Tutorial : Introduction to Data Mining with WEKA

Basic notions in machine learning/classification

Slideshow (available from MD Devignes's homepage).

Hands-on with WEKA

I. Installation of WEKA software

- WEKA (java software) is already installed on the University computers
- For installation on your personal laptop download WEKA from https://waikato.github.io/weka-wiki/downloading_weka/ . Software is free and can be downloaded for windows, linux or mac.
- Launch the program, version 3.8.6 or above, and select the mode « **Explorer** »
- Other modes will not be explored during this tutorial but there are online tutorials and a MOOC with 91 videos if you are interested.
    - WEKA Experimenter Tutorial
    - WekaMOOC

II. Exploration d'un jeu de données : iris.arff

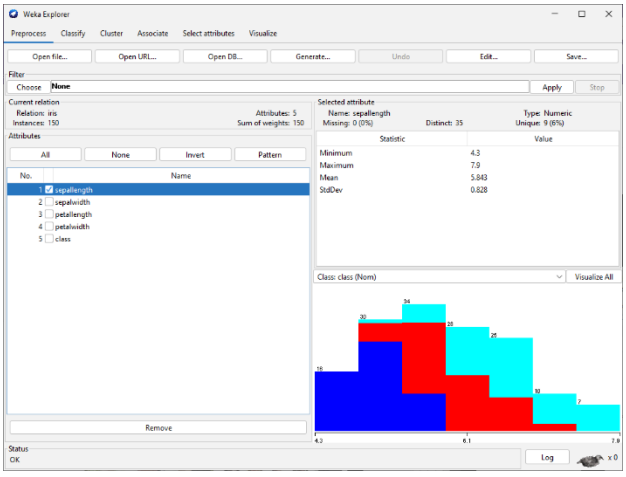
1. Opening the file

Tab **Preprocess** and button **Open file…** -> find the directory in which WEKA is installed, open the weka directory, then data directory and select **iris.arff**, click on **open**

NB : arff (attribute relation file format) is the common input format for all programs used by weka. However, you can also open a csv file (select this file format rather than arff in your file manager) or a file in many other formats and weka will convert them into arff files.

2. Exploration of the content

- Explore the various panels of the interface.
- Left panel : how many attributes are available in this dataset ? What happens in the right panel when you select an attribute in the left panel ?
- Right top panel : What do you learn about each attribute ?
    - Instance count in each class
    - Type of attributes (numeric or nominal)
    - Stats on the values taken by the attributes
    - Missing values.
- Right bottom panel : instances are colored by class and the distribution of each attribute is displayed as histograms for each class. What information can you infer from that?
- Switch to the tab **Visualize** to see co-variations of attributes pairwise.



- Adjust **PlotSize** if the display does not show all the panels on your window. Click on **update** to apply the changes
- Adjust "**PointSize**" if the dots are too small on your display. Click on **update** to apply the changes.
- The "**Jitter**" option introduces some artificial noise to the coordinates of the plotted points in order to spread the data out a bit (to allow visualization of points that could be superposed by chance). Try it.
- Keep **Colour: class (Nom)** (Nom is for 'nominal')
- What do you learn from these visualization results? Can you identify any pair of highly correlated attributes?

III. Non supervised classification of the iris.arff dataset

1. Performing the clustering with K=2

- From the **Visualize** tab, go back to the main window by clicking on the **Preprocess** tab.
- Switch now to the **Cluster** tab.
- Click on the **Choose** button to scroll all the clustering algorithms ('clusterers') proposed by Weka.
- Select the last one: **Simple Kmeans**.
- Check the parameters used by Simple Kmeans by **clicking in the textfield** describing the command that will be used to perform the clustering. A window opens with nearly 20 parameters. We will explain only two of them due to lack of time.
  - **distanceFunction** : by default the EuclideanDistance. The **Choose** button allows to select other distances implemented in Weka.
  - **numClusters** : corresponds to the « **K** » parameter introduced in the slideshow. K is the number of centroids selected at the first step of the algorithm. It is also the final number of clusters. Keep the by default value : K = 2 but we will try other values later.
  - Close the parameter window by clicking on **OK** -> this ensures that the selected parameters will be applied in the next run of simpleKmeans.
- Because clustering is a non supervised classification, on should ignore the Class attribute in the dataset. Click on the bar **Ignore Attributes** and select **Class**. Apply by clicking on **Select**.
- Click on **Start** to run the algorithm. The result is displayed immediately (the task is easy with this dataset) in the right panel. The left bottom panel keeps track of the history of your session, one row for each run.
- Analyze the result (displayed in the right panel)
  - Copy-Paste the text result in a text file to save the results of your work and save it in you working directory.
  - Try to understand the three sections: Run information, Clustering model (full training set), Model and evaluation on training set.
  - Note the value obtained by the "Within cluster sum of squared errors" score. (This value will be used to plot the cohesion curve and determine the optimal value of K in the next section)

  - Note the coordinates (4 values, one for each attribute) of the initial centroids and see the difference with the coordinates of the final cluster centroids. What information provide these coordinates (the final ones)?
  - Note how many instances there are in each cluster.
- Exploration of the clusters (starting from the Result list in the left bottom panel)
  - Right-click on your job and go to **Visualize cluster assignments**. This opens a new window where you see all the instances of the dataset colored by cluster. Instances are displayed according to two attributes selected by default (X and Y assignments can be changed). Colouring is by cluster (2 colors as K=2). What happens if you color by class ? What can you conclude on the distribution of the iris classes among the two clusters ?
  - By clicking on **Save** you can also save the detailed clustering results as a .arff file containing the cluster assignment of each instance of the dataset. The .arff file can be opened as a text file by any simple text editor.
  - NB: if nothing appears in the visualization, check the item "**Store clusters for visualization**" in the left top panel. It should be activated before execution of the clustering algorithm.

2. Determination of optimal cluster number by the elbow method

- Change K value as described above, run again **simpleKmeans**, and record the **Within cluster sum of squared errors** in a spreadsheet file for each value of K (Test K=2, 3, 4, 5).
- Build the cohesion curve.
- Note the elbow and determine the optimal cluster number.

3. Evaluation of clustering with class information

- The class information of the iris dataset can be used to evaluate the clustering. Here, the class is the known species of each instance of the dataset. The purpose of evaluating with class is here to check if the clustering algorithm is capable to reflect the species distribution. If not, this means that the provided information is not sufficient for the clustering algorithm to recognize all three species. Maybe one relevant attribute is missing. This evaluation can be useful to complete a dataset and to identify what attributes or combinations of attributes are relevant to discriminate between known species.
- In the left top panel, check the item « **classes to clusters evaluation** »
- Run again the simpleKmeans algorithm starting with K=2. A confusion table appears now at the end of the report in the right panel. Observe the table and make conclusions. Note the number and % of incorrectly classified instances.
- Repeat this execution for K = 2, K = 3, K = 4, K = 5 and plot the % of incorrectly classified instances as a function of K.
- Interpret and conclude.

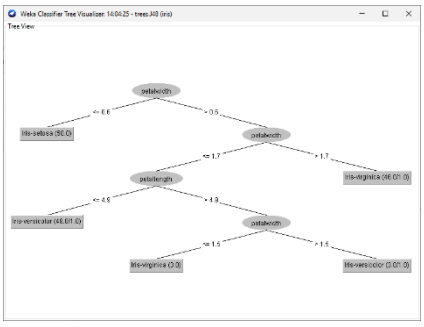
IV. Supervised classification of the iris.arff dataset

1. Use of decision tree J48

- Go back to the main window with the same dataset by clicking on the **Preprocess** tab.
- Swith to the **Classify** tab.
- Scroll the menu of the **Choose** button: this gives you an idea of the diversity of the Classifiers algorithms implemented in Weka (Weka is a very complete toolbox for datamining). Go to the last directory named "**tree**", open it by clicking on the '**>**' sign on the left and select **J48**.
- As before, open the windows with all parameters by **clicking in the text field** describing the main parameters
    - Several parameters are easy to understand and tune like for instance **MinNumObj** (2 by default), **unpruned** (False par défaut) which means that the tree is pruned to avoid getting too detailed branches which could lead to overfitting. We will not explain all the parameters due to lack of time.
    - Keep **default parameters** and close the panel by clicking on **OK**
- In the left top panel, check that the **Cross-validation** button is activated and keep the default number of folds (**10**)
- Click on **Start** to launch the execution.

2. Visualization and evaluation of the classifier

- Inspect the results in the right panel (copy-paste as before in order to keep a track of your exploration). Results are divided in three parts: Run information, Classifier model (full training set), Stratified cross-validation. A text representation of the decision tree is provided. Note the number of leaves and the size of the tree (total number of nodes). Explore all the statistics computed from the cross-validation, including the % of incorrectly classified instances, Kappa statistics, etc. Absolute error is the difference of predicted versus real value. Relative error is a measure in percent compared to the real value. See the table providing a certain number of metrics according to the class. See finally the confusion matrix.
- Keep track of the **F-measure per class and average** and of the **ROC area value** which is independent of the class.
- Visualize the tree as a figure from the **Result list** in the left bottom panel. Right-click on the appropriate run and normal click on **Visualize tree**. You can keep the figure by capturing a screenshot and pasting the results in a text document.



V. Supervised classification with other datasets : Fight-HF fibrotic datasets

Datasets pertaining from experiments described in the seminar and in the following paper: Bresso Emmanuel et al. (2022) Inductive database to support iterative data mining, J Biomed Informatics 135:104212 *doi:10.1016/j.jbi.2022.104212*

Download the two datasets from MD Devignes's homepage.

1. Dataset exploration

466 patients, 40 nominal attributes including

- the class attribute (fibrotic or healthy)
- 39 frequent itemsets of proteomic markers

There are two datasets for the same data

- One is unbalanced, reflecting the different counts of healthy versus fibrotic patients: 393 versus 73.
- One is balanced, with a corrective probability associated to each sample leading to the equivalent of 233 healthy and 233 fibrotic. This dataset was produced by WEKA using the **Filter** functionality in the **Preprocess** tab and selecting the **ClassBalancer** filter under the **Choose** menu (**Filters > Supervised > Instances > ClassBalancer**).

2. J48 classifier with minNumObj = 2

Run **J48** as before on both datasets consecutively to see the differences in the results. Copy-paste the stats report to save the results of your tests. Visualize the trees and save a screenshot.

Compare (for the unbalanced and balanced datasets):

- the complexity of the tree (number of leaves and total number of nodes),
- the F-measure per class,
- the ROC area
- and the confusion matrix.

What can you conclude?

3. J48 classifier with minNumObj = 5

When the trees are too complex (too many nodes and leaves) you can modify the parameters of J48 to get more simple trees. In practice, increase the minNumObj (minimal number of objects in a node) from 2 to 5 for instance.

Repeat the comparison between the two datasets as above.

See the consequences on the tree complexity, on F-measure per class, on ROC area and on confusion matrix.

What can you conclude ?

Which one of these trees is the most similar to the tree presented in the publication ?

Conclusion

Enjoy using WEKA and explore many more functionalities !

Please note: this tutorial is only an extremely simplified introduction and data mining is full of subtleties ! If you are interested, consider taking classes and completing your training !