

COMP37332

WEKA: Classification tutorial¹

Task A: decision trees

In the first part, you will perform experiments using Weka's classification method J48 to build a decision tree as a model for the data. J48 is Weka's version of C4.5, and C4.5 is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on. Recall that the ID3 algorithm builds a decision tree given a set of non-categorical attribute.

A1. In order to believe any predictive model, the accuracy of the model must be estimated. Explain what is the accuracy measurement of a classifier? Find out and explain what 10-fold cross validation is.

A2. Download and load the 'labour' data set (labor.arff) that contains data of acceptable and unacceptable labour contracts. Examine the data and attributes and make sure you understand their meaning.

A3. Change to the Classify Panel. In the classifier tree, select *trees* and then *J48*. Run it with all options set to default. Make sure that the class attribute is selected as the classification label, and that you have selected cross-validation. Right click on the experiment (left panel – the Result list) and select the *Visualize tree* option. Examine options when right-clicking the tree visualisation panel (*Fit to Screen, Auto Scale, etc*). Analyse the classifier output, in particular the resulting accuracy (correctly and incorrectly classified instances). Analyse the detailed accuracy by classes – what are TP and FP? Make sure you understand all the data reported in the classifier output pane (e.g. confusion matrix)? Which class is easier to predict?

A4. Run other decision tree algorithm (e.g. ADTree) on the same data. Compare the results (both the generated tree and the accuracy). Why ID3 does not give any result?

A5. Repeat the classification using the vacation attribute as the classification label. Are these results acceptable?

A6. A baseline model is one that can be used to evaluate the success of your target model, in this case a DT model. Baseline models are typically simple and inaccurate, but occasionally data is so simple to describe that attempting to use a complex model may result in worse behaviour than a simple model. For this exercise, use ZeroR and OneR as baseline models. The ZeroR model simply classifies every data item in the same class. The OneR model seeks to generate classification rules using a single attribute only. Run these two models (select them under *rules*) and compare the performance to the decision tree models you generated previously.

¹ Tailored after: *Labs 2&3: Classification with Decision Trees and Other Classification Techniques* (MSCS 228: Data Mining) by C. A. Struble, Marquette University and *Practical Data Mining Tutorials 2 and 3* (COMP-321B) by Schmidberger and Hall, University of Waikato.

A7. Repeat the experiments with the 'glass' data set (glass.arff) that contains data of 6 types of glass and comes from the USA Forensic Science Service. The glass is described by its oxide content (i.e. Na, Fe, K, etc). How many classes do we have in this case? Analyse the results.

A8. Using the generated decision tree and the following instance as test instance, what value will this instance be classified as:

RI: 1.52127, Na: 14.32, MG: 3.9, Al: 0.83, Si: 71.5, K: 0.0,
CA: 9.49, Ba: 0.0, Fe: 0.0

The class value of this instance is 'tableware' - is the classification outcome correct?

A9. Load the data set *glass-minusatt.arff*. The data in this data set is the same as in the 'glass' data set but with some of the attributes removed. Explore the data and run J48 with all options set to default. Compare the results.

A10: Create a data set *glass-withnoise.arff* by selecting randomly 10% of the instances in the 'glass' data and changing their class attribute. This means that the data set now contains 10% class noise. Run J48 with all options set to default and compare the results.

A11. Which test had the best results? With these results, what do you conclude about the performance of the decision tree classifier J48? What about noisy data and irrelevant attributes?

Task B: SVM-based classification

In the second part of this tutorial, you will explore SVM classification. The SVM implementation in Weka is called SMO. It can be found in the Weka Explorer GUI, under the *functions* category.

B1. Load the 'glass' data set (glass.arff) again and run SMO with the default values. Compare the results (accuracy, confusion matrix) to the DT model. Repeat the test using the 'Use training set' option for testing, and compare the results. Examine the options that are available when right-clicking on the experiment in the Result list.

B2. Experiment with options 'c' and 'exponent'. 'c' is the complexity value and 'exponent' the exponent of the dividing hyperplane. The default of exponent is 1.0 which means that the dividing hyper-plane is linear. Click on the box with parameters and change the option 'c' to 20.0 and run SMO again with test option again set to 'Use training set'. Run SMO again but change the 'exponent' option (an option of the kernel 'PolyKernel') to 2. Compare the resulting accuracy and time needed to build the models.

Repeat the experiment with data in vehicle.arff. This data set contains examples of vehicle silhouettes. The purpose of the data set is to classify one of four types of vehicles. The vehicle may be viewed from one of many different angles.