Introductory Applied Machine Learning: Assignment 1

School of Informatics, University of Edinburgh

Instructor: Nigel Goddard
Assignment prepared by Sean Moran, revised by Boris Mitrovic, revised by Nigel Goddard

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 **1350 words**.

1 Part I

Description of the dataset

This Part 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.

1.1 Exploration of the dataset [30%]

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).

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

(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 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?

(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.

1.2 Feature Selection [25%]

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?

2 Part II

Description of the dataset

This Part is based on the automobile pricing dataset. Our goal will be to predict the price of automobiles based on various attributes. This data set consists of three types of entities: (a) the specification of an automobile in terms of various characteristics, (b) its assigned insurance risk rating, (c) its normalized losses in use as compared to other cars. The second rating corresponds to the degree to which the auto is more risky than its price indicates. Cars are initially assigned a risk factor symbol associated with its price. Then, if it is more risky (or less), this symbol is adjusted by moving it up (or down) the scale. Actuaries call this process ”symboling”. A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe. The third factor is the relative average loss payment per insured vehicle year. This value is normalized for all autos within a particular size classification (two-door small, station wagons, sports/speciality, etc...), and represents the average loss per car per year.

To save you time and to make the problem manageable with limited computational resources, we preprocessed the original dataset. We removed any instances that had one or more missing values and randomized the data set. The resulting representation is much more compact and can be used directly to perform our experiments in WEKA.

2.1 Simple Linear Regression [15%]

We will begin by studying a simple Linear Regression model. Such a model will consider the relationship between a dependent (response) variable and only one independent (explanatory) variable. When applying machine learning in practice it can be prudent to start out simple in order to get a feeling for the dataset and for any potential difficulties that might warrant a more sophisticated model. In this Section we will consider one independent variable `engine-power` against the dependent variable `price`.

Important: *In order to reduce computation time you should use 5-fold cross validation unless explicitly specified otherwise (the default setting is 10).*

(a) Load the dataset `train_auto_partA.arff`. Have a look at the scatter plot of `price` against `engine-power` under the WEKA *Visualize* tab. Do you think that `engine-power` alone is sufficient for predicting the `price`? Please explain your answer in 2-3 sentences. Look at the distribution of the car prices. How would you preprocess it to improve the performance of linear regression? Don't do it at this stage, but instead in one sentence explain why you would do what you suggested.

(b) Now we will build a `SimpleLinearRegression` model. Under the *Classify* tab, choose *Functions > SimpleLinearRegression*. Keep the default settings for the model. Train the model and observe the results. Examine the learnt Linear Regression model in the WEKA results buffer. What happens to the `price` as one more unit of `engine-power` is added? By examining the magnitude of the regression coefficient is it possible to tell whether or not `engine-power` is an important influential variable on `price`? Explain your answer in 1-2 sentences.

(c) Record the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Correlation Coefficient (CC) from the `SimpleLinearRegression` model you build in (b). What do these metrics intend to measure? Relate the values of CC, MAE and RMSE to the observations you made in question (a).

2.2 Multivariate Linear Regression [30%]

In this Section we will fit a Multivariate Linear Regression model (`LinearRegression`) to the dataset. In contrast to Question 2.1, we will now train a model with multiple explanatory variables and ascertain how they affect our ability to predict the retail price of a car. One of our foremost concerns will be to determine exactly which attributes to include in the model and which may be left out.

Important: *In order to reduce computation time you should use 5-fold cross validation unless explicitly specified otherwise (the default setting is 10). When training a `LinearRegression` model be sure to set `attributeSelectionMethod` to `No attribute selection` and the `eliminateColinearAttributes` option to `False`. Leave all other parameters on default unless otherwise indicated in the questions that follow..*

Important: *When performing various transformations to the data you must ensure that the "class" variable is set to `price` in the WEKA drop down list (this drop down list is located sandwiched between `More Options` and the `Start` button in the `Classify` tab) before running your model. If you do not do this your results will be vastly optimistic and entirely incorrect.*

(a) Load the dataset `train_auto_partB_numeric.arff` into WEKA. This version of the dataset contains the numeric attributes only. Go to the Visualize tab and examine whether or not any of the other attributes are particularly good at predicting the price. Can you find any? Do any attributes appear useless at predicting the price? Do any attributes exhibit significant correlations? As you answer these questions list two attributes for each question but do not modify the dataset at this stage. Of the attributes you listed, which ones could you safely remove?

(b) We will now make a first attempt at building a Multivariate `LinearRegression` model using the numeric attributes only. Under the *Classify* tab, choose *Functions > LinearRegression*. Now run the regression and note down the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Correlation Coefficient (CC). Comment on each in comparison to what you have obtained for the *SimpleLinearRegression* model in 1(c).

(c) Examine the histogram for the `engine-size` attribute. Is the distribution expected to cause a problem for regression? Explain your answer in 2-3 sentences. Transform this attribute using an appropriate simple technique from the lecture 3. Re-build a `LinearRegression` model on the transformed dataset and report the RMSE, MAE and CC metrics. How has the performance of your model changed? In 1-2 sents explain why.

(d) Reload the `train_auto_partB_numeric.arff` dataset. Regression performance can occasionally be enhanced by adding quadratic (interaction) terms to the model. For example, multiplying the attributes `Width` and `Height` together (`Width`*`Height`) would constitute an interaction term. What does adding interaction terms attempt to capture? We will experiment with interaction terms that involve `engine_size` and one other attribute. Only multiplication should be used as the means of combination. Add various interaction terms of this type and report the results. Can you find any interaction terms that improved the regression performance of your model? *Hint:* The interaction terms can be added by using the `AddExpression` filter.