Introductory Applied Machine Learning: Assignment 1

School of Informatics, University of Edinburgh

Instructors: Victor Lavrenko and Nigel Goddard
Assignment prepared by Sean Moran, revised by Boris Mitrovic

For due date and time, see the course web page.
Hard copy **and** electronic submission required. The time of the deadline will be strictly enforced.
Ensure that your name does not appear on the document AND that your matriculation number does appear.

Remember that plagiarism is a university offence. Please read the policy at
http://www.inf.ed.ac.uk/teaching/plagiarism.html .

Marking Breakdown

70-100% results/answer correct plus extra achievement at understanding or analysis of results. Clear explanations, evidence of creative or deeper thought will contribute to a higher grade.

60-69% results/answer correct or nearly correct and well explained.

50-59% results/answer in right direction but significant errors.

40-49% some evidence that the student has gained some understanding, but not answered the questions properly.

0-39% serious error or slack work.

Mechanics

You should produce a word processed report in answer to this assignment (e.g. with LaTeX).

- **postscript or pdf** formats are acceptable for the report, other formats are not.
- you need to submit this report as a hard copy to the ITO **and** electronically as described below.

For the electronic submission place your report in a directory called `iamlans` and submit this using the `submit` command on a DICE machine. The format is

```
submit iaml 1 iamlans
```

You can check the status of your submissions with the `show_submissions` command.

NOTE: Your electronic submission will **not** count if you do not submit a hard copy of your report to the ITO.

Late submissions: The policy stated in the School of Informatics MSc Degree Guide is that normally you will not be allowed to submit coursework late. See
`http://www.inf.ed.ac.uk/teaching/years/msc/courseguide10.html#exam` for exceptions to this, e.g. in case of serious medical illness or serious personal problems.

Collaboration: You may discuss the assignment with your colleagues, provided that the writing that you submit is entirely your own. That is, you should NOT borrow actual text from other students. We ask that you list on the assignment sheet a list of the people who you've had discussions with (if any).

Important Instructions

(a) In the following questions you are asked to run experiments using WEKA. The WEKA version installed on DICE is **Version 3.6.2**. If you are working on a machine other than DICE (e.g. your laptop), please make sure that you download and install the **same** version. This is important as your results need to be reproducible on DICE.

(b) In many cases, the WEKA *Explorer* allows you to modify the random seed that will be used. Just using the default seed is fine. If you do change the seed you need to report the seed you have chosen.

(c) You may find that WEKA crashes with an out-of-memory exception. If this should occur, refer to the instructions in IAML lab 1 in order to run WEKA with a larger memory allocation.

(d) The .arff files that you will be using are available from the IAML website.

(e) **IMPORTANT:** Keep your answers brief and concise. NOTE: you may **lose points** for a report longer than **900 words**.

Description of the dataset

This assignment is based on the 20 Newsgroups Dataset[1]. This dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware, comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale, soc.religion.christian).

There are three versions of the 20 Newsgroups Dataset. In this assignment we will use the *bydate* matlab version in which documents are sorted by date into training (60%) and test (40%) sets, newsgroup-identifying headers are dropped and duplicates are removed. This collection comprises roughly 61,000 different words, which results in a bag-of-words representation with frequency counts. More specifically, each document is represented by a 61,000 dimensional vector that contains the counts for each of the 61,000 different words present in the respective document.

To save you time and to make the problem manageable with limited computational resources, we preprocessed the original dataset. We will use documents from only 5 out of the 20 newsgroups, which results in a 5-class problem. More specifically the 5 classes correspond to the following newsgroups 1:*alt.atheism*, 2:*comp.sys.ibm.pc.hardware*, 3:*comp.sys.mac.hardware*, 4:*rec.sport.baseball* and 5:*rec.sport.hockey*. However, note here that classes 2-3 and 4-5 are rather closely related. Additionally, we computed the mutual information of each word with the class attribute and selected the 520 words out of 61,000 that had highest mutual information. Therefore, our dataset is a $N \times 520$ dimensional matrix, where N is the number of documents. For very sophisticated technical reasons 1 was added to all the word counts in part A.

The resulting representation is much more compact and can be used directly to perform our experiments in WEKA.

1 Exploration of the dataset [65%]

Load the dataset `train_20news_partA.arff`. Your first task is to get a feel for the data that you will be dealing with in the rest of the assignment. For now, we will use only the training set and we will use cross validation (CV) on this dataset to evaluate the classifiers. In order to reduce computation time you should use 5-fold cross validation unless explicitly specified otherwise (the default setting is 10).

(a) To explore the dataset use the visualization functions in WEKA explorer. Note that WEKA allows you to see details of an individual data point: Open the dialog that shows the scatter plot for two attributes and

[1] *http://people.csail.mit.edu/jrennie/20Newsgroups/*

click on the respective data point. Another useful feature in this dialog is the "Jitter" slider bar: Jittering the data points is helpful if many data points lie very close together or on top of each other in the scatter plot. Write down any important observations about the data.

(b) Next, train and evaluate different classifiers on the dataset using 5-fold CV. Try a Decision tree (`J48`) and a Naive Bayes (`NaiveBayes`) classifier. For now, use the default settings for the classifiers. We are not interested in optimizing their parameters, we just want to get a first idea of the dataset. Compare the results (percent correct, PC) for the two different classifiers. Relate the classification performance to your observations in part (a), explaining why the two classifiers perform so differently. What is a reasonable baseline against which to compare the classification performance? *Hint:* What is the simplest classifier you can think of and what would its performance be on this dataset? Keep the result buffer for the Naive Bayes classifier, you will need it in question (d).

(c) Based on what you found out about the data in the previous questions, clean up the training set as you see fit. Briefly say what you have done and more importantly explain why. Please be sure to include the WEKA functionality you have used along with the relevant settings of that functionality.

(d) Retrain and evaluate the classifiers that you have been using in (b) using 5-fold CV. What is their performance on the modified dataset? Compare the parameters of the model learned by the `NaiveBayes` classifier for the two datasets. Is there an important difference?

2 Feature Selection [35%]

For this part of the assignment we supply you with another version of the 20 Newsgroups dataset `train_20news_partB.arff`. **Important:** *Please ensure that you are using this version of the dataset when answering the questions that follow. No marks will be awarded if you use the incorrect dataset.*

Feature selection and feature engineering are important aspects of machine learning in practice. In the following section we want to assess the usefulness of the features and the impact of feature engineering on the classification task. As in the previous section, all experiments should be performed on the training data using 5 fold CV and the default options.

(a) First, assess the "usefulness" of the 520 features using the attribute evaluator `InfoGainAttributeEval` (in the *Select Attributes* tab). How does this "attribute evaluator" work? Give a short description and interpretation of the results. Go to the *Visualize* tab and look at the plots of the attributes against the class variable for the 5 highest ranked attributes and the 5 lowest ranked ones. In one sentence explain what you notice. Remove the 20 attributes that are least useful from the dataset. Save this reduced dataset as you will need it in Part (b).

(b) Retrain and evaluate the Naive Bayes (`NB`) classifier on the reduced dataset from Part (a) i.e. the dataset with the 20 least useful attributes removed. Report its performance and compare it to the performance on the full dataset (that is, the dataset with all 520 attributes). What does this suggest for the 20 attributes we removed?