F20DL / F21DL Data Mining and Machine Learning: Coursework 1

Handed Out: Friday, 2 October 2020.
Work organisation: Group work, in groups of 4 students. Subscribe via: the **Assessment** page on Vision.
What must be submitted: A report of maximum 4 sides of A4 (five sides of A4 for Level 11), in PDF format, and accompanying software.
Submission deadline: 15:00pm GMT Monday 09 November 2020 -- via Vision
Worth: 25% of the marks for the module**.**

The purpose:

Data preparation and analysis, confusion matrices, correlation and feature selection are all important in real-world machine learning tasks. Data clustering and probabilistic data analysis are two core sets of methods in data mining and machine learning. So this coursework gives you experience with each of these things.

The data set:

The data set for the coursework is a sample from Stallkamp et al's *German Street Sign Recognition Benchmark*. Originally the data set consisted of 39,209 RGB-coloured train and 12,630 RGB-coloured test images of different sizes displaying 43 different types of German traffic signs. These images are not centred and are taken during different times of the day.

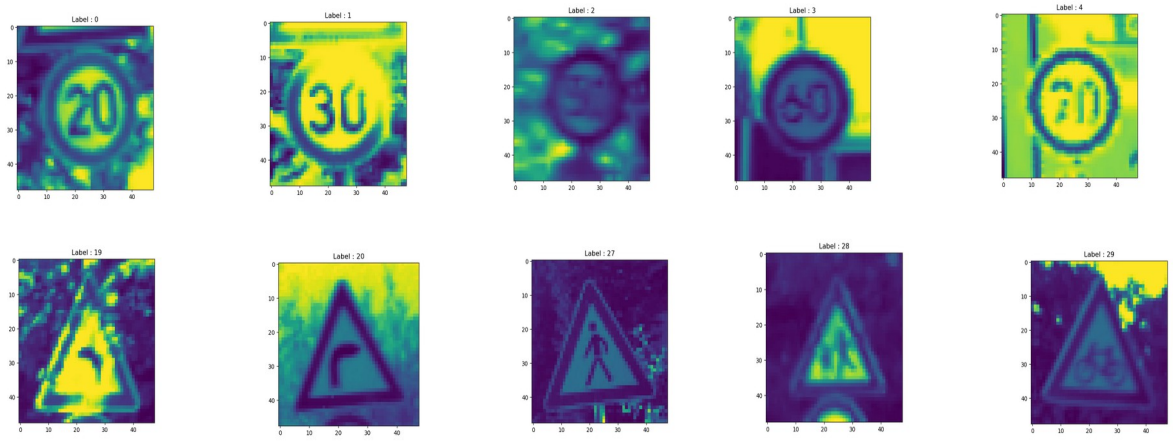
This data set is considered to be an important benchmark for Computer Vision and has close relation to the street sign recognition tasks that autonomous cars have to perform. And safe deployment of autonomous cars is the next big challenge that researchers and engineers face.

You will be working with a sample of this data set which consists of 10 classes and 9690 images. The images have been converted to grey-scale with pixel values ranging from 0 to 255 and were rescaled to a common size of 48*48 pixels. Hence, each row (= feature vector) in the data set has 2305 features and represents a single image in row-vector format (2304 features) plus its associated class label. We changed the class labels from the original dataset so the classes we use are now labelled from 0 to 9. Compensating the light conditions and position of the images is not necessary for the coursework and is left for the interested student to do.

Below, the class labels and their meanings are displayed:

Class label	Meaning
0	speed limit 20
1	speed limit 30
2	speed limit 50
3	speed limit 60
4	speed limit 70
5	left turn
6	right turn
7	beware pedestrian crossing
8	beware children
9	beware cycle route ahead

Below are examples of images of the street signs in this data set:



For this coursework, we provide 11 training data sets which can be downloaded here: http://www.macs.hw.ac.uk/~ek19/data/. The naming convention is as follows:

1. One entire sample:

[train_gr_smpl.arff] (for Weka) or **[x_train_gr_smpl.csv]** and **[y_train_gr_smpl.csv]** (for Python) contain training features and labels from the entire sample represented as row vectors. Class labels range from 0 to 9.

2. Ten one-vs-rest samples:

[train_gr_smpl_<label>.arff] (for Weka) or **[x_train_smpl_<label>.csv]** and **[y_train_smpl_<label>.csv]** (for Python) contain training features and labels for one-vs-rest classification. In each file, the images with class <label> have a 0 and all the other images a 1. For example, if the <label> is 6 (as in **train_gr_smpl_6.arff**) then all the images of right turn signs have a 0 as their label and all the other images have a 1.

Note: the csv files labelled with x_train contain the image data for each sign and the csv files labelled with y_train contain the corresponding class label for each sign in the same order. This is a standard for data representation in machine learning.

What to do:

Form or join a group in which you will work; discuss with the group your strategy for completing the course works: the workload split, the tools, the methods… Please use Vision **F21DL_2020-2021: Data Mining and Machine Learning (**subpage Assessment) to register your group or join an existing group.

Choose the software in which to conduct the project. Options are: Weka or Python (both supported by labs and tutorials).

After collecting the files as above, you will:

1. *[Data Randomisation]* Produce versions of the above files that have the instances in a randomised order.

2. *[Reducing the size, dealing with computational constraints]* The given files may be too big for standard settings of the Weka Explorer Interface: decide how you are going to deal with this:

- You may reduce the number of attributes, as taught during the course. Record and explain all choices made when you perform the reduction of attributes. A number of algorithms and options are available in Weka. See Sections 2.1 -2.3 in www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf
- Alternatively, you may use the full data set and the Weka graphical Knowledge Flow interface or the Command Line interface. See Section 5 of www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf) and manipulate the heap size (see https://weka.wikispaces.com/OutOfMemoryException).
- Either choice is acceptable as long as you can perform the next task.

3. *[Classification: Performance of the Naive Bayes algorithm on the given data set]* Run the Naive Bayes tool on the resulting version of **train_gr_smpl**. To be able to do this in Weka, you may need to apply several Weka "Filters". Explain the reason for choosing and using these filters. Once you can run the algorithm, record, compare and analyse the classifier's accuracy on different classes (as given by the confusion matrix).

4. *[Deeper analysis of the data: the data is split into 10 classes, search for important attributes for each class]* For each **train_smpl_<label>** file, record the first 10 pixels, in order of the absolute correlation value, for each street sign.

5. *[Improvement in classification, based on feature selection*] Using the information about the top correlating features obtained in item (4), transform the full data set **train_smpl** so as to keep the following attributes:
- Using only the top 5 pixels from each **train_smpl_<label>**.
- Using only the top 10 pixels from each **train_smpl_<label>**.
- Using only the top 20 pixels from each **train_smpl_<label>**.
- You will have three data sets, with approximately 50, 100 and 200 features (pixels) each. Repeat the experiment described in item (3) on these three data sets.

6. *[Making conclusions:]* What kind of information about this data set did you learn, as a result of the above experiments? You should ask questions such as: Which streets signs are harder to recognise? Which street signs are most easily confused? Which pixels are more reliable and which are less reliable in classification of street signs? What was the purpose of Tasks 4 and 5? What would happen if the data sets you used were not randomised? What happens when there is cross-correlation between pixels and classes? You will get more marks for more interesting and ``out of the box" questions and answers. Explain your conclusions logically and formally, using the material from the lecture notes and from your own reading to interpret the results that Weka produces.

7. [*Beyond Naïve Bayes: complex Bayesian Network Architectures*] Build two or three Bayes networks of more complex architecture for (a smaller version of) this data set, increasing the number of connections among the nodes. Construct one of them semi-manually (e.g use K2 algorithm and vary the maximum number of parents), and two others – using Weka's algorithms for learning Bayes net construction (e.g. use TAN or Hill Climbing algorithms). Run the experiments described in item 5 on these new Bayes network architectures. Record, compare and analyse the outputs, in the light of the previous conclusions about the given data.

8. [*Making conclusions*] What kind of new properties and dependencies in the data did you discover by means of using the complex Bayesian Network Architectures? Does it help, and how, to use Bayes nets that are more sophisticated than Naïve Bayes nets? (You may want to read Chapter 6.7, pages 266-270 and pages 451-454 of the Data Mining textbook by Witten et al. before you do these exercises or https://www.cs.waikato.ac.nz/~remco/weka.bn.pdf.)

9. [*Clustering, k-means*] Cluster the data sets **train_smpl**, **train_smpl_<label>** (apply required filters and/or attribute selections if needed), using the k-means algorithm:

- First try to work in a classical clustering scenario and assume that classes are not given. Research methods which allow you to visualise and analyse clusters (and the performance of the clustering algorithm on your data set).
- Note the accuracy of k-means relative to the given clusters.

10. [*Making conclusions*] What did your experiments tell you about the data set and the k-means clustering? Make comparison with classification results obtained in 4-7.

11. *[Beyond k-means: other clustering algorithms, tools for computation of optimal number of clusters]* Try different clustering algorithms (hard and soft). Try to vary the number of clusters manually and then use Weka's/Python's facilities to compute the optimal number of clusters. Explore various options that help to improve clustering results. Use visualisation tools for clustering to analyse the results.

12. [*Making conclusions]* Make conclusions on the obtained improvements to clustering results. Make sure you understand the workings and the output of different (hard and soft) clustering algorithms when clustering is completed. Test the accuracy of these clustering algorithms using classes given in this data set. Based on your experiments, explain all pros and cons of using different clustering algorithms on the given data set. Compare to the results of Bayesian classification on the same data set.

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

13. *[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 Bayesian learning and Clustering. 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.

14. *[Answer your research question]* Provide a full or preliminary/prototype solution to the problem or question that you have posed. Confirm your findings and conjectures by means of experiments, where appropriate. Give logical and technical explanation why your solution is valid and useful.

An Important note:
Before you start completing the above tasks, create folders on your computer to store software you produce, classifiers, Weka settings, screenshots and results of all your experiments. Archive these folders and a repository link within your report. As part of your coursework marking, you may be asked to re-run your experiments in the lab or show the trace of your work. Remember, this assignment is worth a quarter of your overall module mark! So please store all code and data safely in a way that will allow you to re-produce your results on request.

What to Submit:

(a) 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. Figures and illustrations do not count as part of the page limit. Only one report per group should be submitted, as multiple submissions will trigger the plagiarism checker.
(b) All evidence of conducted experiments: data sets, scripts, tables comparing the accuracy, screenshots, etc. Supply a link to your HW web space, Github or Google drive.
(c) Declaration of what parts of the coursework each group member contributed. Data preparation, programming, analysis, report writing (and generation of figures and illustrations for the report) all count as contribution. You are encouraged to solve more complex research tasks

collaboratively: you will learn more and do a better job if you discuss your progress and your actions regularly.

Marking: See the full marking Rubric on Vision. Maximum points possible: 100.

You will get up to 69 points (up to B1 grade) for completing the Tasks 1-6, 9-10 (and Task 13 for **Level 11**) well and thoroughly.

In order to get an A grade (70 points and higher), you will need to first score 69 points as described above, and in addition, you will need to show substantial skill in either research or programming:

- Research skills: Higher marks will be assigned to submissions that 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. Tasks 7, 8, 11, 12, 14 are formulated as research tasks.
- Programming skills: You will need to produce a sizeable piece of software produced to automate some tasks. You will need to show competence in visualising and presenting your results.

The report should: (a) give logical and technical explanations of the decisions you made; (b) use graphs, charts and illustrations effectively; (c) explain your programming/experiment efforts convincingly; (d) use the space wisely; make sure you label axes, charts and graphs clearly.

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.