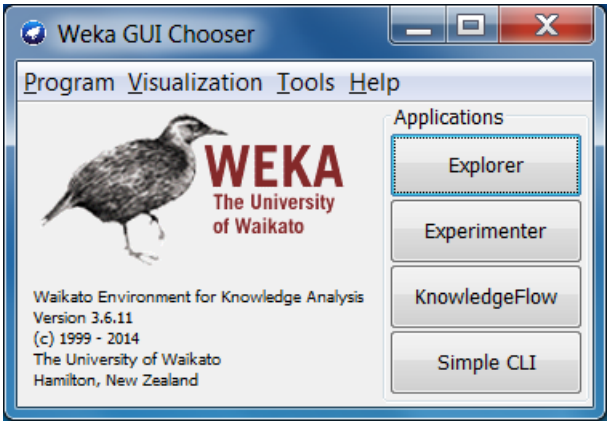
Clustering Iris Data with Weka

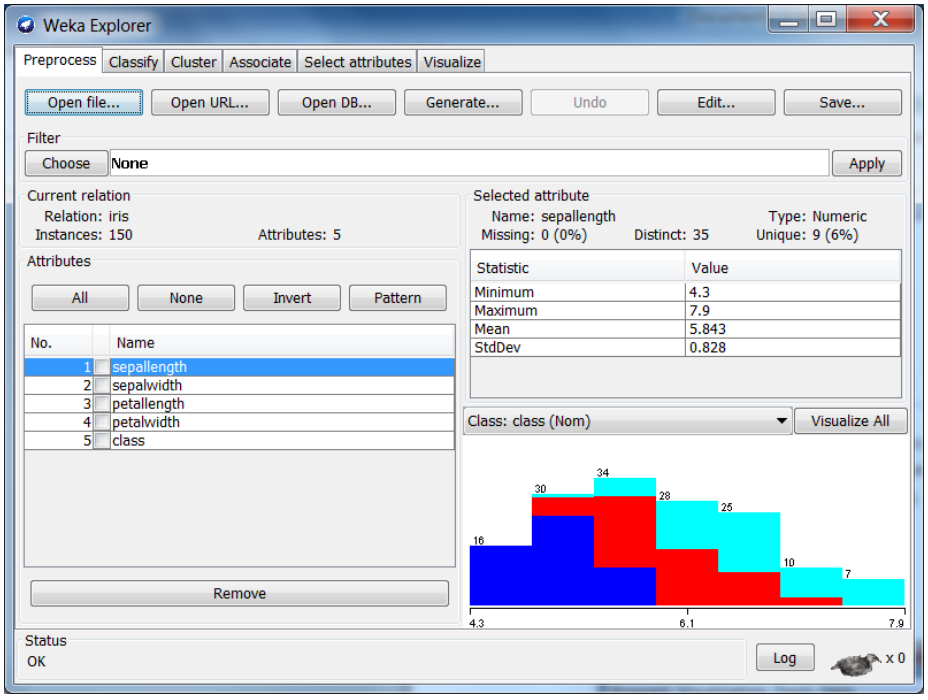
The following is a tutorial on how to apply simple clustering and visualization with Weka to a common classification problem. Beyond basic clustering practice, you will learn through experience that more data does not necessarily imply better clustering.

Loading the Iris Data

- Start Weka
- Press the Explorer Button:



- Download the iris.arff data file, e.g. from http://tunedit.org/repo/UCI/iris.arff .
- In the "Preprocess" tab of the Weka Explorer window, click the "Open file…" button, and select the iris.arff file from your download location. The window should then look like this:

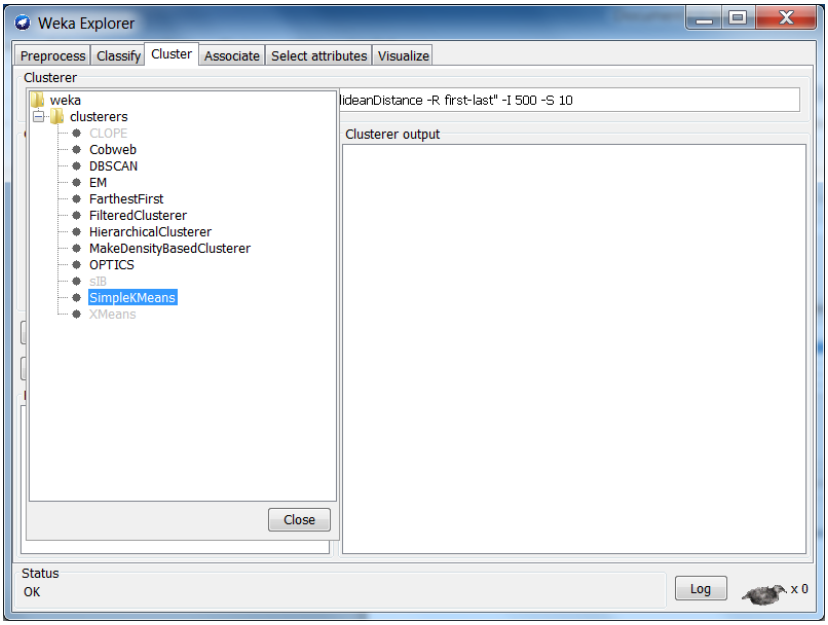


This simple and commonly used dataset contains 150 instances with real valued data for iris sepal and petal lengths and widths. The 5th attribute of the data set is the "class", that is, the genus and species of the iris measured.

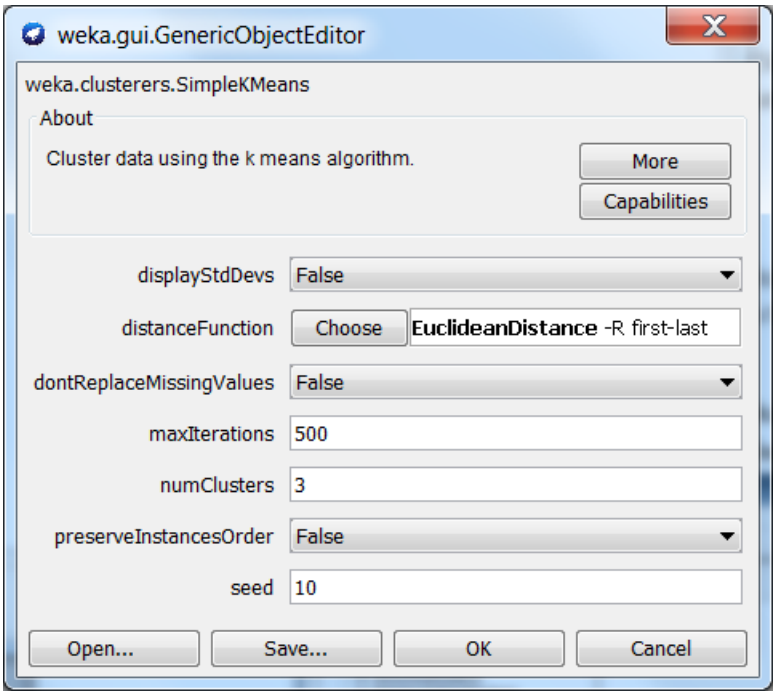
Simple k-Means Clustering

While this dataset is commonly used to test classification algorithms, we will experiment here to see how well the k-Means Clustering algorithm clusters the numeric data according to the original class labels.

- Click the "Cluster" tab at the top of the Weka Explorer.
- Click the Clusterer "Choose" button and select "SimpleKMeans".



- Click the SimpleKMeans command box to the right of the Choose button, change the "numClusters" attribute to 3, and click the OK button.



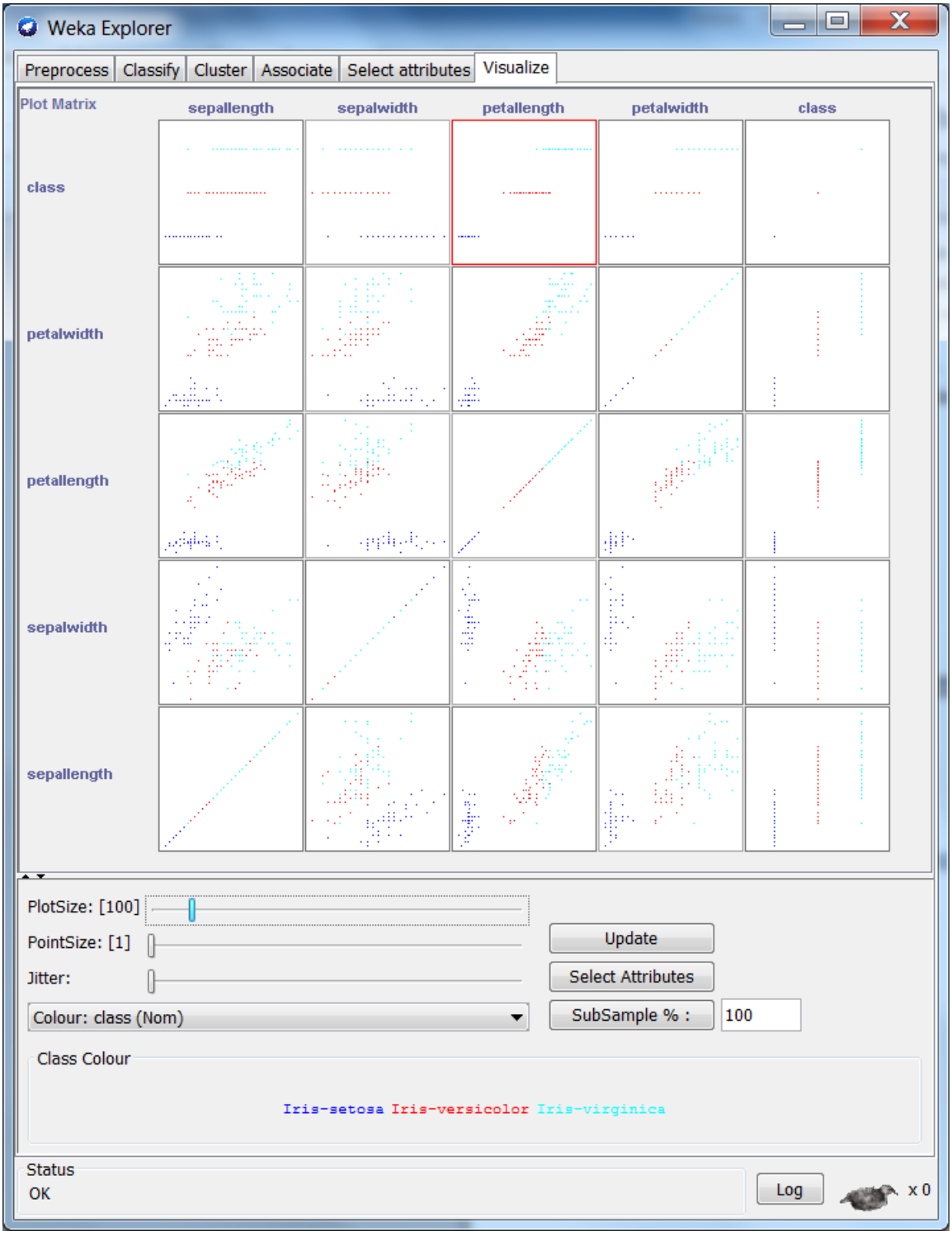
- Under Cluster Mode, select the radio button "Classes to cluster evaluation" which should be follows by "(nom) class" by default.
- Press Start to begin k-Means Clustering and evaluation.
- Note the cluster centroids in the Clusterer output pane.
- What instance percentage is incorrectly clustered? _____

Visualization

In k-Means Clustering, there are a number of ways one can often improve results. One of the most common is to normalize the results in some fashion so that the differences in scale of the numerical attributes do not dominate the Euclidean distance measure. This can be accomplished by linearly scaling the data of each attribute between -1 and 1, or by replacing attribute values with the number of standard deviations each have from the attribute mean value.

For this dataset, we will demonstrate an even simpler approach. Visualization can sometimes help us discern the attributes that best separate the data. Recall that k-Means Clustering assumes non-overlapping, hyperspherical clusters with similar size and density.

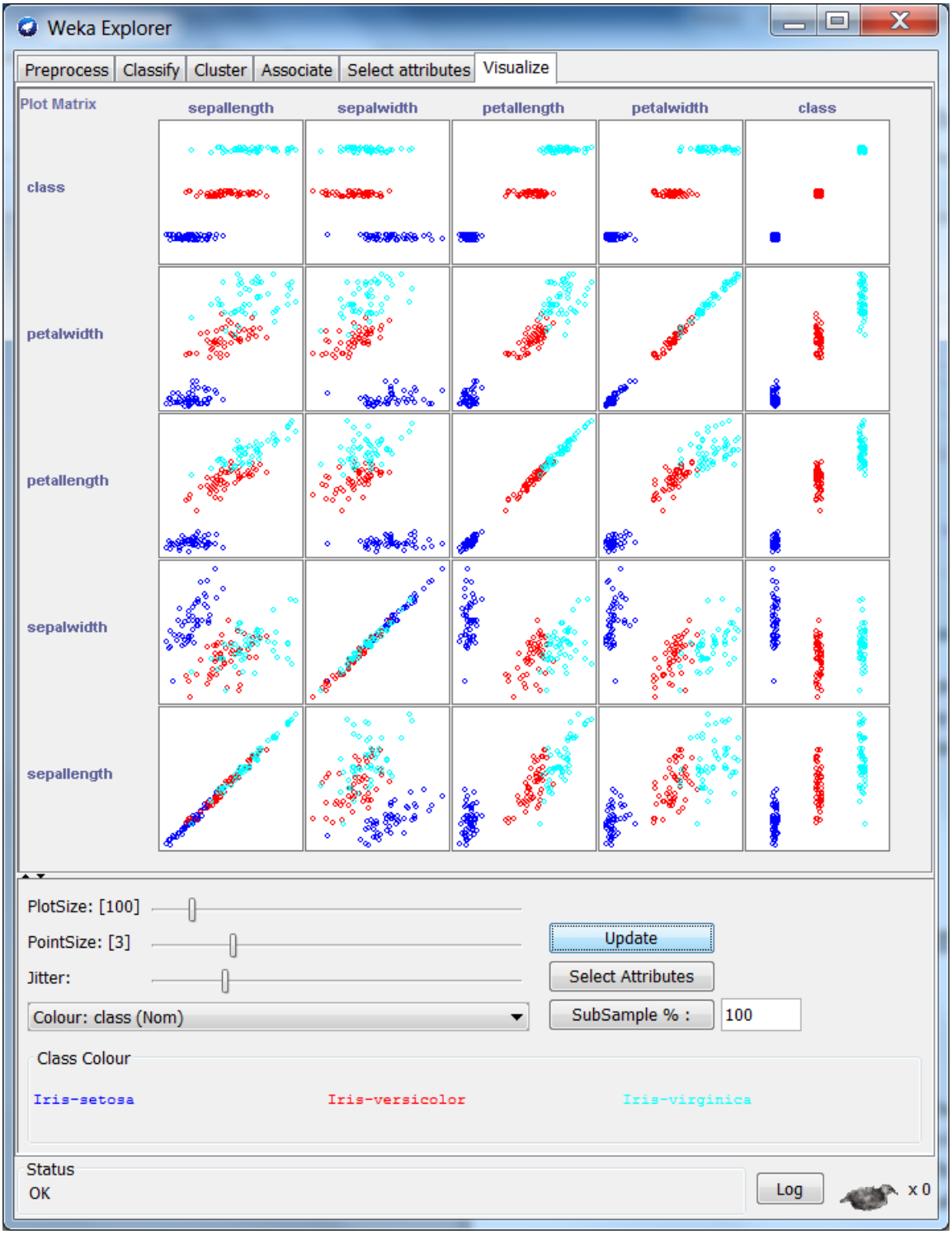
- Click on the “Visualize” tab.



Each subplot (except those paired with “class”) shows the normalized value distribution of a pair of attributes, colored by class.

- Adjust the visualization settings:

- PlotSize increases the size of each subplot.
- PointSize makes the individual points larger. (Set to 3.)
- Jitter randomly shifts the points so that overlapping points can be seen and density becomes more visually apparent. (Set to about 20%.)



Note that different pairs of attributes do better/worse at separating the data classes. Which pair of non-class attributes appears to separate the classes best? _____

Simply k-Means Clustering Ignoring Attributes

- Click on the "Cluster" tab again.
- Click the "Ignore Attributes" button. Use a control-click to toggle the attributes dark (ignored) or light (used). Ignore the two numeric attributes that don't separate as well as the ones you have written above.
- Press Start to begin k-Means Clustering and evaluation.
- What instance percentage is incorrectly clustered using the chosen pair? _____
- Can you further ignore one of these two attributes and achieve the same performance? If not, write "no". If yes, write the attribute that can be ignored: _____

Conclusion

Because k-Means Clustering assumes non-overlapping, hyperspherical clusters of data with similar size and density, data attributes that violate this assumption can be detrimental to clustering performance. Less is sometimes more. This is why the use of visualization tools can be helpful in the best application of clustering algorithms.

Further Reading

Chapter 10 of *Introduction to Statistical Learning with Applications in R* (http://www-bcf.usc.edu/~gareth/ISL/) is an excellent source of further information about clustering algorithms and their application.

Todd W. Neller