F21DL Data Mining and Machine Learning: Coursework 2

Handed Out: Monday 16th October 2017.
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.
To be 'Handed in': 15:00pm Monday 6th November 2017 -- via Vision
Worth: 30% of the marks for the module**.**

The point: confusion matrices, correlation and feature selection, data clustering and probabilistic data analysis are all important in data mining and machine learning. So this coursework gives you experience with each of these things.

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 from
Vision -> Assessment -> Coursework 2 -> fer2017.zip

The data set (of 35887 examples) consists 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).

In the zip file, the main data set is called fer2017.csv. It is supplemented by 7 csv files, in which each othe above 7 emotions is tested separately:

- fer2017angry.csv,
- fer2017disgust.csv
- etc.

You will practice using Naïve Bayes and Clustering algorithms on these data sets.
In all these datasets, the first field corresponds to the class field: categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The remaining fields correspond to a 48x48 array image, so that each field indicates the general amount of ink in a specific area of the image. In fer2017.csv, the first field is the emotion – i.e. if it is '6', then the face shown on the image has neutral emotion. In fer2017*EmotionX*.csv, however, the class field is either 0 or 1. It is '1' if the image shows the emotion 'X', and it is 0 otherwise. For example, in fer2017angry.csv, if the last field is '1', then that instance corresponds to an image with an angry face; if the last field is '0', then that instance might correspond to a face with any other emotion.

What to do

Everyone:
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.

After collecting the files as above, you will:

1. *[Data Conversion]* Convert all csv files into arff format suitable for Weka. You should have a Weka data set with 2305 attributes as a result. So, some suitable data set pre-processing will be needed before you load the csv file to Weka.
2. *[Data Randomisation]* Produce versions of these files that have the instances in a randomised order.
3. *[Reducing the size, dealing with computational constraints]* The given files may be too big for standard settings of GUI Weka: make a decision how you are going to go about this obstacle in this assignment. You may reduce the number of attributes, as you learned in Part 1 of 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 command-line interface of Weka (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). Any choice is acceptable as long as you can perform the next task.
4. *[Classification: Performance of Naïve Bayes Algorithm on the given data set]* Run the naïve Bayes tool in Weka on the resulting version of fer2017.arff. To be able to do this, 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 Weka Summary and the confusion matrix).
5. *[Deeper analysis of the data: its split into 7 classes, search for important attributes for each class]* For each fer2017*EmotionX*.arff file,
   5.1 Run the naïve Bayes tool in Weka u*sing only the first 50% of randomised instances*, note the accuracy. Compare with your results in item (4).
   5.2 Using the Weka facility "Select Attributes" for each of these 7 files, record the first 10 fields, in order of the absolute correlation value, for each emotion.
6. *[Attempt at better Bayesian classification, based on information from 5*] Using the information about the top correlating features obtained in item (5), transform the full data set fer2017.arff in such a way as to keep the following attributes:

6.1. Using only the top 2 non-class fields from each fer2017*EmotionX*.arff.
6.2. Using only the top 5 non-class fields from each fer2017*EmotionX*.arff.
6.3. Using only the top 10 non-class fields from each fer2017*EmotionX*.arff.
6.4. You will have three data sets, with 14, 35 and 70 non-class attributes respectively. Repeat experiment described in item (4) on these three data sets.

7. *[Make conclusions:]* What kind of information about this data set you learned, as a result of the above experiments? You may want to ask questions such as: Which emotion is it harder to recognise? Which attributes (fields) are more and which are less reliable in classification of emotions? What was the purpose of the task of items (5-6)? What would happen if the data sets you used in items 4-6 were not randomised? You will get higher marks for more interesting and ``out of the box" questions and answers. You should explain your

conclusions logically and formally, using the material from the lecture notes to interpret results that Weka produces.

8. *[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).
   8.1 Run the experiments described in items (4-6) on these new Bayes network architectures. Record, compare and analyse the outputs, in the light of the previously conclusions about the given data. What kind of dependencies in the data did you discover? 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 fer2017.arff, fer2017*EmotionX*.arff (apply required filters and/or attribute selections if needed), using the k-means algorithm:

9.1. first excluding the class attribute (use *classes to clusters* evaluation to achieve this). This will emulate the situation when the learning of digit shapes is performed in unsupervised manner.

9.2. then including the class attribute. This will emulate the general data analysis scenario.

9.3. Make conclusions about the results, compare with classification results obtained in items (4-8).

10. *[Beyond k-means, tools for computation of optimal number of clusters]* Try different clustering algorithms. Try also to vary the number of clusters manually and then use Weka's facilities to compute the optimal number of clusters. Explore various options in Weka that help to improve clustering results. Use the visualisation tool for clustering to analyse the results. Make conclusions on the obtained improvements to clustering results.
11. *[Conclusions]* Make sure you understand the various details of Weka's output for different (hard and soft) clustering algorithms when clustering is completed. Use Weka's facilities to test the precision of clustering on this data set. Using your work with Weka as a source, 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):

12. *[Research Question]* Often it is useful or necessary to find a value for the correlation between a numeric field and a categorical (nominal) field, or between two categorical (nominal) fields. Do some research (using the www) to find out how it can be done.

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 submit via Vision.

As part of your coursework marking, you may be asked to re-run all your experiments in the lab. So please store all of this data safely in a way that will allow you to re-produce your results on request.

What to Submit

You will submit:

(a) A zip file with all evidence of conducted experiments: data sets, scripts, tables comparing the accuracy, screenshots, etc.

(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, containing the following:

Everyone:

1. HOW: up to a half page describing how you did steps 1, 2, and 3.
2. RESULTS: up to two and a half pages showing and discussing the results from step 4 to step 11 (I expect these to include a discussion or display of the selected settings, confusion matrices for Bayesian classification and various output parameters for clustering).
3. Figures and screenshots should take about 1 page.

Level 11 only:

In addition, about a page with the title "Calculating correlation values for categorical data", explaining how this can be done for pairs of fields when either one or two of the pair is non-numeric.

Marking: see also the tentative marking sheet on Vision.
Maximum points possible: 100.

You will get up to 69 points (up to B1 grade) for completing the tasks 1-7 and 9,11, 12 well and thoroughly (task 12 is for level 11 only).
In order to get an A grade (70 points and higher), you will need to do well in tasks 1-7 and 9,11, 12 but in addition, you will need to show substantial skill in either research or programming:

- Research skills: You will need to complete tasks 8 and 10 well, and explain the results convincingly. 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.
- Programming skills: You will need to produce a sizeable piece of software produced to cover at least tasks 1-7 and 9,11. You will get even higher marks if your software can cover all tasks 1-11 or 12 (for level 11 only).

- The mark distribution will thus follow the below scheme: