WEKA Tutorial

TIM 245: Data Mining

Yunfei Chen
Based on Anita Krishnakumar
What is WEKA?

- Machine learning/data mining software written in Java (distributed under the GNU Public License)
- Java package - Collection of ML Algorithms
- Main features:
  - Comprehensive set of data pre-processing tools, learning algorithms and evaluation methods
  - Graphical user interfaces (incl. data visualization)
  - Environment for comparing learning algorithms
What is WEKA? (contd.)

- Schemes for classification include: decision trees, rule learners, naïve Bayes, decision tables, locally weighted regression, SVMs, instance-based learners, logistic regression, voted perceptrons, multi-layer perceptron

- Schemes for numeric prediction include: linear regression, model tree generators, locally-weighted regression, instance-based learners, decision tables, multi-layer perceptron

- Meta-schemes include: Bagging, boosting, stacking, regression via classification, classification via regression, cost sensitive classification

- Schemes for clustering: EM and Cobweb
Getting started

- The downloads and installation instructions can be found at http://www.cs.waikato.ac.nz/ml/weka

- It comes with a Graphical User Interface (GUI), but can also be called from your own Java code. You can even write your own batch files for tasks that you need to execute more than once, maybe with slightly different parameters each time.

- To start Weka in command line interface, change into the weka directory and type:
  java -jar weka.jar
Weka GUI Chooser

- Weka is a Java program distributed as a JAR file. You can run it directly using the command:
  ```java
  java -Xmx256m -jar weka.jar
  (-Xmx256m indicating 256m memory is allocated to run this program.)
  ```

- When you start up Weka, you'll have a choice between the command line interface (CLI), the Experimenter, the Explorer and Knowledge flow.

- The Explorer, Knowledge flow and Experimenter are GUIs whereas the CLI is the simple command line interface.
The Explorer

- Gives access to all facilities of Weka using menu selection and form filling

- Prepare the data, open the Explorer and load the data

- Flip back and forth between results, evaluate models built on different datasets and visualize graphically both models and datasets, including classification errors
Preparing the data

- Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
- Weka’s native data storage format is ARFF (Attribute-relation File Format)
- ARFF files have 2 sections HEADER and DATA
- HEADER:
  ```
  @RELATION iris
  @ATTRIBUTE sepallength NUMERIC
  @ATTRIBUTE sepalwidth NUMERIC
  @ATTRIBUTE petallength NUMERIC
  @ATTRIBUTE petalwidth NUMERIC
  @ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
  ```
- DATA:
  ```
  @DATA
  5.1,3.5,1.4,0.2,Iris-setosa
  4.9,3.0,1.4,0.2,Iris-setosa
  4.7,3.2,1.3,0.2,Iris-setosa
  4.6,3.1,1.5,0.2,Iris-setosa
  5.0,3.6,1.4,0.2,Iris-setosa
  5.4,3.9,1.7,0.4,Iris-setosa
  4.6,3.4,1.4,0.3,Iris-setosa
  ```
- More information regarding ARFF format can be found at http://www.cs.waikato.ac.nz/~ml/weka/arff.html
Loading the data

- For example, open the Iris data to find the following screen.

- You'll notice that Weka now provides some information about the data, such as for example the number of instances, the number of attributes, and also some statistical information about the attributes one at a time.

- The six tabs along the top are the basic operations that the Explorer supports.
Visualizing the data

- The main GUI will show a histogram for the attribute distributions for a single selected attribute at a time, by default this is the class attribute.

- Note that the individual colors indicate the individual classes (the Iris dataset has 3).

- If you move the mouse over the histogram, it will show you the ranges and how many samples fall in each range.

- The button VISUALIZE ALL will let you bring up a screen showing all distributions at once as in the picture below.
There is also a tab called VISUALIZE which can be used to obtain the scatterplots for all attribute pairs.

In the picture we can see that in some examples the clusters and the different colors correspond to each other such as for example in the plots for class/(any attribute) pairs and the petalwidth/petallength attribute pair, whereas for other pairs (sepalwidth/sepallength for example) it's much harder to separate the clusters by color.
Filters

- Pre-processing tools in WEKA are called “filters”
- This allows you to discard parts of your matrix without having to manipulate your original data file
- To apply a filter, you first have to select which type of filter you’d like by clicking on the CHOOSE button right underneath Filter
- WEKA contains filters for:
  - Discretization
  - Normalization
  - Resampling
  - Attribute selection
  - Combining attributes
  - Transforming, …
The Log file

- The log file is used to keep track of what you did.

- Clicking on LOG in your main GUI will bring up another window which will show exactly what you did, in this case it shows that we loaded the Iris data set and applied a filter.
Selecting Attributes

- Weka also provides techniques to discard irrelevant attributes and/or reduce the dimensionality of your dataset.

- After loading a dataset, click on the select attributes tag to open a GUI which will allow you to choose both the evaluation method (such as Principal Components Analysis for example) and the search method (ex. greedy or exhaustive search).

- Depending on the chosen combination, the actual time spend on selecting attributes can vary substantially and can also be very long, even for small datasets such as the Iris data with only five features (including the class attribute) for each of the 150 samples.

- It is also important to note that not all evaluation/search method combinations are valid, watch out for the error message in the Status bar.

- There's also a problem using Discretize while in the preprocessing mode, which leads to false results. If you need to use this filter, you can work around this by using the FilteredClassifier option in the classify menu.
The picture shows the results for selecting attributes in a sample application.
Classification

- Clicking on the classifier tab after loading a dataset into Weka and selecting the choose tab will bring up a menu with a number of choices for the classifier that is to be applied to the dataset.

- Note that you have 4 options on how to test the model you're building: Using the training set, supplied test set (you will need to specify the location of the test set in this case), cross validation and a percentage.

- The achieved accuracy of your model will vary, depending on the option you select. One pitfall to avoid is to select the training set as a test set, as that will result in an underestimate of the error rate.

- The resulting model with a lot of additional information will be displayed after you click on start. What exactly is contained in the output can be determined under options.
A sample output for applying the J4.8 decision tree algorithm to the Iris dataset is shown in the Figure.

The confusion matrix is displayed, as this gives a lot more information than just the prediction accuracy.
Classification (contd.)

- Visualizing the tree
- Visualizing the errors
Classification (contd.)

- A sample output for applying the BayesNet algorithm to the Iris dataset is shown in the Figure.
Clustering

- The clustering option is very similar to the classification described above, with a few differences regarding the options you select.

- For instance, there is an easy way to discard undesired attributes.
Weka also provides algorithms to extract association rules from non-numerical data as shown in the picture.
Opening Databases and URLs

- Detailed instructions for loading a windows database into Weka can be found at http://www.cs.waikato.ac.nz/\%7Eml/weka/opening_windows_DBs.html

- Weka also allows a user to select a URL as the source for a datafile (click on Open URL from the preprocess pane) or download data from a database (Open DB from the preprocess pane).
Sometimes it's desirable to run Weka from a command line interface. For example if you want to set up your experiment using a batch file.

Instructions on how to call Weka from Java

Examples for using the CLI are:

- `java weka.filters.unsupervised.attribute.Remove -V -R 1,4 -i trainingFile.arff -o myTrainingFile.arff`

  This filter removes all but the first and fourth attribute from a dataset stored in a file called trainingFile.arff and stores the result in myTrainingFile.arff

- `java weka.classifiers.trees.J48 -t myTrainingFile.arff -T myTestFile.arff -U -p 1 > Results.arff`

  Here, the J48 decision tree is applied to the file myTrainingFile.arff, which was created in the previous step. The test file is specified with the -T option. The results are redirected from the screen to a file called Results.arff and the options -U and -p 1 are options that determine the specific output form.
The Experimenter

- The experimenter, which can be run from both the command line and a GUI (easier to use), is a tool that allows you to perform more than one experiment at a time, maybe applying different techniques to a dataset, or the same technique repeatedly with different parameters.

- Experimenter makes it easy to compare the performance of different learning schemes.

- For classification and regression problems.

- Results can be written into file or database.

- Evaluation options: cross-validation, learning curve, hold-out.

- Can also iterate over different parameter settings.

- Significance-testing built in.
The Experimenter (contd.)

- Setting up the Experiment for different datasets and different algorithms

- The picture shows the experiment setup for the datasets iris, contact-lenses and weather and the algorithms chosen are Decision trees, BayesNet and naiveBayes
The Experimenter (contd.)

- Running the experiment

![Image of Woka Experiment Environment](image)

Log:

- 13:47:00: Started
- 13:48:01: Started
- 13:48:20: Started
- 13:48:22: There were 0 errors

Status:

- Not running
The Experimenter (contd.)

- Analyzing the results
The Knowledge Flow GUI

- Java-Beans-based interface for setting up and running machine learning experiments

- Data sources, classifiers, etc. are beans and can be connected graphically

- Data “flows” through components: e.g.,
  - “data source” -> “filter” -> “classifier” -> “evaluator”

- Layouts can be saved and loaded again later
The Knowledge Flow GUI (contd.)

- Setting up the Knowledge flow
The Knowledge Flow GUI (contd.)

- An example of the complete Knowledge flow setup
The Knowledge Flow GUI (contd.)

- Results from Text Viewer

```plaintext
### Evaluation result

Scheme: SMO  

Correctly Classified Instances 144 95 %
Incorrectly Classified Instances 6 4 %
Kappa statistic 0.94
Mean absolute error 0.3311
Root mean squared error 0.286
Relative absolute error 52 %
Root relative squared error 60.8301 %
Total Number of Instances 150

### Detailed Accuracy By Class

<table>
<thead>
<tr>
<th></th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris-setosa</td>
<td>0.96</td>
<td>0.04</td>
<td>0.923</td>
<td>0.96</td>
<td>0.941</td>
<td>Iris-setosa</td>
</tr>
<tr>
<td>Iris-versicolor</td>
<td>0.92</td>
<td>0.02</td>
<td>0.958</td>
<td>0.92</td>
<td>0.939</td>
<td>Iris-versicolor</td>
</tr>
<tr>
<td>Iris-virginica</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Iris-virginica</td>
</tr>
</tbody>
</table>

### Confusion Matrix

```

```
Using the Weka API

- Add weka.jar to class path.
- API Documentation
- Weka Manual

```java
import weka.core.Instances;
import weka.classifiers.trees.J48;
...
Instances data = ... // from somewhere
String[] options = new String[1];
options[0] = "-U"; // unpruned tree
J48 tree = new J48(); // new instance of tree
tree.setOptions(options); // set the options
tree.buildClassifier(data); // build classifier
```
Further reading

- The Explorer Guide

- The Experimenter Guide

- The Knowledge Flow Guide
  http://www-i9.informatik.rwth-aachen.de/Lehre/PraktikumDM/weka/README_KnowledgeFlow.txt