WEKA tutorial exercises

These tutorial exercises introduce WEKA and ask you to try out several machine learning, visualization, and preprocessing methods using a wide variety of datasets:

- **Learners**: decision tree learner (J48), instance-based learner (IBk), Naïve Bayes (NB), Naïve Bayes Multinomial (NBM), support vector machine (SMO), association rule learner (Apriori)
- **Meta-learners**: filtered classifier, attribute selected classifiers (CfsSubsetEval and WrapperSubsetEval)
- **Visualization**: visualize datasets, decision trees, decision boundaries, classification errors
- **Preprocessing**: remove attributes and instances, use supervised and unsupervised discretization, select features, convert strings to word vectors
- **Testing**: on training set, on supplied test set, using cross-validation, using TP and FP rates, ROC area, confidence and support of association rules
- **Datasets**: *weather.nominal*, *iris*, *glass* (with variants), *vehicle* (with variants), *kr-vs-kp*, *waveform-5000*, *generated*, *sick*, *vote*, *mushroom*, *letter*, *ReutersCorn-Train* and *ReutersGrain-Train*, *supermarket*

Tutorial 1: Introduction to the WEKA Explorer

Set up your environment and start the Explorer

Look at the Preprocess, Classify, and Visualize panels

In Preprocess:

- load a dataset (*weather.nominal*) and look at it
- use the Data Set Editor
- apply a filter (to remove attributes and instances).

In Visualize:

- load a dataset (*iris*) and visualize it
- examine instance info
- (note discrepancy in numbering between instance info and dataset viewer)
- select instances and rectangles; save the new dataset to a file.

In Classify:

- load a dataset (*weather.nominal*) and classify it with the J48 decision tree learner (test on training set)
- examine the tree in the Classifier output panel
- visualize the tree (by right-clicking the entry in the result list)
- interpret classification accuracy and confusion matrix
- test the classifier on a supplied test set
- visualize classifier errors (by right-clicking the entry in the result list)

Answers to this tutorial are given.

Tutorial 2: Nearest neighbour learning and decision trees

Introduce the *glass* dataset, plus variants *glass-minusatt*, *glass-withnoise*, *glass-mini-normalized*, *glass-mini-train* and *glass-mini-test*

- Explain how classifier accuracy is measured, and what is meant by class noise and irrelevant attributes

Experiment with the IBk classifier for nearest neighbour learning:

- load *glass* data; list attribute names and identify the class attribute
- classify using IBk, testing with cross-validation
- repeat using 10 and then 20 nearest neighbours
- repeat all this for the *glass-minusatt* dataset
- repeat all this for the *glass-withnoise* dataset
- interpret the results and draw conclusions about IBk.

Perform nearest neighbour classification yourself:

- load *glass-mini-normalized* and view the data
- pretend that the last instance is a test instance and classify it (use the Visualize panel to help)
- verify your answer by running IBk on *glass-mini-train* and *glass-mini-test*

Experiment with the J48 decision tree learner:

- load *glass* data and classify using J48
- visualize the tree and simulate its effect on a particular test instance
- visualize the classifier errors and interpret one of them
- note J48 classification accuracy on *glass*, *glass-minusatt* and *glass-withnoise*.
- interpret the results and draw conclusions about J48.

Compare nearest neighbour to decision tree learning:

- draw conclusions about relative performance of IBk and J48's performance on *glass*, *glass-minusatt* and *glass-withnoise*.

Tutorial 3: Naïve Bayes and support vector machines

Introduce the boundary visualizer tool

Introduce the datasets *vehicle*, *kr-vs-kp*, *glass*, *waveform-5000* and *generated*.

Apply Naïve Bayes (NB) and J48 on several datasets:

- apply NB to *vehicle*, *kr-vs-kp*, *glass*, *waveform-5000* and *generated*, using 10-fold cross-validation.
- apply J48 to the same datasets.
- summarize the results
- draw an inference about the datasets where NB outperformed J48.

Investigate linear support vector machines:

- introduce the datasets *glass*, *glass-RINa*, *vehicle* and *vehicle-sub*
- apply a support vector machine learner (SMO) to *glass-RINa*, evaluating on the training set

- apply the classification boundary visualizer, and also visualize the classification errors (separately)
- describe the model built and explain the classification errors
- change SMO's complexity parameter *c* option and repeat
- comment on the difference *c* makes.

Investigate linear and non-linear support vector machines:

- apply SMO to *vehicle-sub*, again evaluating on the training set
- apply the classification boundary visualizer, and visualize the classifier errors
- change the "exponent" option of the kernel "PolyKernel" from 1 to 2 and repeat
- explain the differences in the test results
- add/remove points in the boundary visualizer to change the decision boundary's shape.

Tutorial 4: Preprocessing

Introduce the datasets *sick*, *vote*, *mushroom* and *letter*.

Apply discretization:

- explain what discretization is
- load the *sick* dataset and look at the attributes
- classify using NB, evaluating with cross-validation
- apply the supervised discretization filter and look at the effect (in the Preprocess panel)
- apply unsupervised discretization with different numbers of bins and look at the effect
- use the FilteredClassifier with NB and supervised discretization, evaluating with cross-validation
- repeat using unsupervised discretization with different numbers of bins
- compare and interpret the results.

Apply feature selection using CfsSubsetEval:

- explain what feature selection is
- load the *mushroom* dataset and apply J48, IBk and NB, evaluating with cross-validation
- select attributes using CfsSubsetEval and GreedyStepwise search
- interpret the results
- use AttributeSelectedClassifier (with CfsSubsetEval and GreedyStepwise search) for classifiers J48, IBk and NB, evaluating with cross-validation
- interpret the results.

Apply feature selection using WrapperSubsetEval:

- load the *vote* dataset and apply J48, IBk and NB, evaluating with cross-validation
- select attributes using WrapperSubsetEval with InfoGainAttributeEval and RankSearch, with the J48 classifier
- interpret the results
- use AttributeSelectedClassifier (with WrapperSubsetEval, InfoGainAttributeEval and RankSearch) with classifiers J48, IBk and NB, evaluating with cross-validation

- interpret the results.

Sampling a dataset:

- load the *letter* dataset and examine a particular (numeric) attribute
- apply the Resample filter to select half the dataset
- examine the same attribute and comment on the results.

Tutorial 5: Text mining

How to increase the memory size for Weka.

Introduce the datasets *ReutersCorn-Train* and *ReutersGrain-Train*.

Classify articles using binary attributes:

- load *ReutersCorn-train*
- apply StringToWordVector, with lower case tokens, alphabetic tokenizer, 2500 words to keep
- examine and interpret the result
- classify using NB and SMO, recording the TP and FP rates for positive instances, and the ROC area
- interpret the results to compare the classifiers
- discuss whether TP or FP is likely to be more important for this problem
- use AttributeSelectedClassifier (with InfoGain and Ranker search, selecting 100 attributes) with the same classifiers
- look at the words that have been retained, and comment
- compare the results for classification with and without attribute selection

Classify articles using word count attributes:

- load *ReutersCorn-train*
- apply StringToWordVector, with lower case tokens, alphabetic tokenizer, 2500 words to keep, and wordCount set to true
- examine and interpret the results
- classify using Naïve Bayes Multinomial (NBM) and SMO, recording the same figures as above
- compare the results with those above for binary attributes
- undo StringToWordVector and reapply with wordCount set to false
- reclassify with AttributeSelectedClassifier (with InfoGain and Ranker search) using NB and SMO, with 100, 50, 25 attributes
- compare NB with and without attribute selection, and the same for SMO
- compare NB with binary attributes against NBM with word count attributes, and the same for SMO

Classify unknown instances:

- use NBM models built from *ReutersCorn-train* and *ReutersGrain-train* to classify a mystery instance (*Mystery1*)
- repeat using SMO models
- comment on the findings
- use the same NBM models to classify a second mystery instance (*Mystery2*).

Tutorial 6: Association rules

Introduce the datasets *vote*, *weather.nominal* and *supermarket*.

Apply an association rule learner (Apriori):

- load *vote*, go to the Associate panel, and apply the Apriori learner
- discuss the meaning of the rules
- find out how a rule's confidence is computed
- identify the "support" and "number of instances predicted correctly" of certain rules
- change the number of rules in the output
- what is the criterion for "best rules"?
- find rules that mean certain things

Finding association rules manually:

- load *weather.nominal* and look at the data
- find the support and confidence for a certain rule
- consider rules with multiple parts in the consequent

Make association rules for the supermarket dataset:

- load *supermarket*
- generate 30 association rules and discuss some inferences you would make from them