removing 9000 instances from the original training set, and placing them into the testing set again repeat step 5.
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 original training examples to the testing set (instead of moving 4000 and 9000 instances).

---

[1] https://keras.io/

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 of your choice (e.g. random forest).

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 to prototype more efficiently in Visualizers. What do can you learn about the data set and the machine learning task at hand by observing the decision tree structure?
- 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 a Linear classifier on the data set. This will be your base for comparison.
- 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. *[Research Question]* Think about your own research question and/or research problem that may be raised in relation to the given data set, and the topics of Decision Tree learning, Linear Classifiers and Neural Networks. Formulate this question/problem clearly, explain why it is of research value. The problem may be of engineering nature (e.g. how to improve automation or speed of the algorithms), or it may be of exploratory nature (e.g. something about finding interesting properties in data), -- the choice is yours.
10. *[Answer your research question]* Provide a full or preliminary/prototype solution to the problem or question that you have posed. Give logical and technical explanation why your solution is valid and useful.

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) My own research topic.

In each of these sections you will speculate on the reasons that might underpin the performance and the 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:
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(9-10) 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: a sizeable piece of software produced to cover some tasks.

Plagiarism
This project is assessed as **group work**. You must work within your group and not share work with other groups. Readings, web sources and any other material that you use from sources other than lecture material must be appropriately acknowledged and referenced. Plagiarism in any part of your report will result in referral to the disciplinary committee, which may lead to you losing all marks for this coursework and may have further implications on your degree. https://www.hw.ac.uk/students/studies/examinations/plagiarism.htm

Lateness penalties
Standard university rules and penalties for late coursework submission will apply to all coursework submissions. See the student handbook.

The mark distribution will thus follow the below scheme: