Classification with WEKA

Data mining lab 2

WEKA inputs

The file format for WEKA is .**ARFF** file format.
A-attribute
R-relation
F-file
F-format

An .arff file contains the following tags
Example of the ARFF file format:
% name of the relation
@relation fruits

%list of the attributes, for example
@attribute size real
@attribute numberOfLegs numeric
@attribute color {red, blue, green}
@attribute description string
@attribute dateOfExperiment date %in format yyyy-MM-dd-THH:mm:ss (2004-04-03T12:00:00)

Then follows a list of comma-separated data instances.
Missing values are represented as ? in .arff data format.
@data

Explore .arff format

1.To understand the .arff format open file ***zoo.arff*** in the text editor.

2.The WEKA explorer interface is launched automatically when you double-click on .arff file. The file can be also chosen after starting WEKA Explorer from the Open menu. This is done while in the *Preprocess* tab.

3.In the Preprocess tab we also can do, for example, the following tasks:

•Remove attributes from the learning model (what attribute you think should be removed from the list of animals?)

•Perform data filtering operations: replace missing values, discretize numeric values etc.

Naive Bayes with WEKA

1. Launch WEKA

2. Choose Explorer interface

3. Open file wild_animals.arff

4. In Classify tab choose weka->classifiers->bayes->naiveBayes

5. The instances to be classified should be supplied in .arff format. Create a new .arff file, similar to wild_animals.arff, with 1 instance (row): hairy,red,large,soft,safe (for example)

6. Use test option: supplied test set. From file menu select your file with an instance to be classified.

7. Click Run

8. The results appear in the window on the left. What is the prediction produced by WEKA? Is it the same as we calculated by hand? If no, explain why.

Classification rules in WEKA

You may have to apply filters to your data, if they contain numeric values.

Since the ***PRISM*** algorithm can deal only with nominal attributes, to build a set of rules from data with numeric attributes apply Discretize filter.

1. Load file ***wild_animals.arff*** into WEKA explorer by double-clicking on it.

2. In the Classify panel select ***PRISM*** algorithm from tree->rules

3. Run the algorithm and compare 2 first rules with the hand results on the same dataset.

4. If the results are not the same as obtained by the non-machine deduction, please give an explanation.

Decision trees with WEKA

In order to compare your results of the tree induction for wild animals data with the WEKA results, run decision tree classifier on file ***wild_animals.arff.***

In order to put the resulting tree into a report, choose the tree (without evaluation) and copy it into your assignment file.

If the first split does not correspond to the split calculated by hand, please give an explanation.

Decision trees with WEKA on larger dataset - I

In this part of the lab we build decision tree for a bigger dataset: we train computer to learn the concept of different animal genera (mammals, fish, insects and so on).

Our input is file ***zoo.arff***

1. Open file zoo.arff in WEKA by double-clicking on it.
2. Remove unnecessary attributes by choosing them and pressing remove (how many attributes did you remove?)
3. Select Classifier tab, choose classifiers->tree->J48 (implementation of C4.5 decision tree algorithm). We will use 66% of animals to train the computer and 34% to evaluate the classifier. For this choose percentage split 66% option.
4. Build decision tree by clicking on run button. The result appears on the left and as the line in the history list. To see the tree, right-click on the line in the history list and choose visualize tree. In the tree window right-click and adjust the size of the tree using menu options.
5. In the results panel below the tree itself you see the estimation of the tree predictive performance.

Decision trees with WEKA on larger dataset - I

6. Put results of the experiment into a table with the following format: ***experiments.rtf***
7. Record values of the incorrectly classified instances (%), false positives and false negatives
8. Repeat experiment with ***Use training set*** option. The evaluation is performed on the training set itself, it is highly optimistic and represents an upper bound of the performance you can achieve with this model. Record your results.
9. Repeat experiment with ***10 folds cross-validation*** (the set is divided into 10 parts, then 9 parts are used for training and 1 for testing. The process is repeated 10 times and averaged). Record results.
10. Repeat experiment with ***5 folds cross-validation***. Record results.
11. From the results draw the conclusion of the ability of J48 classifier to extract concept from a given dataset. What is the predictive performance of the model (on the scale : good-bad) and whether the performance depends on the random selection of the subset of the training data?