

Clustering

UROŠ KRČADINAC

EMAIL: uros@krcadinac.com

URL: <http://krcadinac.com>

Clustering

Clustering belongs to a group of techniques of unsupervised learning. It enables grouping instances into groups, where we know which are the possible groups in *advance*.

These groups are called **clusters**.

As the result of clustering each instance is being added a new attribute – the cluster to which it belongs. The clustering is said to be successful if the final clusters make sense, if they could be given meaningful names.

K-Means algorithm in Weka

FishersIrisDataset.arff

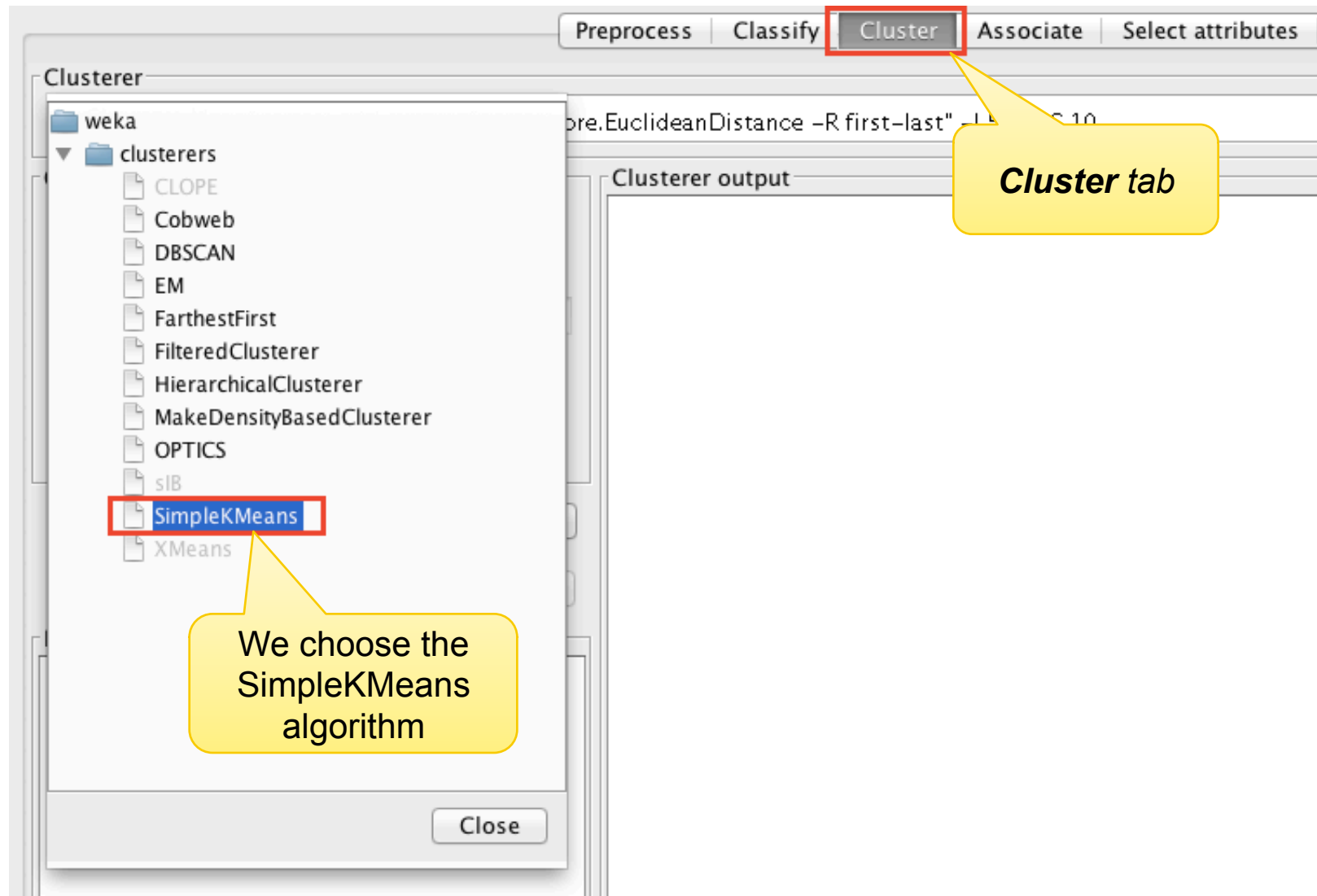
The screenshot shows the Weka software interface with the 'Preprocess' tab selected. The 'Open file...' button is highlighted. The 'Filter' section shows 'None' selected. The 'Current relation' section displays 'Relation: FishersIrisDataset-weka.filters.unsupervised.attribute.Remove-R1' and 'Instances: 150'. The 'Attributes' section shows a list of attributes: 'Sepal Length', 'Sepal Width', 'Petal Length', 'Petal Width', and 'Species'. The 'Selected attribute' section shows 'Name: Sepal Length' with statistics: 'Missing: 0 (0%)', 'Distinct: 35', 'Type: Numeric', and 'Unique: 9 (6%)'. A table of statistics for 'Sepal Length' is shown below:

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

The 'Class: Species (Nom)' dropdown is set to 'Species (Nom)'. The 'Visualize All' button is visible. A histogram of the 'Sepal Length' attribute is displayed, showing three distinct clusters of data points colored blue, red, and cyan. The x-axis ranges from 4.3 to 7.9, and the y-axis shows counts for each bin.

Status: OK

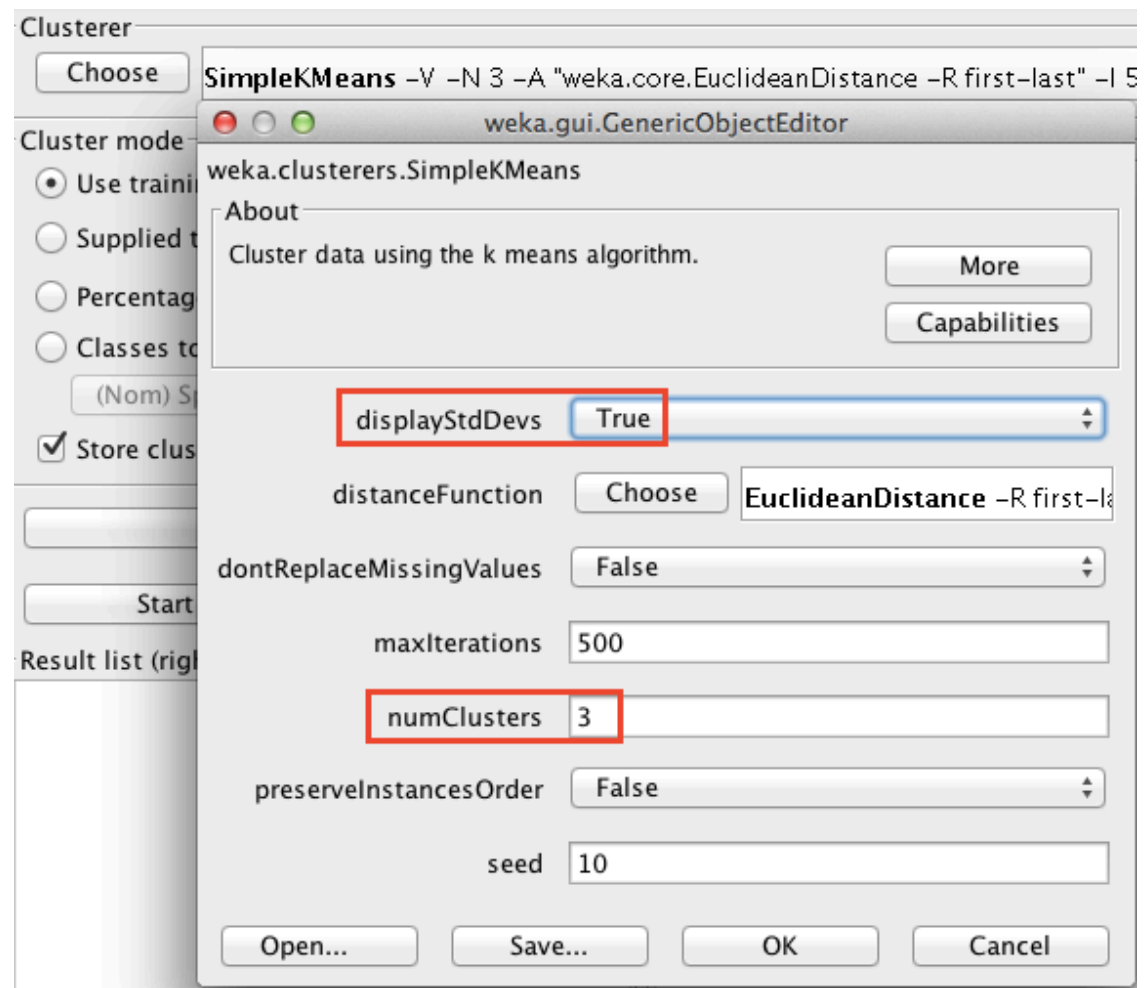
Choosing the clustering algorithm



Parameter settings

numClusters – the number of desired clusters;
we set it to 3 because we have 3 kinds

displayStdDevs – if
true, the standard
deviation will be
displayed



Running the Clustering

The screenshot shows the Weka GUI with the SimpleKMeans clustering algorithm selected. A yellow callout bubble points to the 'SimpleKMeans' button with the text 'Clustering over the imported data'. Another yellow callout bubble points to the 'Ignore attributes' field with the text 'We ignore the **Species** attribute'. The 'Ignore attributes' field contains the text 'Species'. The 'Store clusters for visualization' checkbox is checked. The 'Result list' shows the output of the clustering process.

Clusterer
Choose SimpleKMeans

Cluster mode
☒ Use training set
☐ Supplied test set Set...
☐ Percentage split % 66
☐ Classes to clusters evaluation
(Nom) Species
☒ Store clusters for visualization
Ignore attributes

Result list (right-click for options)
15:07:08 - SimpleKMeans

Run information
Scheme: weka.clusterers.SimpleKMeans -V -N 3 -A "weka.core.Euc
Relation: FishersIrisDataset-weka.filters.unsupervised.at
Instances: 150
Attributes: 5
Sepal Length
Sepal Width
Petal Length
Petal Width

Ignored: Species

Test mode: evaluate on training data

Model and evaluation on training set

of iterations: 6
Within cluster sum of squared errors: 6.982216473785234
Missing values globally replaced with mean/mode

Cluster centroids:

Attribute	Full Data (150)	Cluster# 0 (61)	1 (50)	2 (39)
Sepal Length	5.8433 +/-0.8281	5.8885 +/-0.4487	5.006 +/-0.3525	6.8462 +/-0.5025
Sepal Width	3.0573	2.7377	3.428	3.0821

Results of Clustering

Cluster mode

☒ Use training set

☐ Supplied test set

☐ Percentage split %

☒ Centroids of each cluster and their standard deviations

Result list (right-click for options)

15:07:08 - SimpleKMeans

Clusterer output

kMeans

=====
Number of iterations: 6
Within cluster sum of squared errors: 6.982216473785234
Missing values globally replaced with mean/mode

Cluster centroids:

Attribute	Full Data (150)	Cluster# 0 (61)	1 (50)	2 (39)
Sepal Length	5.8433 +/-0.8281	5.8885 +/-0.4487	5.006 +/-0.3525	6.8462 +/-0.5025
Sepal Width	3.0573 +/-0.4359	2.7377 +/-0.2934	3.428 +/-0.3791	3.0821 +/-0.2799
Petal Length	3.758 +/-1.7653	4.3967 +/-0.5269	1.462 +/-0.1737	5.7026 +/-0.5194
Petal Width	1.1993 +/-0.7622	1.418 +/-0.2723	0.246 +/-0.1054	2.0795 +/-0.2811

Time taken to build model (full training data) : 0.04 seconds

=== Model and evaluation on training set ===

Clustered Instances

0	61 (41%)
1	50 (33%)
2	39 (26%)

Evaluation of Results

The screenshot shows the Orange3 SimpleKMeans interface. On the left, the 'Cluster mode' section has four radio buttons: 'Use training set', 'Supplied test set', 'Percentage split', and 'Classes to clusters evaluation'. The 'Classes to clusters evaluation' option is selected and highlighted with a red box. Below it, a dropdown menu shows '(Nom) Species' and is also highlighted with a red box. A yellow callout bubble points to this dropdown with the text 'Select the attribute which we want to compare the results with.' Below the dropdown is a checked checkbox 'Store clusters for visualization'. Further down are buttons for 'Ignore attributes', 'Start', and 'Stop'. At the bottom left, the 'Result list (right-click for options)' shows two entries: '15:07:08 - SimpleKMeans' and '15:20:38 - SimpleKMeans', with the latter selected. A yellow callout bubble points to this list with the text 'Names of classes which are given to clusters'.

Cluster mode

- ☐ Use training set
- ☐ Supplied test set
- ☐ Percentage split % 66
- ☒ Classes to clusters evaluation
(Nom) Species
- ☒ Store clusters for visualization

Result list (right-click for options)

- 15:07:08 - SimpleKMeans
- 15:20:38 - SimpleKMeans

Clusterer output

	3.428	3.0821		
	0.3791	+/-0.2799		
	1.462	5.7026		
	0.1737	+/-0.5194		
Petal width	1.1995	1.416	0.246	2.0795
	+/-0.7622	+/-0.2723	+/-0.1054	+/-0.2811

Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0	61 (41%)	
1	50 (33%)	
2	39 (26%)	

Class attribute: Species
Classes to Clusters:

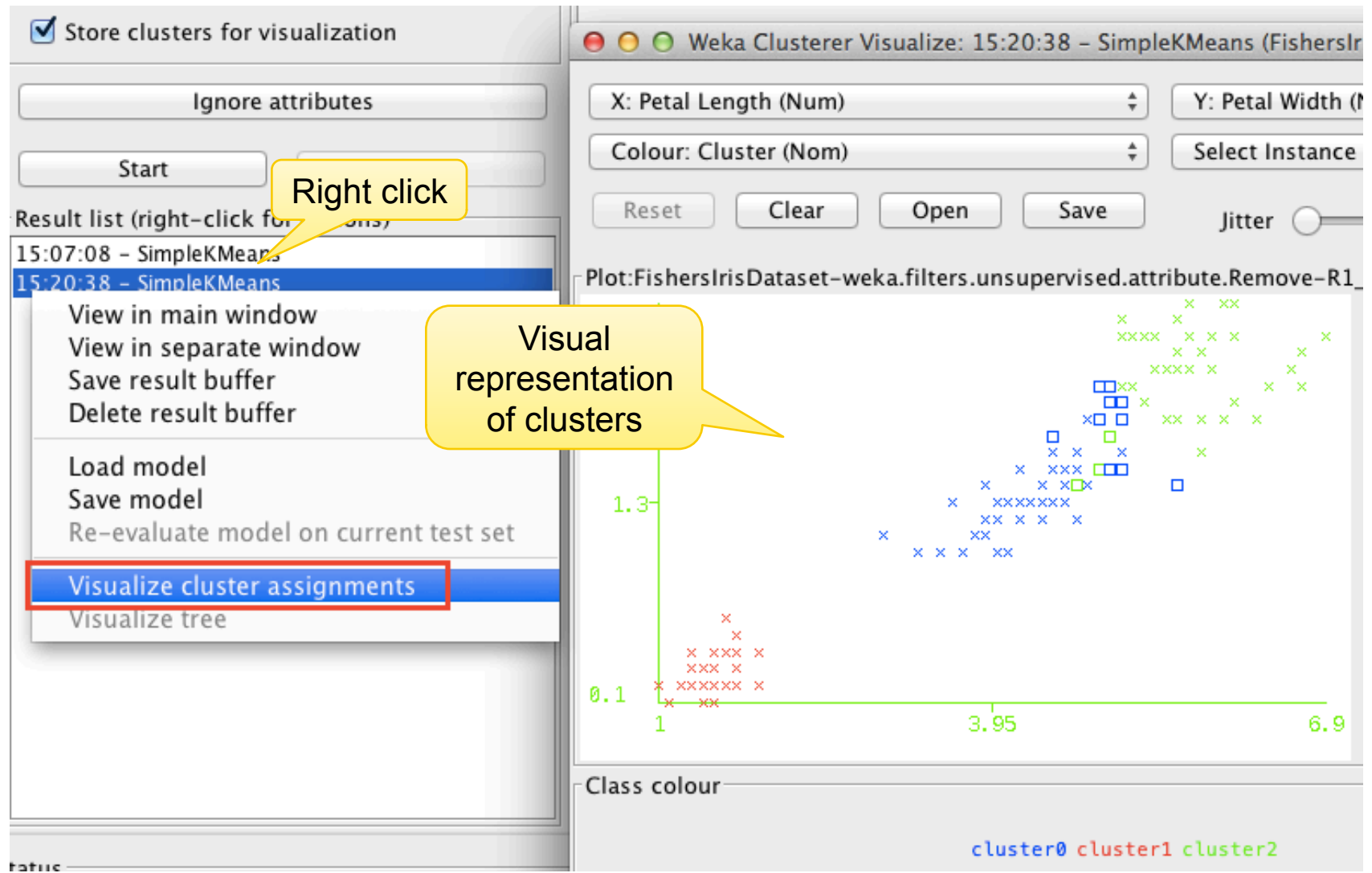
0	1	2	<-- assigned to cluster
0	50	0	setosa
47	0	3	versicolor
14	0	36	virginica

Cluster 0 <-- versicolor
Cluster 1 <-- setosa
Cluster 2 <-- virginica

Incorrectly clustered instances : 17.0 11.3333 %

Which classes are in which clusters

Visualization of Clusters



Was clustering successful?

Within cluster sum of squared error gives us the assessment of quality

Cluster mode

☐ Use training set

It is being counted as the sum of square differences between the value of the attribute of each instance and the value of the centroid of the given attribute

ignore attributes

Start Stop

Result list (right-click for options)

- 15:07:08 - SimpleKMeans
- 15:20:38 - SimpleKMeans

Clusterer output

kMeans

====

Number of iterations: 6

Within cluster sum of squared errors: 6.982216473785234

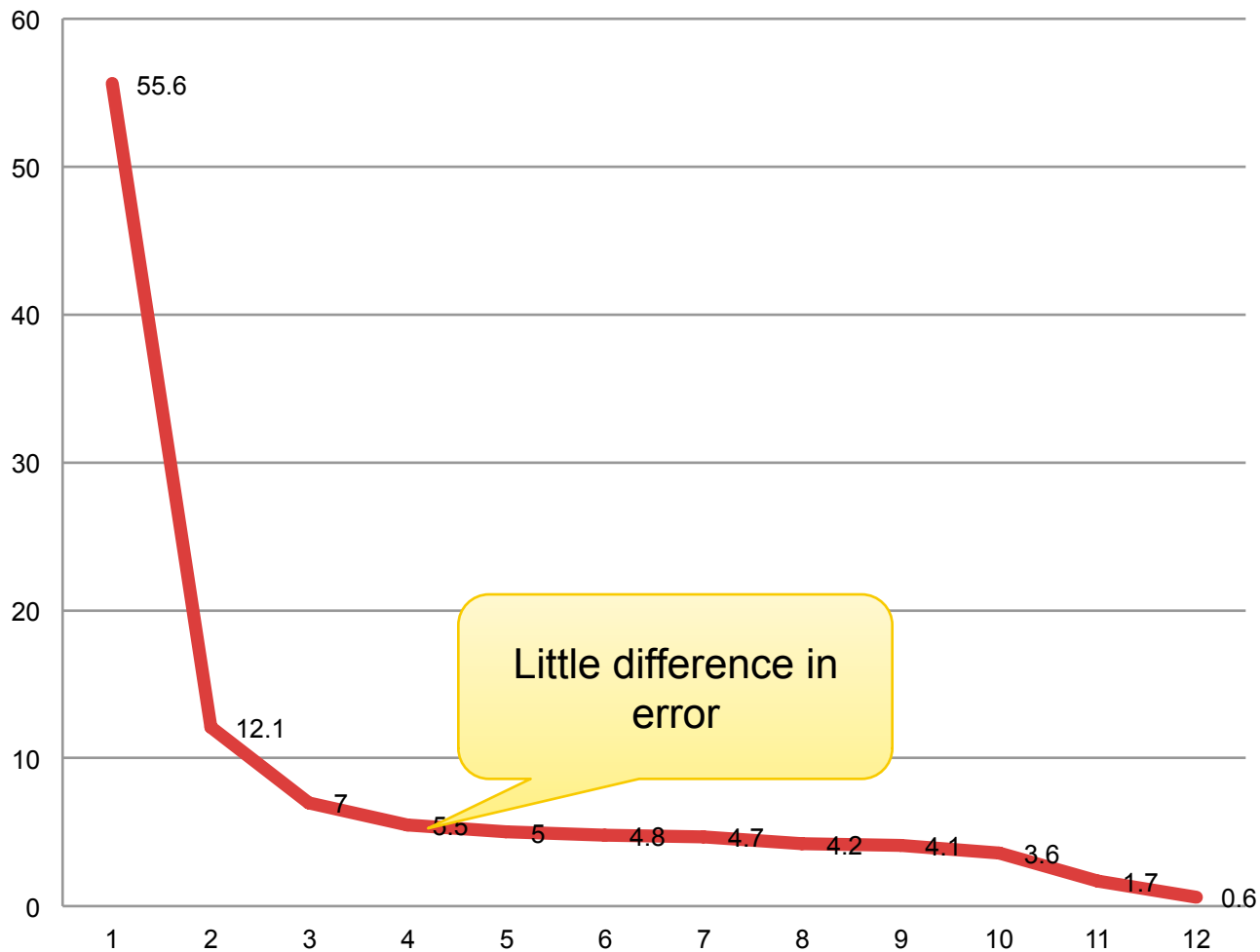
Missing values globally replaced with mean/mode

Cluster centroids:

Attribute	Full Data (150)	Cluster# 0 (61)	1 (50)	2 (39)
Sepal Length	5.8433 +/-0.8281	5.8885 +/-0.4487	5.006 +/-0.2555	6.8462 +/-0.5055
Sepal Width	3.0573 +/-0.4359	2.7377 +/-0.2934	2.9729 +/-0.2527	3.4283 +/-0.3683
Petal Length	3.758 +/-1.7653	4.3967 +/-0.5269	1.4623 +/-0.2659	5.1036 +/-0.5194
Petal Width	1.1993 +/-0.7622	1.418 +/-0.2723	0.246 +/-0.1054	2.0795 +/-0.2811

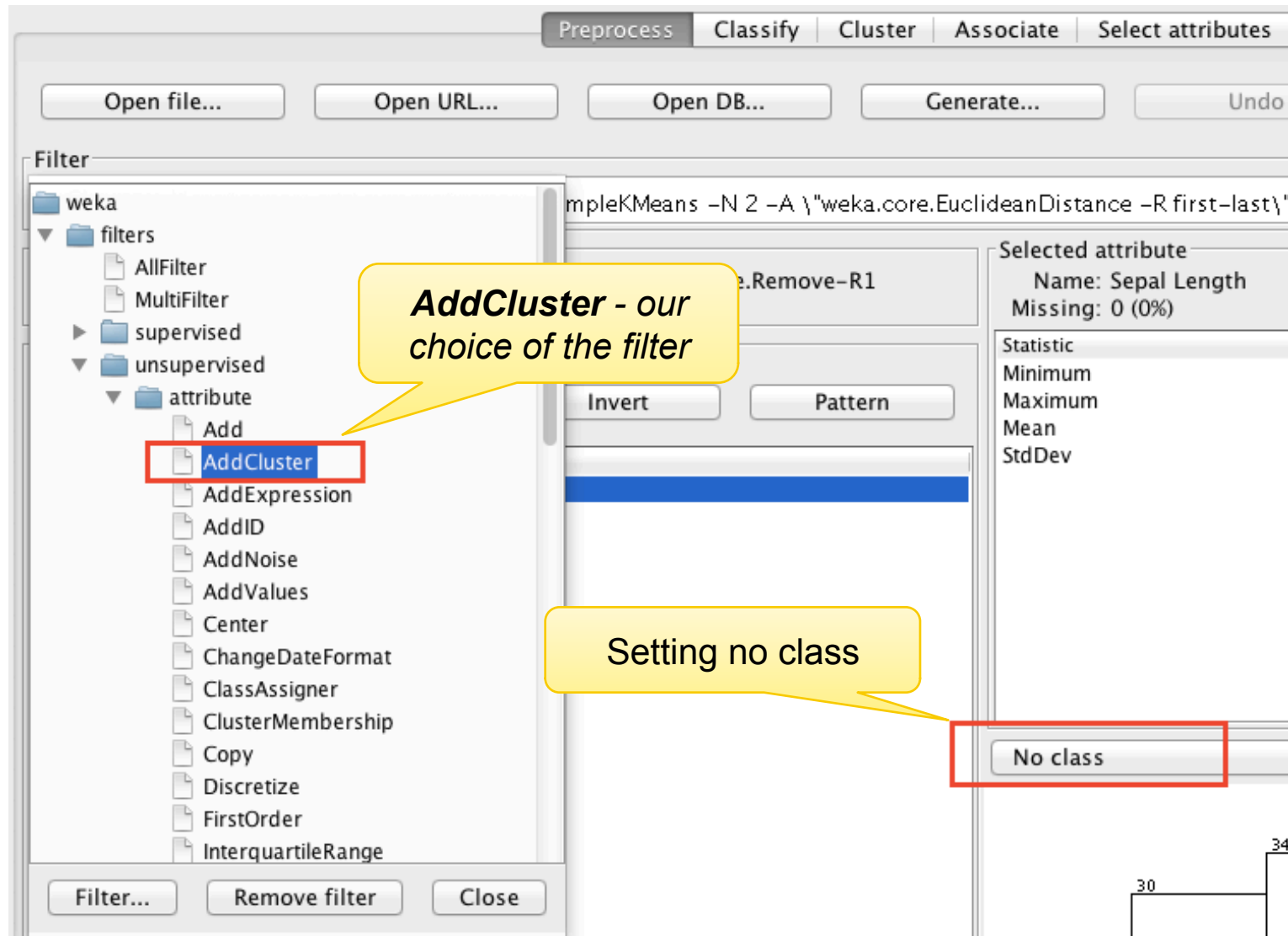
Attribute values for the centroids

How to figure out the number of clusters?



Clusters	Errors
1	55.6
2	12.1
3	7.0
4	5.5
5	5.0
6	4.8
7	4.7
8	4.2
9	4.1
10	3.6
20	1.7
50	0.6

Using Clusters for Classification



Using Clusters for Classification

The screenshot displays the Weka GUI with two windows. The 'weka.gui.GenericObjectEditor' window shows the 'AddCluster' button highlighted with a red box. A yellow callout bubble points to this button, stating: 'We choose the **SimpleKMeans** as the clustering algorithm'. Below this, the 'clusterer' dropdown is set to 'SimpleKMeans', also highlighted with a red box. Another yellow callout bubble points to the 'ignoredAttributeIndices' field, which contains the value '5', stating: 'In terms of clustering, we ignore the attribute 5 (Species)'. The 'weka.clusterers.SimpleKMeans' configuration window is also visible, showing various parameters. The 'numClusters' field is highlighted with a red box and set to '3'.

Filter

Choose **AddCluster** -V "weka.clusterers.SimpleKMeans -V -N 3 -A \"weka.core.EuclideanDistance -R first-last\" -

Current relation: weka.gui.GenericObjectEditor

Relationship: weka.gui.GenericObjectEditor

Attribute: weka.gui.GenericObjectEditor

We choose the **SimpleKMeans** as the clustering algorithm

clusterer: Choose **SimpleKMeans** -V

ignoredAttributeIndices: **5**

Open... Save... OK

weka.clusterers.SimpleKMeans

About: Cluster data using the k means algorithm.

displayStdDevs: True

distanceFunction: Choose Euclidean

dontReplaceMissingValues: False

maxIterations: 500

numClusters: 3

preserveInstancesOrder: False

seed: 10

Open... Save... C

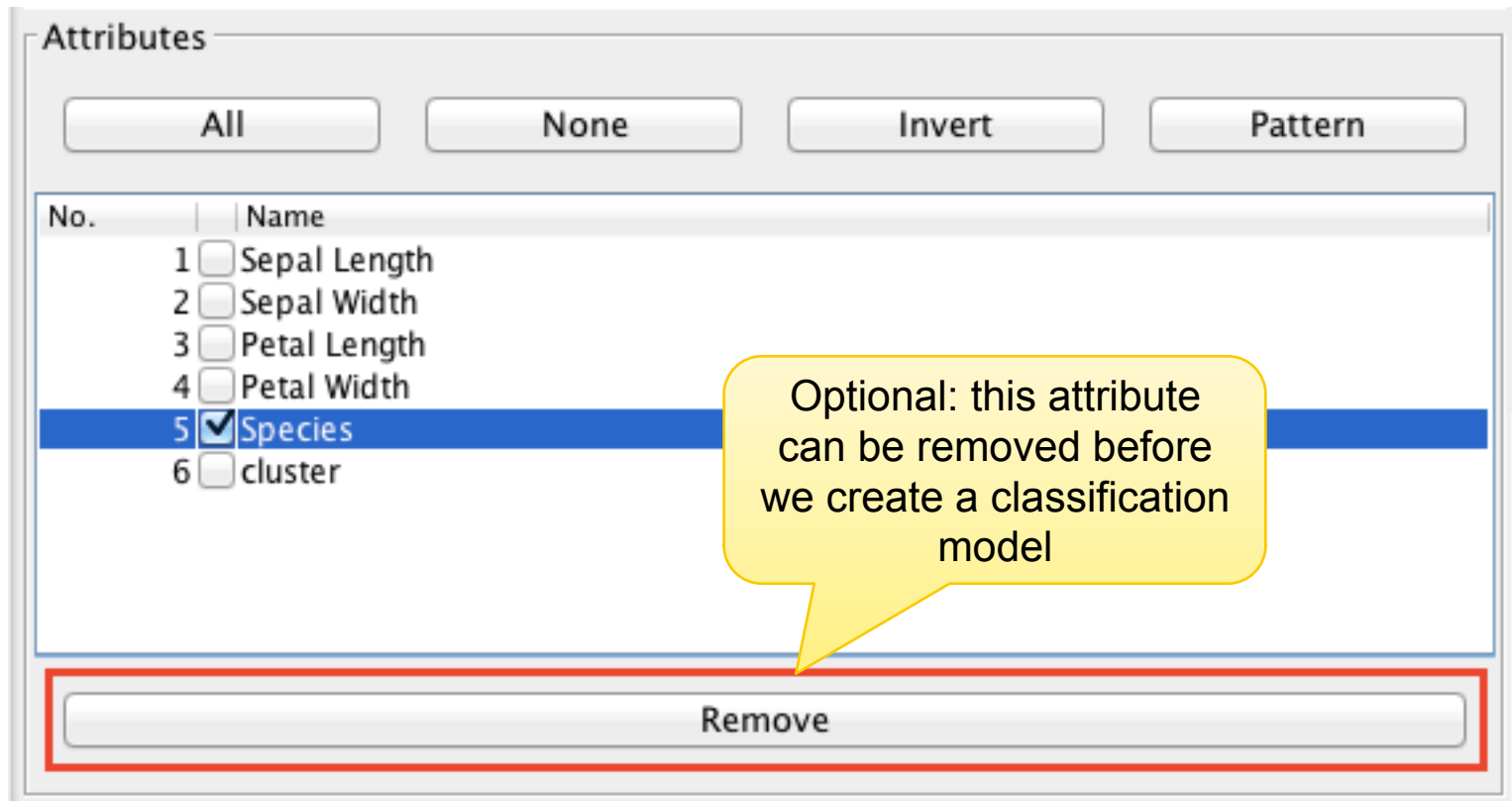
Using Clusters for Classification

The screenshot shows the Weka software interface with the 'Preprocess' tab selected. The 'Filter' section at the top contains the 'AddCluster' filter with the following command: `-W "weka.clusterers.SimpleKMeans -V -N 3 -A \"weka.core.EuclideanDistance -R first-last\" -I 500 -S 10\" -I 5`. The 'Apply' button is highlighted with a red box. Below the filter, the 'Current relation' is 'FishersIrisDataset-weka.filters.unsupervised.attribute.Remove-R1-wek...' with 150 instances and 6 attributes. The 'Attributes' section on the left lists 'Sepal Length', 'Sepal Width', 'Petal Length', 'Petal Width', 'Species', and 'cluster'. The 'cluster' attribute is highlighted with a blue bar and a red box. A yellow callout bubble points to the 'cluster' attribute with the text: 'After the filter is being applied (**Apply**) we add the new attribute by the name of **cluster**'. On the right, the 'Selected attribute' section shows 'Name: cluster', 'Missing: 0 (0%)', 'Distinct: 3', and 'Type: Nominal Unique: 0 (0%)'. Below this, a table displays the distribution of the 'cluster' attribute:

No.	Label	Count
1	cluster1	61
2	cluster2	50
3	cluster3	39

At the bottom right, a bar chart visualizes this data with three bars: a blue bar for 'cluster1' (61), a red bar for 'cluster2' (50), and a cyan bar for 'cluster3' (39). The 'Class: cluster (Nom)' dropdown is set to 'cluster (Nom)', and the 'Visualize All' button is visible.

Using Clusters for Classification



Using Clusters for Classification

The screenshot shows the Orange3 software interface with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'NaiveBayes'. Under 'Test options', 'Use training set' is selected. The 'Result list' shows a single entry: '16:18:49 - bayes.NaiveBayes'. The main results pane displays various performance metrics and a confusion matrix.

Classifier: NaiveBayes

Test options:

- ☒ Use training set
- ☐ Supplied test set (Set...)
- ☐ Cross-validation (Folds: 10)
- ☐ Percentage split (%: 66)

Result list (right-click for options):

- 16:18:49 - bayes.NaiveBayes

Summary:

	mean	std. dev.	weight sum	precision
Pe	1.4108	0.2837	61	0.1143
	0.2766	0.1074	50	0.1143
	2.0806	0.2717	39	0.1143

Time taken:

=== Evaluation Summary ===

Correctly Classified Instances: 98.6667 %
Incorrectly Classified Instances: 1.3333 %

Kappa statistic: 0.9796
Mean absolute error: 0.0206
Root mean squared error: 0.0851
Relative absolute error: 4.7192 %
Root relative squared error: 18.209 %
Total Number of Instances: 150

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	1	0.022	0.968				cluster1
1		0	1				cluster2
0.949		0	1				cluster3
Weighted Avg.	0.987	0.009	0.987				

=== Confusion Matrix ===

a	b	c	<-- classified as
61	0	0	a = cluster1
0	50	0	b = cluster2
2	0	37	c = cluster3

Annotations:

- We use the NaiveBayes classifier
- We do the classification according to the cluster attribute
- The confusion matrix

Expectation Maximization (EM)

The EM algorithm consists of 2 key steps:

- **E (expectation) step** – calculation of the cluster probabilities; in this step we assume that we know the values of all the model parameters;
- **M (maximization) step** – calculation of the model parameters; we aim to “maximize” the likelihood of the model given the available data

These steps are repeated until the algorithm starts to converge

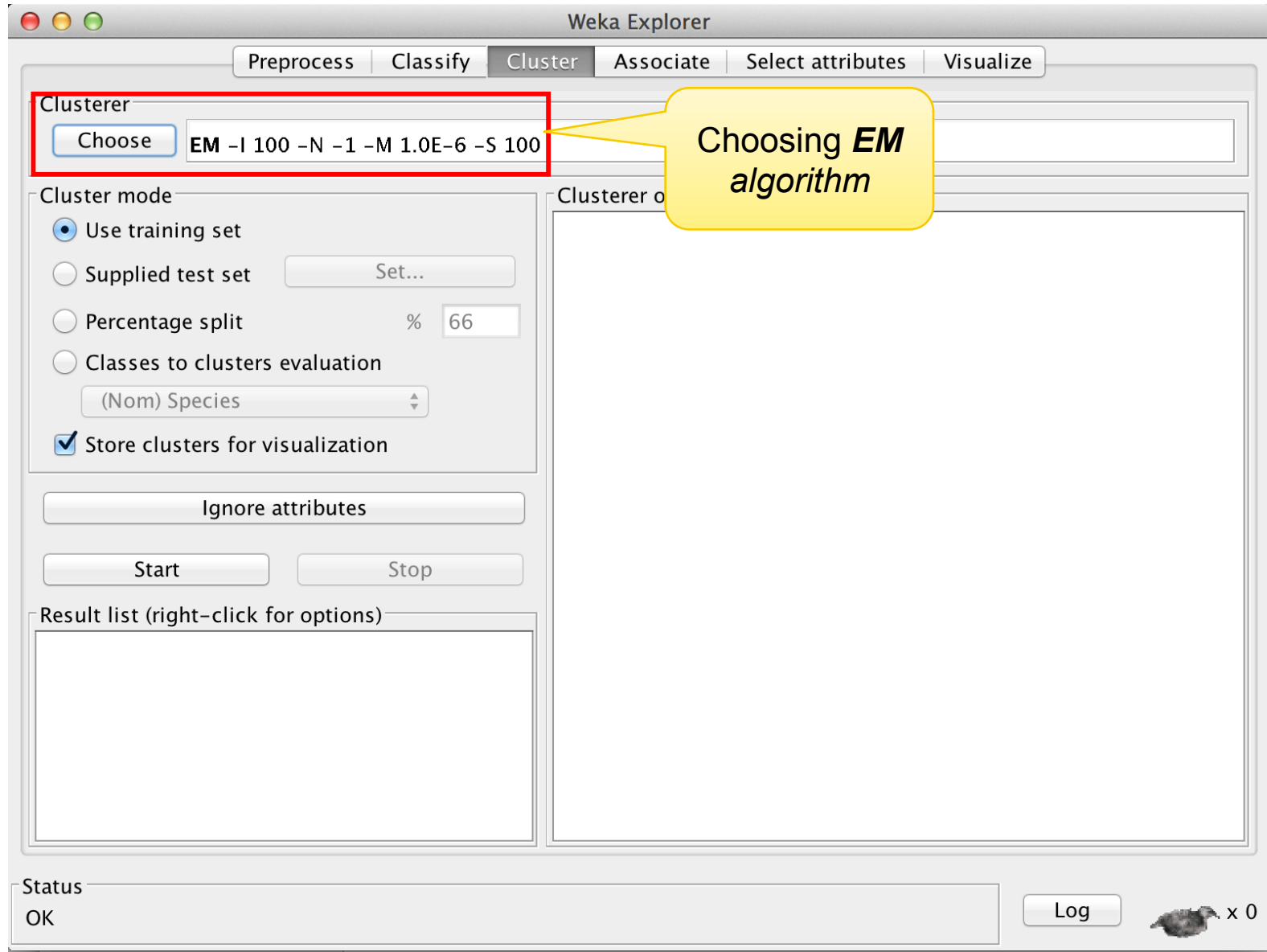
Expectation Maximization (EM)

To solve the described problem, we can apply a procedure similar to the one used for the K means algorithm:

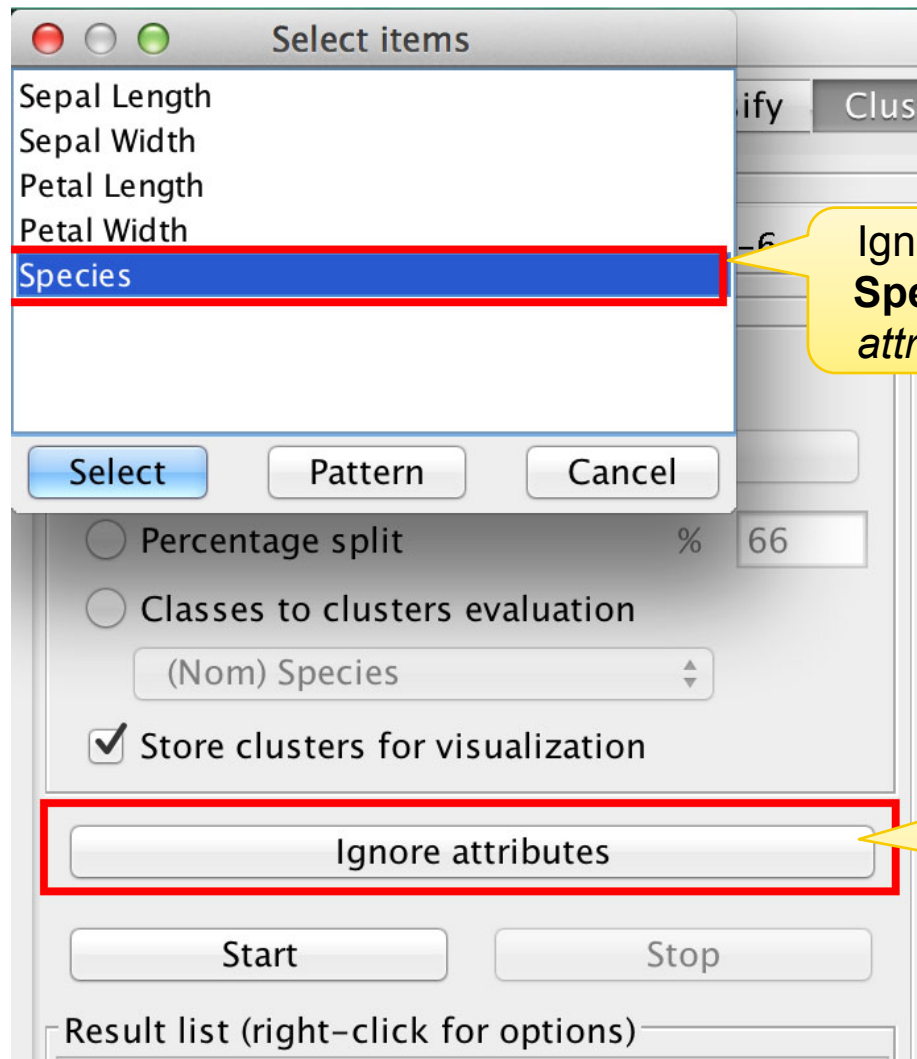
1. start by defining the number of clusters (k) and randomly choosing the model parameters ($\mu_i, \sigma_i, p_i, i = 1, k$)
2. for the given parameter values, compute, for each instance, the probability of belonging to each of the k clusters
3. use the computed probabilities to re-estimate the parameter values

Repeat steps 2) and 3) until the parameter values start to converge

Using EM in Weka



Ignoring Class Attribute



Ignoring
Species
attribute

Selecting which
attributes to ignore
during the clustering
process

Recommendations and credits

Weka Tutorials and Assignments @ The Technology Forge

- <http://www.technologyforge.net/WekaTutorials/>

"Data Mining with Weka" and "More Data Mining with Weka": MOOCs from the University of Waikato. A self-paced session of "Data Mining with Weka" runs until 23 October June 2015.

- <https://www.youtube.com/user/WekaMOOC/>

(Anonymous) survey for your
comments and suggestions:

<http://goo.gl/cqdp3l>

ANY QUESTIONS?

UROŠ KRČADINAC

EMAIL: uros@krcadinac.com

URL: <http://krcadinac.com>