



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

WEKA Explorer User Guide for Version 3-5-7

Richard Kirkby
Eibe Frank
Peter Reutemann

December 19, 2007

Contents

1	Launching WEKA	2
2	The WEKA Explorer	4
2.1	Section Tabs	4
2.2	Status Box	4
2.3	Log Button	4
2.4	WEKA Status Icon	5
2.5	Graphical output	5
3	Preprocessing	6
3.1	Loading Data	6
3.2	The Current Relation	6
3.3	Working With Attributes	7
3.4	Working With Filters	8
4	Classification	10
4.1	Selecting a Classifier	10
4.2	Test Options	10
4.3	The Class Attribute	11
4.4	Training a Classifier	12
4.5	The Classifier Output Text	12
4.6	The Result List	12
5	Clustering	14
5.1	Selecting a Clusterer	14
5.2	Cluster Modes	14
5.3	Ignoring Attributes	14
5.4	Working with Filters	15
5.5	Learning Clusters	15
6	Associating	16
6.1	Setting Up	16
6.2	Learning Associations	16
7	Selecting Attributes	17
7.1	Searching and Evaluating	17
7.2	Options	17
7.3	Performing Selection	17
8	Visualizing	19
8.1	The scatter plot matrix	19
8.2	Selecting an individual 2D scatter plot	19
8.3	Selecting Instances	20

1 Launching WEKA

The new menu-driven GUI in WEKA (class `weka.gui.Main`) succeeds the old GUI Chooser (class `weka.gui.GUIChooser`). Its MDI (“multiple document interface”) appearance makes it easier to keep track of all the open windows. If one prefers an SDI (“single document interface”) driven layout, one can invoke this with option `-gui sdi` on the commandline.



The menu consists of six sections:

1. Program

- **LogWindow** Opens a log window that captures all that is printed to *stdout* or *stderr*. Useful for environments like MS Windows, where WEKA is normally not started from a terminal.
- **Exit** Closes WEKA.

2. Applications Lists the main applications within WEKA.

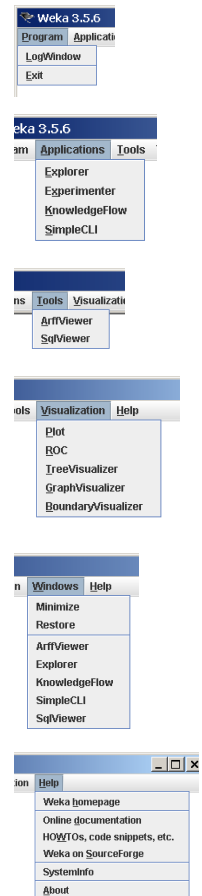
- **Explorer** An environment for exploring data with WEKA (the rest of this documentation deals with this application in more detail).
- **Experimenter** An environment for performing experiments and conducting statistical tests between learning schemes.
- **KnowledgeFlow** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.
- **SimpleCLI** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

3. Tools Other useful applications.

- **ArffViewer** An MDI application for viewing ARFF files in spreadsheet format.
- **SqlViewer** represents an SQL worksheet, for querying databases via JDBC.

4. Visualization Ways of visualizing data with WEKA.

- **Plot** For plotting a 2D plot of a dataset.
- **ROC** Displays a previously saved ROC curve.



- **TreeVisualizer** For displaying directed graphs, e.g., a decision tree.
 - **GraphVisualizer** Visualizes XML BIF or DOT format graphs, e.g., for Bayesian networks.
 - **BoundaryVisualizer** Allows the visualization of classifier decision boundaries in two dimensions.
5. **Windows** All open windows are listed here.
- **Minimize** Minimizes all current windows.
 - **Restore** Restores all minimized windows again.
6. **Help** Online resources for WEKA can be found here.
- **Weka homepage** Opens a browser window with WEKA's homepage.
 - **Online documentation** Directs to the WekaDoc Wiki [4].
 - **HOWTOs, code snippets, etc.** The general WekaWiki [3], containing lots of examples and HOWTOs around the development and use of WEKA.
 - **Weka on Sourceforge** WEKA's project homepage on Sourceforge.net.
 - **SystemInfo** Lists some internals about the Java/WEKA environment, e.g., the `CLASSPATH`.
 - **About** The infamous "About" box.

To make it easy for the user to add new functionality to the menu without having to modify the code of WEKA itself, the GUI now offers a plugin mechanism for such add-ons. Due to the inherent dynamic class discovery, plugins only need to implement the `weka.gui.MainMenuExtension` interface and WEKA notified of the package they reside in to be displayed in the menu under "Extensions" (this extra menu appears automatically as soon as extensions are discovered). More details can be found in the Wiki article "Extensions for Weka's main GUI" [6].

If you launch WEKA from a terminal window, some text begins scrolling in the terminal. Ignore this text unless something goes wrong, in which case it can help in tracking down the cause (the *LogWindow* from the *Program* menu displays that information as well).

This User Manual, which is also available online on the *WekaDoc Wiki* [4], focuses on using the Explorer but does not explain the individual data preprocessing tools and learning algorithms in WEKA. For more information on the various filters and learning methods in WEKA, see the book *Data Mining* [2].

2 The WEKA Explorer

2.1 Section Tabs

At the very top of the window, just below the title bar, is a row of tabs. When the Explorer is first started only the first tab is active; the others are greyed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.

The tabs are as follows:

1. **Preprocess.** Choose and modify the data being acted on.
2. **Classify.** Train and test learning schemes that classify or perform regression.
3. **Cluster.** Learn clusters for the data.
4. **Associate.** Learn association rules for the data.
5. **Select attributes.** Select the most relevant attributes in the data.
6. **Visualize.** View an interactive 2D plot of the data.

Once the tabs are active, clicking on them flicks between different screens, on which the respective actions can be performed. The bottom area of the window (including the status box, the log button, and the Weka bird) stays visible regardless of which section you are in.

The Explorer can be easily extended with custom tabs. The Wiki article “Adding tabs in the Explorer” [7] explains this in detail.

2.2 Status Box

The status box appears at the very bottom of the window. It displays messages that keep you informed about what’s going on. For example, if the Explorer is busy loading a file, the status box will say that.

TIP—right-clicking the mouse anywhere inside the status box brings up a little menu. The menu gives two options:

1. **Memory information.** Display in the log box the amount of memory available to WEKA.
2. **Run garbage collector.** Force the Java garbage collector to search for memory that is no longer needed and free it up, allowing more memory for new tasks. Note that the garbage collector is constantly running as a background task anyway.

2.3 Log Button

Clicking on this button brings up a separate window containing a scrollable text field. Each line of text is stamped with the time it was entered into the log. As you perform actions in WEKA, the log keeps a record of what has happened. For people using the command line or the SimpleCLI, the log now also contains the full setup strings for classification, clustering, attribute selection, etc., so

that it is possible to copy/paste them elsewhere. Options for dataset(s) and, if applicable, the class attribute still have to be provided by the user (e.g., `-t` for classifiers or `-i` and `-o` for filters).

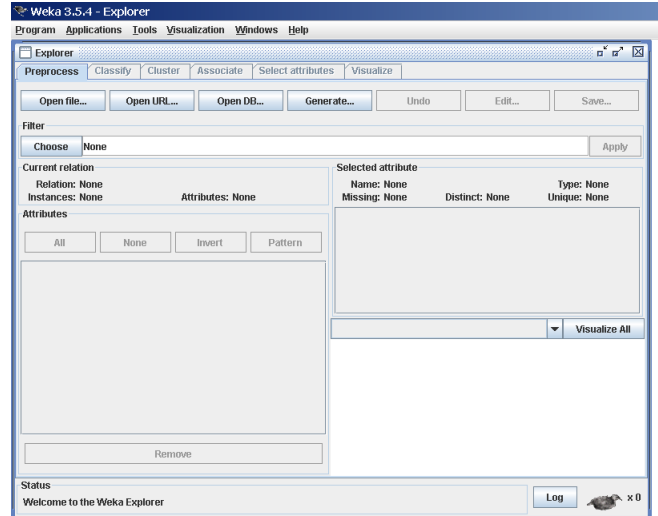
2.4 WEKA Status Icon

To the right of the status box is the WEKA status icon. When no processes are running, the bird sits down and takes a nap. The number beside the \times symbol gives the number of concurrent processes running. When the system is idle it is zero, but it increases as the number of processes increases. When any process is started, the bird gets up and starts moving around. If it's standing but stops moving for a long time, it's sick: something has gone wrong! In that case you should restart the WEKA Explorer.

2.5 Graphical output

Most graphical displays in WEKA, e.g., the GraphVisualizer or the TreeVisualizer, support saving the output to a file. A dialog for saving the output can be brought up with *Alt+Shift+left-click*. Supported formats are currently Windows Bitmap, JPEG, PNG and EPS (encapsulated Postscript). The dialog also allows you to specify the dimensions of the generated image.

3 Preprocessing



3.1 Loading Data

The first four buttons at the top of the preprocess section enable you to load data into WEKA:

1. **Open file....** Brings up a dialog box allowing you to browse for the data file on the local file system.
2. **Open URL....** Asks for a Uniform Resource Locator address for where the data is stored.
3. **Open DB....** Reads data from a database. (Note that to make this work you might have to edit the file in `weka/experiment/DatabaseUtils.props`.)
4. **Generate....** Enables you to generate artificial data from a variety of DataGenerators.

Using the **Open file...** button you can read files in a variety of formats: WEKA's ARFF format, CSV format, C4.5 format, or serialized Instances format. ARFF files typically have a `.arff` extension, CSV files a `.csv` extension, C4.5 files a `.data` and `.names` extension, and serialized Instances objects a `.bsi` extension.

NB: This list of formats can be extended by adding custom file converters to the `weka.core.converters` package.

3.2 The Current Relation

Once some data has been loaded, the Preprocess panel shows a variety of information. The **Current relation** box (the "current relation" is the currently loaded data, which can be interpreted as a single relational table in database terminology) has three entries:

1. **Relation.** The name of the relation, as given in the file it was loaded from. Filters (described below) modify the name of a relation.
2. **Instances.** The number of instances (data points/records) in the data.
3. **Attributes.** The number of attributes (features) in the data.

3.3 Working With Attributes

Below the **Current relation** box is a box titled **Attributes**. There are four buttons, and beneath them is a list of the attributes in the current relation. The list has three columns:

1. **No..** A number that identifies the attribute in the order they are specified in the data file.
2. **Selection tick boxes.** These allow you select which attributes are present in the relation.
3. **Name.** The name of the attribute, as it was declared in the data file.

When you click on different rows in the list of attributes, the fields change in the box to the right titled **Selected attribute**. This box displays the characteristics of the currently highlighted attribute in the list:

1. **Name.** The name of the attribute, the same as that given in the attribute list.
2. **Type.** The type of attribute, most commonly Nominal or Numeric.
3. **Missing.** The number (and percentage) of instances in the data for which this attribute is missing (unspecified).
4. **Distinct.** The number of different values that the data contains for this attribute.
5. **Unique.** The number (and percentage) of instances in the data having a value for this attribute that no other instances have.

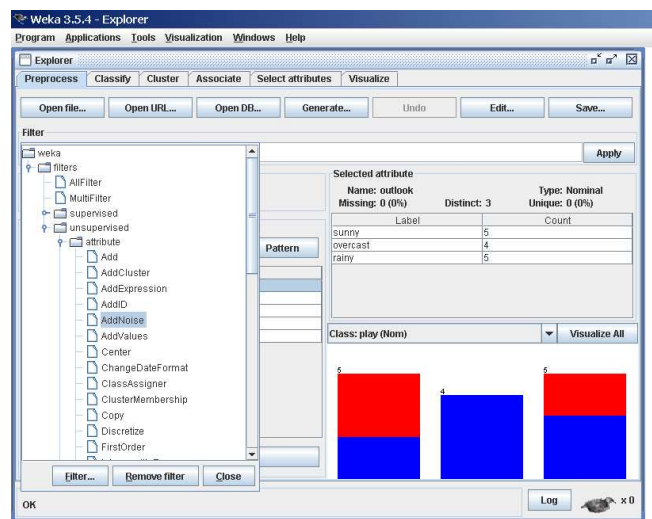
Below these statistics is a list showing more information about the values stored in this attribute, which differ depending on its type. If the attribute is nominal, the list consists of each possible value for the attribute along with the number of instances that have that value. If the attribute is numeric, the list gives four statistics describing the distribution of values in the data—the minimum, maximum, mean and standard deviation. And below these statistics there is a coloured histogram, colour-coded according to the attribute chosen as the *Class* using the box above the histogram. (This box will bring up a drop-down list of available selections when clicked.) Note that only nominal *Class* attributes will result in a colour-coding. Finally, after pressing the **Visualize All** button, histograms for all the attributes in the data are shown in a separate window.

Returning to the attribute list, to begin with all the tick boxes are unticked. They can be toggled on/off by clicking on them individually. The four buttons above can also be used to change the selection:

1. **All.** All boxes are ticked.
2. **None.** All boxes are cleared (unticked).
3. **Invert.** Boxes that are ticked become unticked and *vice versa*.
4. **Pattern.** Enables the user to select attributes based on a Perl 5 Regular Expression. E.g., `.*_id` selects all attributes which name ends with `_id`.

Once the desired attributes have been selected, they can be removed by clicking the **Remove** button below the list of attributes. Note that this can be undone by clicking the **Undo** button, which is located next to the **Edit** button in the top-right corner of the Preprocess panel.

3.4 Working With Filters



The preprocess section allows filters to be defined that transform the data in various ways. The **Filter** box is used to set up the filters that are required. At the left of the **Filter** box is a **Choose** button. By clicking this button it is possible to select one of the filters in WEKA. Once a filter has been selected, its name and options are shown in the field next to the **Choose** button. Clicking on this box with the *left* mouse button brings up a GenericObjectEditor dialog box. A click with the *right* mouse button (or *Alt+Shift+left click*) brings up a menu where you can choose, either to display the properties in a GenericObjectEditor dialog box, or to copy the current setup string to the clipboard.

The GenericObjectEditor Dialog Box

The GenericObjectEditor dialog box lets you configure a filter. The same kind of dialog box is used to configure other objects, such as classifiers and clusterers (see below). The fields in the window reflect the available options.

Right-clicking (or *Alt+Shift+Left-Click*) on such a field will bring up a popup menu, listing the following options:

1. **Show properties...** has the same effect as left-clicking on the field, i.e., a dialog appears allowing you to alter the settings.
2. **Copy configuration to clipboard** copies the currently displayed configuration string to the system's clipboard and therefore can be used anywhere else in WEKA or in the console. This is rather handy if you have to setup complicated, nested schemes.
3. **Enter configuration...** is the “receiving” end for configurations that got copied to the clipboard earlier on. In this dialog you can enter a classname followed by options (if the class supports these). This also allows you to transfer a filter setting from the Preprocess panel to a **FilteredClassifier** used in the Classify panel.

Left-Clicking on any of these gives an opportunity to alter the filters settings. For example, the setting may take a text string, in which case you type the string into the text field provided. Or it may give a drop-down box listing several states to choose from. Or it may do something else, depending on the information required. Information on the options is provided in a tool tip if you let the mouse pointer hover of the corresponding field. More information on the filter and its options can be obtained by clicking on the **More** button in the **About** panel at the top of the GenericObjectEditor window.

Some objects display a brief description of what they do in an **About** box, along with a **More** button. Clicking on the **More** button brings up a window describing what the different options do. Others have an additional button, *Capabilities*, which lists the types of attributes and classes the object can handle.

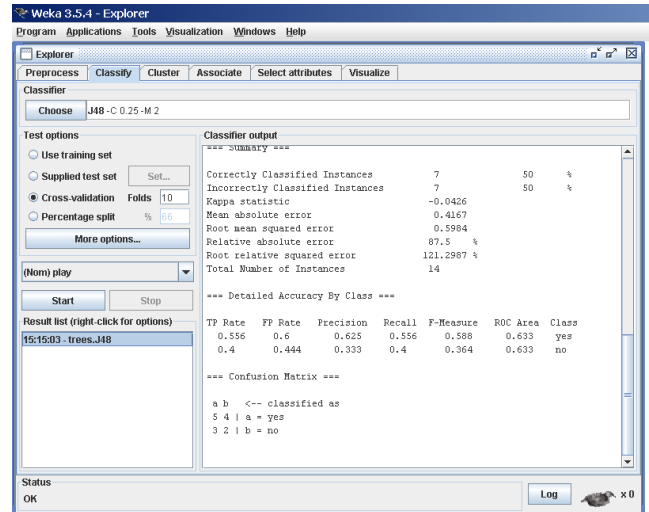
At the bottom of the GenericObjectEditor dialog are four buttons. The first two, **Open...** and **Save...** allow object configurations to be stored for future use. The **Cancel** button backs out without remembering any changes that have been made. Once you are happy with the object and settings you have chosen, click **OK** to return to the main Explorer window.

Applying Filters

Once you have selected and configured a filter, you can apply it to the data by pressing the **Apply** button at the right end of the **Filter** panel in the Preprocess panel. The Preprocess panel will then show the transformed data. The change can be undone by pressing the **Undo** button. You can also use the **Edit...** button to modify your data manually in a dataset editor. Finally, the **Save...** button at the top right of the Preprocess panel saves the current version of the relation in file formats that can represent the relation, allowing it to be kept for future use.

Note: Some of the filters behave differently depending on whether a class attribute has been set or not (using the box above the histogram, which will bring up a drop-down list of possible selections when clicked). In particular, the “supervised filters” require a class attribute to be set, and some of the “unsupervised attribute filters” will skip the class attribute if one is set. Note that it is also possible to set *Class* to *None*, in which case no class is set.

4 Classification



4.1 Selecting a Classifier

At the top of the classify section is the **Classifier** box. This box has a text field that gives the name of the currently selected classifier, and its options. Clicking on the text box with the left mouse button brings up a `GenericObjectEditor` dialog box, just the same as for filters, that you can use to configure the options of the current classifier. With a *right click* (or *Alt+Shift+left click*) you can once again copy the setup string to the clipboard or display the properties in a `GenericObjectEditor` dialog box. The **Choose** button allows you to choose one of the classifiers that are available in WEKA.

4.2 Test Options

The result of applying the chosen classifier will be tested according to the options that are set by clicking in the **Test options** box. There are four test modes:

1. **Use training set.** The classifier is evaluated on how well it predicts the class of the instances it was trained on.
2. **Supplied test set.** The classifier is evaluated on how well it predicts the class of a set of instances loaded from a file. Clicking the **Set...** button brings up a dialog allowing you to choose the file to test on.
3. **Cross-validation.** The classifier is evaluated by cross-validation, using the number of folds that are entered in the **Folds** text field.
4. **Percentage split.** The classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the **%** field.

Note: No matter which evaluation method is used, the model that is output is always the one build from *all* the training data. Further testing options can be set by clicking on the **More options...** button:

1. **Output model.** The classification model on the full training set is output so that it can be viewed, visualized, etc. This option is selected by default.
2. **Output per-class stats.** The precision/recall and true/false statistics for each class are output. This option is also selected by default.
3. **Output entropy evaluation measures.** Entropy evaluation measures are included in the output. This option is not selected by default.
4. **Output confusion matrix.** The confusion matrix of the classifier's predictions is included in the output. This option is selected by default.
5. **Store predictions for visualization.** The classifier's predictions are remembered so that they can be visualized. This option is selected by default.
6. **Output predictions.** The predictions on the evaluation data are output. Note that in the case of a cross-validation the instance numbers do not correspond to the location in the data!
7. **Output additional attributes.** If additional attributes need to be output alongside the predictions, e.g., an ID attribute for tracking misclassifications, then the index of this attribute can be specified here. The usual Weka ranges are supported, "first" and "last" are therefore valid indices as well (example: "first-3,6,8,12-last").
8. **Cost-sensitive evaluation.** The errors is evaluated with respect to a cost matrix. The **Set...** button allows you to specify the cost matrix used.
9. **Random seed for xval / % Split.** This specifies the random seed used when randomizing the data before it is divided up for evaluation purposes.
10. **Preserve order for % Split.** This suppresses the randomization of the data before splitting into train and test set.
11. **Output source code.** If the classifier can output the built model as Java source code, you can specify the class name here. The code will be printed in the "Classifier output" area.

4.3 The Class Attribute

The classifiers in WEKA are designed to be trained to predict a single 'class' attribute, which is the target for prediction. Some classifiers can only learn nominal classes; others can only learn numeric classes (regression problems); still others can learn both.

By default, the class is taken to be the last attribute in the data. If you want to train a classifier to predict a different attribute, click on the box below the **Test options** box to bring up a drop-down list of attributes to choose from.

4.4 Training a Classifier

Once the classifier, test options and class have all been set, the learning process is started by clicking on the **Start** button. While the classifier is busy being trained, the little bird moves around. You can stop the training process at any time by clicking on the **Stop** button.

When training is complete, several things happen. The **Classifier output** area to the right of the display is filled with text describing the results of training and testing. A new entry appears in the **Result list** box. We look at the result list below; but first we investigate the text that has been output.

4.5 The Classifier Output Text

The text in the **Classifier output** area has scroll bars allowing you to browse the results. Clicking with the left mouse button into the text area, while holding **Alt** and **Shift**, brings up a dialog that enables you to save the displayed output in a variety of formats (currently, BMP, EPS, JPEG and PNG). Of course, you can also resize the Explorer window to get a larger display area. The output is split into several sections:

1. **Run information.** A list of information giving the learning scheme options, relation name, instances, attributes and test mode that were involved in the process.
2. **Classifier model (full training set).** A textual representation of the classification model that was produced on the full training data.
3. The results of the chosen test mode are broken down thus:
4. **Summary.** A list of statistics summarizing how accurately the classifier was able to predict the true class of the instances under the chosen test mode.
5. **Detailed Accuracy By Class.** A more detailed per-class break down of the classifier's prediction accuracy.
6. **Confusion Matrix.** Shows how many instances have been assigned to each class. Elements show the number of test examples whose actual class is the row and whose predicted class is the column.
7. **Source code** (optional). This section lists the Java source code if one chose "Output source code" in the "More options" dialog.

4.6 The Result List

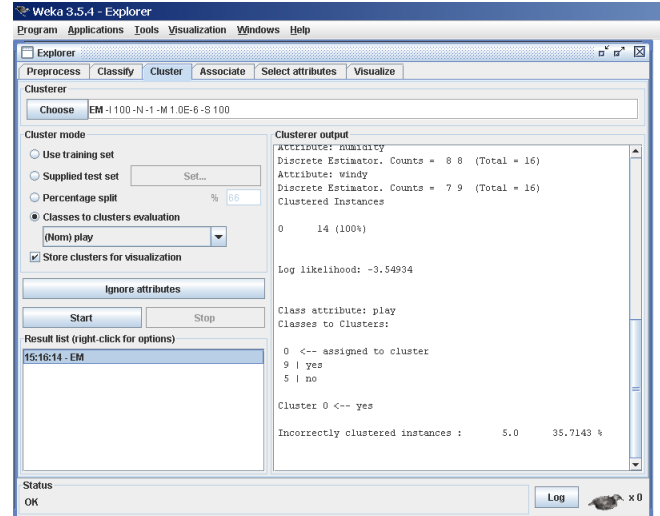
After training several classifiers, the result list will contain several entries. Left-clicking the entries flicks back and forth between the various results that have been generated. Pressing **Delete** removes a selected entry from the results. Right-clicking an entry invokes a menu containing these items:

1. **View in main window.** Shows the output in the main window (just like left-clicking the entry).

2. **View in separate window.** Opens a new independent window for viewing the results.
3. **Save result buffer.** Brings up a dialog allowing you to save a text file containing the textual output.
4. **Load model.** Loads a pre-trained model object from a binary file.
5. **Save model.** Saves a model object to a binary file. Objects are saved in Java ‘serialized object’ form.
6. **Re-evaluate model on current test set.** Takes the model that has been built and tests its performance on the data set that has been specified with the **Set..** button under the **Supplied test set** option.
7. **Visualize classifier errors.** Brings up a visualization window that plots the results of classification. Correctly classified instances are represented by crosses, whereas incorrectly classified ones show up as squares.
8. **Visualize tree** or **Visualize graph.** Brings up a graphical representation of the structure of the classifier model, if possible (i.e. for decision trees or Bayesian networks). The graph visualization option only appears if a Bayesian network classifier has been built. In the tree visualizer, you can bring up a menu by right-clicking a blank area, pan around by dragging the mouse, and see the training instances at each node by clicking on it. CTRL-clicking zooms the view out, while SHIFT-dragging a box zooms the view in. The graph visualizer should be self-explanatory.
9. **Visualize margin curve.** Generates a plot illustrating the prediction margin. The margin is defined as the difference between the probability predicted for the actual class and the highest probability predicted for the other classes. For example, boosting algorithms may achieve better performance on test data by increasing the margins on the training data.
10. **Visualize threshold curve.** Generates a plot illustrating the trade-offs in prediction that are obtained by varying the threshold value between classes. For example, with the default threshold value of 0.5, the predicted probability of ‘positive’ must be greater than 0.5 for the instance to be predicted as ‘positive’. The plot can be used to visualize the precision/recall trade-off, for ROC curve analysis (true positive rate *vs* false positive rate), and for other types of curves.
11. **Visualize cost curve.** Generates a plot that gives an explicit representation of the expected cost, as described by [1].
12. **Plugins.** This menu item only appears if there are visualization plugins available (by default: none). More about these plugins can be found in the *WekaWiki* article “Explorer visualization plugins” [8].

Options are greyed out if they do not apply to the specific set of results.

5 Clustering



5.1 Selecting a Clusterer

By now you will be familiar with the process of selecting and configuring objects. Clicking on the clustering scheme listed in the **Clusterer** box at the top of the window brings up a `GenericObjectEditor` dialog with which to choose a new clustering scheme.

5.2 Cluster Modes

The **Cluster mode** box is used to choose what to cluster and how to evaluate the results. The first three options are the same as for classification: **Use training set**, **Supplied test set** and **Percentage split** (Section 4.1)—except that now the data is assigned to clusters instead of trying to predict a specific class. The fourth mode, **Classes to clusters evaluation**, compares how well the chosen clusters match up with a pre-assigned class in the data. The drop-down box below this option selects the class, just as in the **Classify** panel.

An additional option in the **Cluster mode** box, the **Store clusters for visualization** tick box, determines whether or not it will be possible to visualize the clusters once training is complete. When dealing with datasets that are so large that memory becomes a problem it may be helpful to disable this option.

5.3 Ignoring Attributes

Often, some attributes in the data should be ignored when clustering. The **Ignore attributes** button brings up a small window that allows you to select which attributes are ignored. Clicking on an attribute in the window highlights it, holding down the SHIFT key selects a range of consecutive attributes, and holding down CTRL toggles individual attributes on and off. To cancel the selection, back out with the **Cancel** button. To activate it, click the **Select** button. The next time clustering is invoked, the selected attributes are ignored.

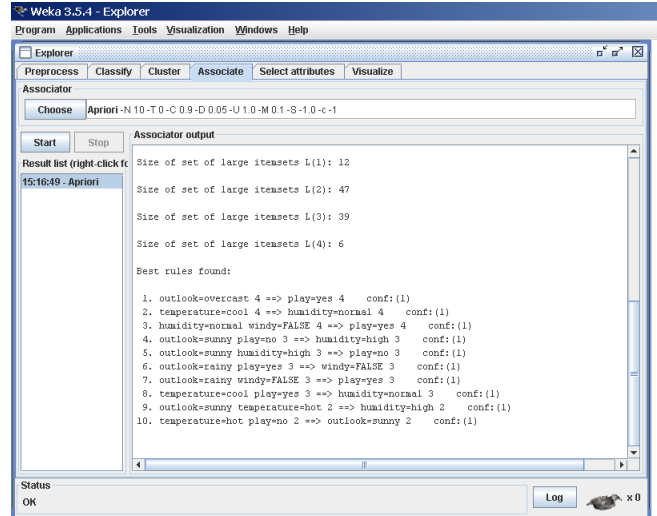
5.4 Working with Filters

The **FilteredClusterer** meta-clusterer offers the user the possibility to apply filters directly before the clusterer is learned. This approach eliminates the manual application of a filter in the **Preprocess** panel, since the data gets processed on the fly. Useful if one needs to try out different filter setups.

5.5 Learning Clusters

The **Cluster** section, like the **Classify** section, has **Start/Stop** buttons, a result text area and a result list. These all behave just like their classification counterparts. Right-clicking an entry in the result list brings up a similar menu, except that it shows only two visualization options: **Visualize cluster assignments** and **Visualize tree**. The latter is grayed out when it is not applicable.

6 Associating



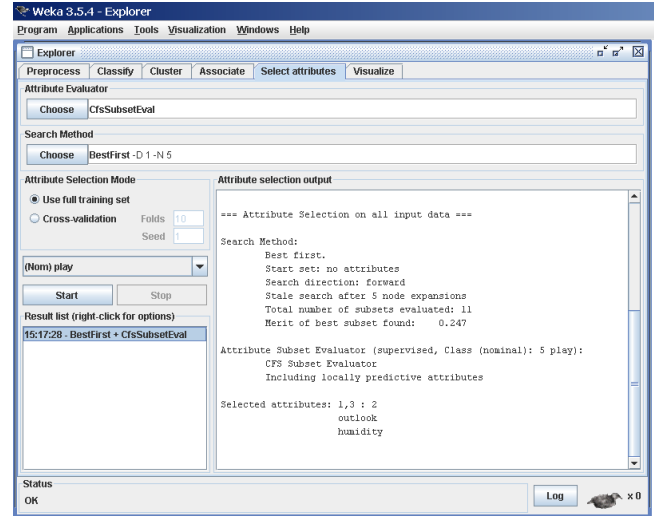
6.1 Setting Up

This panel contains schemes for learning association rules, and the learners are chosen and configured in the same way as the clusterers, filters, and classifiers in the other panels.

6.2 Learning Associations

Once appropriate parameters for the association rule learner have been set, click the **Start** button. When complete, right-clicking on an entry in the result list allows the results to be viewed or saved.

7 Selecting Attributes



7.1 Searching and Evaluating

Attribute selection involves searching through all possible combinations of attributes in the data to find which subset of attributes works best for prediction. To do this, two objects must be set up: an attribute evaluator and a search method. The evaluator determines what method is used to assign a worth to each subset of attributes. The search method determines what style of search is performed.

7.2 Options

The **Attribute Selection Mode** box has two options:

1. **Use full training set.** The worth of the attribute subset is determined using the full set of training data.
2. **Cross-validation.** The worth of the attribute subset is determined by a process of cross-validation. The **Fold** and **Seed** fields set the number of folds to use and the random seed used when shuffling the data.

As with **Classify** (Section 4.1), there is a drop-down box that can be used to specify which attribute to treat as the class.

7.3 Performing Selection

Clicking **Start** starts running the attribute selection process. When it is finished, the results are output into the result area, and an entry is added to the result list. Right-clicking on the result list gives several options. The first three, (**View in main window**, **View in separate window** and **Save result buffer**), are the same as for the classify panel. It is also possible to **Visualize**

reduced data, or if you have used an attribute transformer such as Principal-Components, **Visualize transformed data**. The reduced/transformed data can be saved to a file with the **Save reduced data...** or **Save transformed data...** option.

In case one wants to reduce/transform a training and a test at the same time and not use the AttributeSelectedClassifier from the classifier panel, it is best to use the AttributeSelection filter (a supervised attribute filter) in batch mode ('-b') from the command line or in the SimpleCLI. The batch mode allows one to specify an additional input and output file pair (options -r and -s), that is processed with the filter setup that was determined based on the training data (specified by options -i and -o).

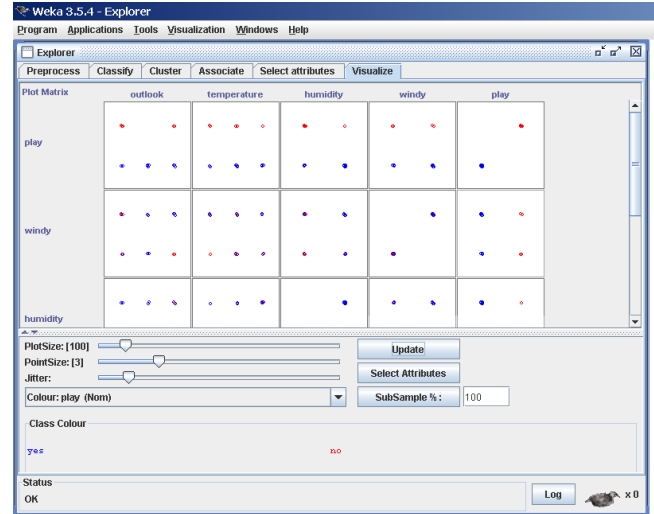
Here is an example for a Unix/Linux bash:

```
java weka.filters.supervised.attribute.AttributeSelection \  
    -E "weka.attributeSelection.CfsSubsetEval " \  
    -S "weka.attributeSelection.BestFirst -D 1 -N 5" \  
    -b \  
    -i <input1.arff> \  
    -o <output1.arff> \  
    -r <input2.arff> \  
    -s <output2.arff>
```

Notes:

- The “backslashes” at the end of each line tell the bash that the command is not finished yet. Using the SimpleCLI one has to use this command in one line without the backslashes.
- It is assumed that WEKA is available in the CLASSPATH, otherwise one has to use the -classpath option.
- The full filter setup is output in the log, as well as the setup for running regular attribute selection.

8 Visualizing



WEKA's visualization section allows you to visualize 2D plots of the current relation.

8.1 The scatter plot matrix

When you select the *Visualize* panel, it shows a scatter plot matrix for all the attributes, colour coded according to the currently selected class. It is possible to change the size of each individual 2D plot and the point size, and to randomly jitter the data (to uncover obscured points). It also possible to change the attribute used to colour the plots, to select only a subset of attributes for inclusion in the scatter plot matrix, and to sub sample the data. Note that changes will only come into effect once the **Update** button has been pressed.

8.2 Selecting an individual 2D scatter plot

When you click on a cell in the scatter plot matrix, this will bring up a separate window with a visualization of the scatter plot you selected. (We described above how to visualize particular results in a separate window—for example, classifier errors—the same visualization controls are used here.)

Data points are plotted in the main area of the window. At the top are two drop-down list buttons for selecting the axes to plot. The one on the left shows which attribute is used for the x-axis; the one on the right shows which is used for the y-axis.

Beneath the x-axis selector is a drop-down list for choosing the colour scheme. This allows you to colour the points based on the attribute selected. Below the plot area, a legend describes what values the colours correspond to. If the values are discrete, you can modify the colour used for each one by clicking on them and making an appropriate selection in the window that pops up.

To the right of the plot area is a series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution of values

of the attribute. These values are randomly scattered vertically to help you see concentrations of points. You can choose what axes are used in the main graph by clicking on these strips. Left-clicking an attribute strip changes the x-axis to that attribute, whereas right-clicking changes the y-axis. The ‘X’ and ‘Y’ written beside the strips shows what the current axes are (‘B’ is used for ‘both X and Y’).

Above the attribute strips is a slider labelled **Jitter**, which is a random displacement given to all points in the plot. Dragging it to the right increases the amount of jitter, which is useful for spotting concentrations of points. Without jitter, a million instances at the same point would look no different to just a single lonely instance.

8.3 Selecting Instances

There may be situations where it is helpful to select a subset of the data using the visualization tool. (A special case of this is the UserClassifier in the *Classify* panel, which lets you build your own classifier by interactively selecting instances.)

Below the y-axis selector button is a drop-down list button for choosing a selection method. A group of data points can be selected in four ways:

1. **Select Instance.** Clicking on an individual data point brings up a window listing its attributes. If more than one point appears at the same location, more than one set of attributes is shown.
2. **Rectangle.** You can create a rectangle, by dragging, that selects the points inside it.
3. **Polygon.** You can build a free-form polygon that selects the points inside it. Left-click to add vertices to the polygon, right-click to complete it. The polygon will always be closed off by connecting the first point to the last.
4. **Polyline.** You can build a polyline that distinguishes the points on one side from those on the other. Left-click to add vertices to the polyline, right-click to finish. The resulting shape is open (as opposed to a polygon, which is always closed).

Once an area of the plot has been selected using **Rectangle**, **Polygon** or **Polyline**, it turns grey. At this point, clicking the **Submit** button removes all instances from the plot except those within the grey selection area. Clicking on the **Clear** button erases the selected area without affecting the graph.

Once any points have been removed from the graph, the **Submit** button changes to a **Reset** button. This button undoes all previous removals and returns you to the original graph with all points included. Finally, clicking the **Save** button allows you to save the currently visible instances to a new ARFF file.

References

- [1] Drummond, C. and Holte, R. (2000) Explicitly representing expected cost: An alternative to ROC representation. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Publishers, San Mateo, CA.
- [2] Witten, I.H. and Frank, E. (2005) *Data Mining: Practical machine learning tools and techniques. 2nd edition* Morgan Kaufmann, San Francisco.
- [3] *WekaWiki* – <http://weka.sourceforge.net/wiki/>
- [4] *WekaDoc* – <http://weka.sourceforge.net/wekadoc/>
- [5] *Ensemble Selection on WekaDoc* – http://weka.sourceforge.net/wekadoc/index.php/en:Ensemble_Selection
- [6] *Extensions for Weka's main GUI on WekaWiki* – http://weka.sourceforge.net/wiki/index.php/Extensions_for_Weka%27s_main_GUI
- [7] *Adding tabs in the Explorer on WekaWiki* – http://weka.sourceforge.net/wiki/index.php/Adding_tabs_in_the_Explorer
- [8] *Explorer visualization plugins on WekaWiki* – http://weka.sourceforge.net/wiki/index.php/Explorer_visualization_plugins