

Introduction to Weka

Objectives of this Class

1. Learn to Use the Weka software tool

Weka

The Weka workbench is a set of tools for preprocessing data, experimenting with data-mining/machine-learning algorithms, and comparing the performance of different methods. Weka also provides a Java class library that enables one to use the Weka filters and classifiers in their own programs and modify them as desired.

1. Although there are other Weka interfaces for advanced users, we will use the Explorer interface for most of our work.
2. Go to <http://www.cs.waikato.ac.nz/ml/weka> and download Weka to your laptop. You should download the stable version for the 4th edition of the textbook (**weka-3-8-6**). Make sure you can invoke Weka — do not use the “with console” option. You should get a screen that displays a bird and offers a choice of four graphical user interfaces. Click on **Explorer**.

ARFF data files

An **ARFF file** (Attribute-Relation File Format) is a standard way of representing machine learning data sets as flat files (no relationships among instances). Weka works with ARFF files.

- Lines beginning with % are comments.
- Lines beginning with @ define the relation and then its attributes.
 1. The relation is defined first using the command **@relation:**
 - @relation <relation-name>
 2. Then each attribute is defined using the command **@attribute:**
 - @attribute <attribute-name> {<set-of-comma-separated-possible-attribute-values>}
 - @attribute <attribute-name> numeric
 3. Then the data values are given:
 - @data
 - set of data instances, given as comma-separated values, with each data instance on a separate line

Using Weka: Tutorial by Sandra Carberry

1. Weka Preprocessor: Loading and Examining Data

- (a) Often data is in an Excel spreadsheet, which can be converted to a CSV file (comma-separated file) which can in turn be converted to an ARFF file in Weka.
- (b) Open the Excel spreadsheet *Mushroom-data-23.xls* which can be found on Canvas under Assignment-1. Save the file on your laptop; you might want to put it in a subfolder of the *data* folder that is created as a subfolder of the *weka-3-8-4* folder that was created when you downloaded Weka. (Notice that the Weka *data* folder already contains some data files that we will use during the course.)
- (c) Open the file *Mushroom-data-23.xls*. Let us examine the structure of *Mushroom-data-23.xls*:
 - The first row gives the attribute names.
 - Each subsequent row represents an instance, with the value for each attribute given in the respective column.
- (d) To convert to a csv file, click on “Save-as”, select “CSV” as the file Format, and save (in this case as *Mushroom-data-23.csv*). This saves the file as a comma-separated csv file, though if one opens the file, it still looks like an Excel spreadsheet. Note that you now have both a csv file and an Excel file named *Mushroom-data-23*.
- (e) To convert the csv file to ARFF:
 - i. Invoke the Weka Explorer GUI
 - ii. Select “Open File” under “Preprocess”, move to the folder in which you stored the CSV file *Mushroom-data-23.csv*, set the file type to CSV, and select the file *Mushroom-data-23.csv* as the file to open. Click on *Open*. You should be opening the file that you saved as a csv file.
 - Weka automatically changes the file to ARFF format.
- (f) The csv file does not specify which attribute is the class attribute. You can specify the class attribute by clicking on “Class” in the middle of the right side and selecting the attribute that should serve as the class. Select the attribute **Status** as the class attribute — it can take on the value *e* for edible or *p* for poisonous.
- (g) By clicking on one of the attributes on the left, you will see a histogram that shows how often each of the two values of the class occurs for each value of the selected attribute.
 - Note that if you select the class attribute itself (in this case **Status**), the histogram shows how often each of the classes occurs in the data. The table on the right above the histogram enables you to identify what the histogram colors mean — for example, the first row of the table shows 668 instances of Status=*p* which correlates with the first column in the histogram; note that the first column of the histogram is labelled as having 668 instances.
- (h) On the left side, select *ring-type* as the attribute (but keep *Status* as the Class value). On the right you see a histogram showing the distribution of values of the *Status* attribute for the different possible values of the *ring-type* attribute.
- (i) Questions-1:
 - i. How many instances are there in the data file?
 - ii. How many different values can *ring-type* take on?
 - iii. Which values of the *ring-type* attribute result in only edible mushrooms?

2. Weka Preprocessor: Saving and Viewing ARFF Files

- (a) Save the file in ARFF format. To do this, click on “Save” in the Weka Explorer; in the window that appears, set the file type to “Arff data files”, give the file a name (in this case, set the name as *Mushroom-data-23.arff*), and then click on “Save”. Weka has stored the file in ARFF format so that you can use it again in the future. Note that the Weka data files stored in the data subfolder of the Weka folder are stored in ARFF format.
- (b) Questions-2:
 - i. Use a text editor (such as TextEdit) to view the ARFF file representing the mushroom data. (This **MUST** be an editor that does not format the data — editors such as Word will not work.) What does the first line tell you?
 - ii. What do the next 17 lines tell you?
 - iii. What do the rest of the lines tell you?

3. The Classifier: Now we want to try using a classifier.

- (a) Click on *Classify*.
- (b) Click on *Choose* under *Classifier* and then click on the symbol to the left of *Trees* and then click on *J48*. This will place *J48* as the name of the classification method shown to the right of *Choose*. (J48 is the Weka name for a decision tree classifier based on C4.5, a well-known decision tree algorithm.)
- (c) Click on the method that is listed next to *Choose* — it should be *J48*. Click on *More* to get information about the method that will be used. In the future, you will find this information helpful. Then close the window, click on *Cancel*, and return to the Explorer.
- (d) Immediately below *More options* on the left side of the Explorer screen, there is a line that gives an attribute name preceded by “(Nom)”; this attribute is probably not the class attribute that you want. Click on the arrow to the right, and select *Status* as the attribute to be used as the class attribute.
- (e) Click on *Use training set* under *Test options* on the left side of the Explorer window. This means that the classifier will be built and tested on the same set of data.
- (f) Click on *Start* to build the classifier. The results appear on the right side of the Explorer screen. Scroll up to where it says “*J48 pruned tree*”; you will see a textual description of the tree. Look at this and get an idea of what the tree looks like.
- (g) To actually visualize the tree, do the following. Right click on the last line on the left side of the screen under *Result list*, and select *Visualize tree*. A new window will appear with a graphical view of the decision tree that correlates with the textual description of the tree. Enlarge the window and then right click on an empty area of the window; select *Fit to Screen* from the menu that appears.
- (h) Questions-3:
 - i. Draw the first three levels of the decision tree that was developed.
 - ii. Go back to the results screen and scroll through it to answer the following two questions. How many instances were classified correctly? How many incorrectly?
- (i) Now we want to remove the column for the attribute *Odor* and see what happens if *Odor* is not available as an attribute.
 - i. Go back to the Preprocessor by clicking on *Preprocess* at the top of the Explorer window. To remove an attribute, one can click on the square box next to the name of the attribute and then click on *Remove*. Do this to remove the attribute *Odor*.

- ii. Now also remove the attributes *ring-type* and *habitat*. Save the revised ARFF file as *Mushroom-data-23-revised.arff* in case you want to retrieve it again.
- (j) Invoke the same classifier on this revised data — make sure that you have reset the Class attribute as *Status*. Visualize the resulting tree. Enlarge the window displaying the tree; then right click on empty space in the window and select *Fit To Screen* from the menu that appears. For each leaf node, the class value for that leaf node is given along with one or two numbers. If there is only one number, that tells how many instances reached that leaf node and were classified correctly; if there are two numbers, the first tells how many of the instances reached this leaf node and the second number tells how many were classified incorrectly.
- (k) **Questions-4:**
 - i. What attribute is at the root of the decision tree?
 - ii. Consider the following path in the decision tree: *ring-number=o*, *gill-spacing=w*, *bruises?=t*. What is the class value assigned to instances that follow this path?
 - iii. How many paths in the decision tree lead to a leaf node where some instances are classified incorrectly?
 - iv. Go back to the results screen to answer the following questions. How many instances were classified correctly? How many instances were classified incorrectly?

4. **The Confusion Matrix**

- (a) At the bottom of *Classifier output* is a matrix that is called a *confusion matrix* which we shall refer to as *C*. If there are n possible classes, then the confusion matrix has n rows and n columns, one for each possible class. The entry $C_{i,j}$ gives the number of data items whose correct class is i and which were classified by J48 as class j
- (b) **Questions-5:**
 - i. How many instances are incorrectly classified?
 - ii. What does the diagonal of the confusion matrix tell you?
 - iii. Which class (p or e) did the classifier get wrong most often?

5. The Classifier: Dividing Data into Training and Test Sets

- (a) Instead of training and testing on the same data set, we can ask Weka to hold out part of the data set (ie., not use it to train our classifier) and use it instead as a test set. To save part of the data set for testing, click on the radio button to the left of *Percentage split* under *Test options*, and enter the number 50 as the %. (Make sure that the Class attribute is still set to *Status*.) Run the Classifier and look at the results.
- (b) Now do this twice more, once with 35 and once with 5 as the % for the split. Look at the results and answer the following questions:
- (c) **Questions-6:**
 - i. How many instances were used for training when there is a 50% split? How many for testing?
 - ii. What was the percent success rate when there is a 50% split?
 - iii. How many instances were used for training when there is a 35% split? How many for testing?
 - iv. What was the percent success rate when there is a 35% split?
 - v. How many instances were used for training when there is a 5% split? How many for testing?
 - vi. What was the percent success rate when there is a 5% split?
 - vii. What is the error rate under each of the different splits?
 - viii. What is causing the differences in classification error rate under the different splits?

6. The Classifier: Viewing the Output

- (a) You can also see how the classifier classifies the individual instances in the test set. Let us examine how to do this when only 5% of the data is in the test set (so that the output is not huge).
- (b) Set the split at 95% in the training set and thus 5% in the test set.
- (c) Click on *More Options*; then click on the box labelled *Choose* next to *Output predictions*, select *plain text* and then click on *OK* at the bottom.
- (d) Run the classifier again, and examine the classifier output. Note that you can now see how each instance in the test set was classified, and the ones that are incorrectly classified are noted by a + sign in the error column.

7. The Classifier: Rerunning Models

- (a) Note that on the lower left side of the Explorer window, there is a *Result-list* with an entry for each run of the Classifier. If you click on one of these, you go back to the results for that run. Try it to see that this is the case.
- (b) You can also save a model for future use or reload a previously saved model. Right click on one of the models in the *Result-list*, save it, and then reload it into Weka. Note that only the model is loaded, not the results of testing the model.
- (c) You can also save the results from testing a particular model, and then go back and view the results using a text editor.

8. The Classifier: Using a New Test Set

- (a) You might want to run a model on a new test set of data instances. This can be a bit tricky, since the test set must be in **EXACTLY** the same format as the original training set. Let us explore the problems involved in doing this.
- (b) Now you want to run your model on a new test set. Open the Excel spreadsheet *Mushroom-data-23-test.xls* which can be found on Canvas.
www.cis.udel.edu/~carberry/CISC-483-683. Save this file on your own machine and name it *Mushroom-data-23-test.xls*. Convert the file to CSV format and save it. Then load the file into Weka and save it in ARFF format as *Mushroom-data-23-test.arff*.
- (c) Reload the original ARFF file *Mushroom-data-23.arff* and build a model using the J48 classifier. (Make sure that you have loaded your original ARFF data file, not the revised one that you saved after removing several attributes.) Make sure that your class attribute is set to *Status*.
- (d) Under *Test options*, click on the radio button to the left of *Supplied test set* and then click on *Set* to the right of *Supplied test set*. Click on *Open file* in the small window that appears, select your ARFF file *Mushroom-data-23-test.arff*, and open it as the file that will be used as test data. Set the Class listed in the small window as *Status* and then click on *Close*.
- (e) Make sure that your classification attribute is still set to *Status*, and then click on *Start*. You should get an error message saying that the training and test sets are not compatible.
- (f) Let's look at the problem. Use a text editor (such as TextEdit) to view the ARFF file *Mushroom-data-23.arff* that was used for training and the ARFF file *Mushroom-data-23-test.arff* that was used for testing, and compare them. What differences do you see?
- (g) The problem is that the training and testing data files must have the same set of possible values for the attributes and these possible values must be listed in the same order in the attribute definition. Copy the ARFF file *Mushroom-data-23-test.arff* into file *Mushroom-data-23-test2.arff* (*Mushroom-data-23-test2.arff* will also be an ARFF file); then edit the ARFF file *Mushroom-data-23-test2.arff* so that it has the same set of possible values for the attributes as the ARFF training file *Mushroom-data-23.arff* and these possible values are listed in the same order in the attribute definition. (The easiest way to do this — since it must be perfect — is to cut the attribute definitions from file *Mushroom-data-23-test2.arff* and then paste the attribute definitions from file *Mushroom-data-23.arff* into file *Mushroom-data-23-test2.arff*) Then save *Mushroom-data-23-test2.arff*.
- (h) Set the test file as the edited *Mushroom-data-23-test2.arff*. Make sure that the Class in the small window (the one where you opened the test file) is set to *Status*. Again click on *Start*. If you have edited the test file correctly, you should get a successful test run on the 10 instances in the test file.
- (i) **Questions-7:**
 - i. How many instances were classified correctly?
 - ii. Draw the confusion matrix.