Data Mining & Organization: Iris and others data sets

Clustering

Donatella Merlini

Università di Firenze
Corso di Laurea Magistrale in Informatica
Curriculum Data Science

WEKA K-means: parameters

- DisplayStdDev: displays the standard deviation of the individual point distances from the center of the cluster. The measurement is reported separately for each attribute.
  - The smaller the StdDev the greater the cluster cohesion with respect to the attribute.
  - Allows you to choose which attributes to use in computing similarity.
- Distance function: type of distance used in the calculation
- MaxIteration: maximum number of iterations to get the convergence
- NumCluster: K value
- Seed: random value for the choice of the initial centroids; changing it changes their initial positioning

K-means clustering the Iris data set

- Apply simple K-means with `DisplayStdDev=true` and `NumCluster=3`, after ignoring the attribute `Class`

```
kMeans
======
Number of iterations: 6
Within cluster sum of squared errors: 6.998114004826762

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4
Cluster 1: 6.2,2.9,4.3,1.3
Cluster 2: 6.9,3.1,5.1,2.3

Missing values globally replaced with mean/mode
```

```
Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2
               (150.0)     (61.0)     (50.0)     (39.0)
=========================================================
slength         5.8433     5.8885      5.006     6.8462
             +/-0.8281  +/-0.4487  +/-0.3525  +/-0.5025

swidth           3.054     2.7377      3.418     3.0821
             +/-0.4336  +/-0.2934   +/-0.381  +/-0.2799

plength         3.7587     4.3967      1.464     5.7026
             +/-1.7644  +/-0.5269  +/-0.1735  +/-0.5194

pwidth          1.1987      1.418      0.244     2.0795
             +/-0.7632  +/-0.2723  +/-0.1072  +/-0.2811

Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

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

- Now apply simple K-means by selecting `Classes to cluster evaluation` instead of `Use training set`

```
Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

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

Class attribute: class
Classes to Clusters:

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

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

Projection on petal attributes

- You can visualize clustering results for each pair of attributes (use `Jitter`)

Projection on sepal attributes

You can save the clustering results as an `.arff` file by selecting the `Save` button

Improving K-means results

- In K-means clustering, there are a number of ways one can often use to improve results
- One of the most common is to normalize the data so that the differences in scale of the numerical attributes do not dominate the distance measure: in `WEKA` this can be done during the Pre-processing phase by using the filter: `Unsupervised` $\rightarrow$ `Attribute` $\rightarrow$ `Normalize`
- Visualization can sometimes help us discern the attributes that best separate the data: to this purpose, we can examine the scatter plots of the Iris data set

Clustering on petal attributes

- Now apply simple K-means by selecting `Classes to cluster evaluation` and ignoring attributes `slength` and `swidth`

```
kMeans
======
Number of iterations: 6
Within cluster sum of squared errors: 1.7050986081225123

Initial starting points (random):

Cluster 0: 4.7,1.4
Cluster 1: 4.3,1.3
Cluster 2: 5.1,2.3

Missing values globally replaced with mean/mode
```

```
Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2
               (150.0)     (52.0)     (50.0)     (48.0)
=============================================================
plength         3.7587     4.2962      1.464     5.5667
             +/-1.7644  +/-0.5053  +/-0.1735   +/-0.549

pwidth          1.1987      1.325      0.244     2.0562
             +/-0.7632  +/-0.1856  +/-0.1072  +/-0.2422

Time taken to build model (full training data) : 0.01 seconds
=== Model and evaluation on training set ===
Clustered Instances

0       52 ( 35%)
1       50 ( 33%)
2       48 ( 32%)

Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 48  0  2 | Iris-versicolor
  4  0 46 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica
Incorrectly clustered instances : 6.0   4      %
```

New projection on petal attributes

Exercise: the *FoodNutrientClassified* data set

Contains the nutrition information of 25 foods: load the *FoodNutrientClassified.arff* file.

- Normalize and cluster data using K-means with a number of clusters between 2 and 6
- Analyze the results by making assumptions about the meaning of the classes according to the characteristics of the centroid and StdDev of the clusters.

Exercise: the *Coordinates* data set

Contains geographic coordinates of 480 points: load the *Coordinates.arff* file.

- Classify data using K-means with a number of clusters between 2 and 6
- How does SSE change?
- Starting from which K SSE value stabilizes?
- Can K-means capture natural clusters?

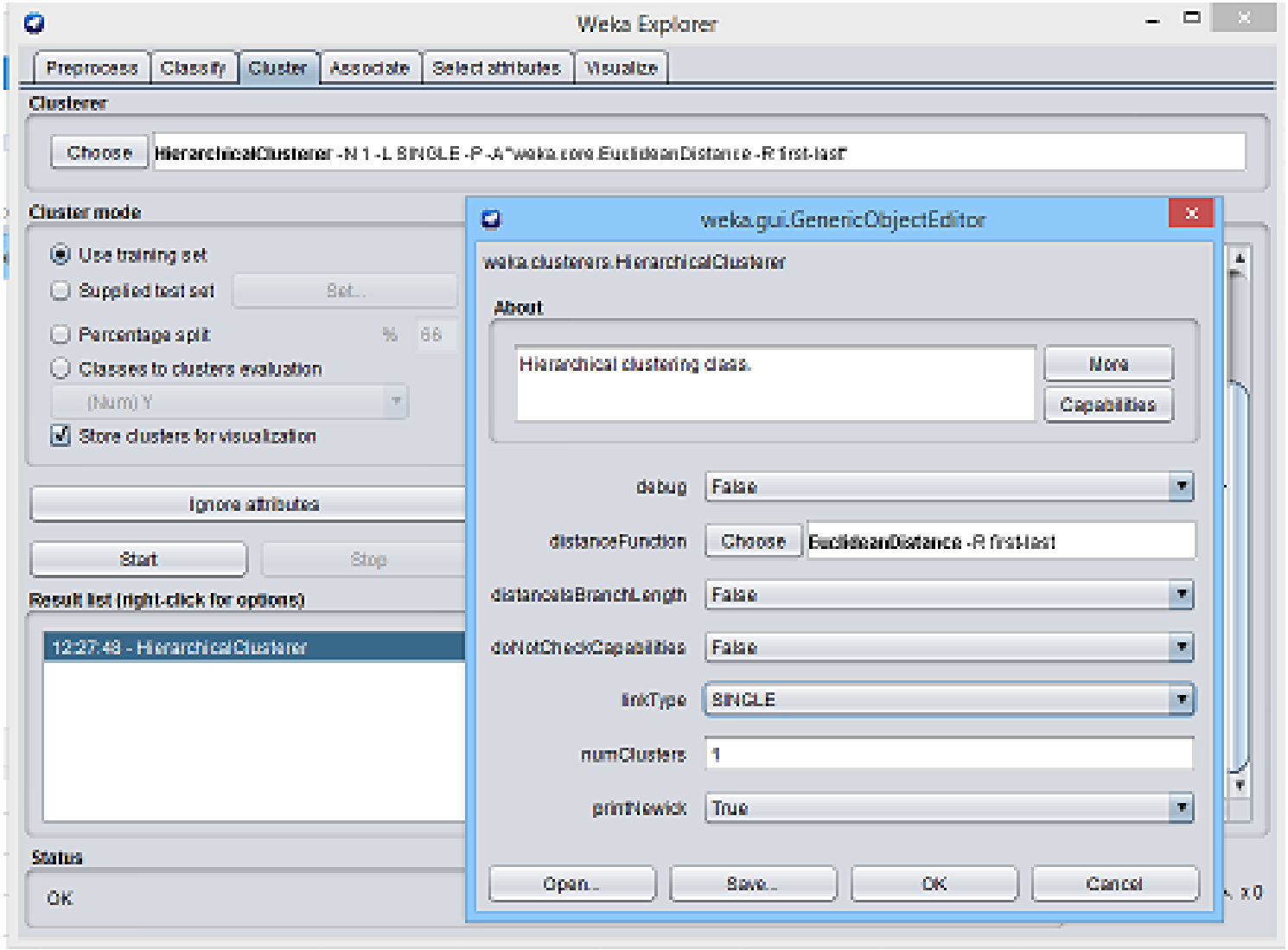
Numerical and nominal attributes in K-means

- For two numeric attribute values x and y, the value of $x - y$ is used in the distance calculation.
- For two nominal attribute values x and y, 0 is used when the two values are the same, and 1 is used when they are different.
- This only makes sense when numeric attributes have been rescaled (normalized) to the $[0, 1]$ interval. `EuclideanDistance` and `ManhattanDistance` both do this by default.

WEKA Agglomerative Hierarchical cluster

- The first column of the data set should be of type string to visualize the correct labels in the dendogram: you can use the filter `Unsupervised` → `Attribute` → `NominalToString`
- Different distance functions and link type can be used
- To visualize the complete dendogram set numCluster to 1
- The cluster can be printed in Newick format
- We work on the following small data set:

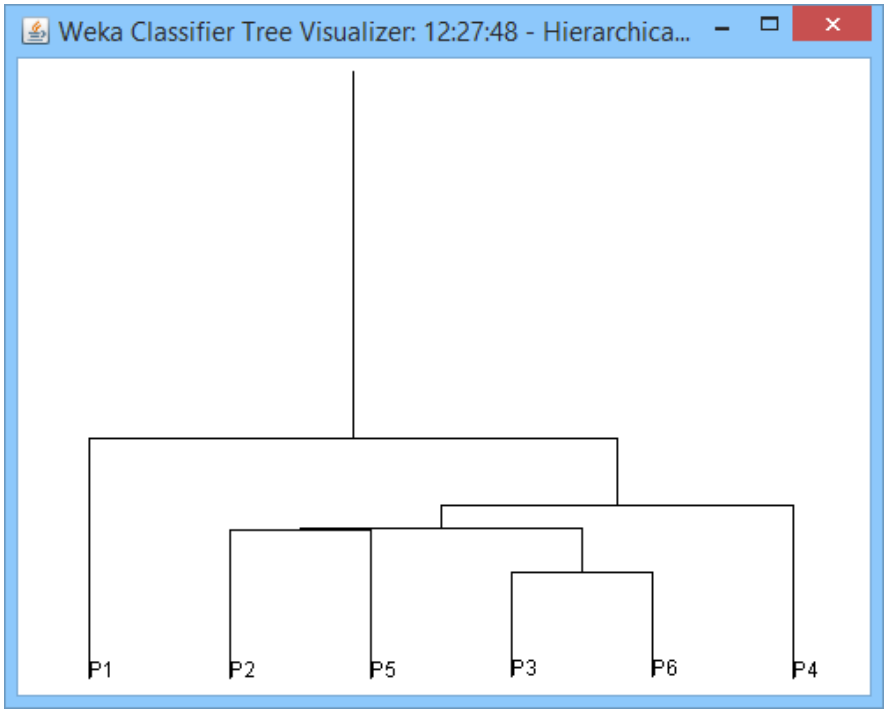
Point	X	Y
P1	0.4	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.3


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
HierarchicalClusterer -N 1 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) Y
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
12:27:49 - HierarchicalClusterer
Status
OK
weka.gui.GenericObjectEditor
weka.clusterers.HierarchicalClusterer
About
Hierarchical clustering class.
More
Capabilities
debug
False
distanceFunction
Choose
EuclideanDistance -R first-last
distanceIsBranchLength
False
doNotCheckCapabilities
False
linkType
SINGLE
numClusters
1
printNewick
True
Open...
Save...
OK
Cancel

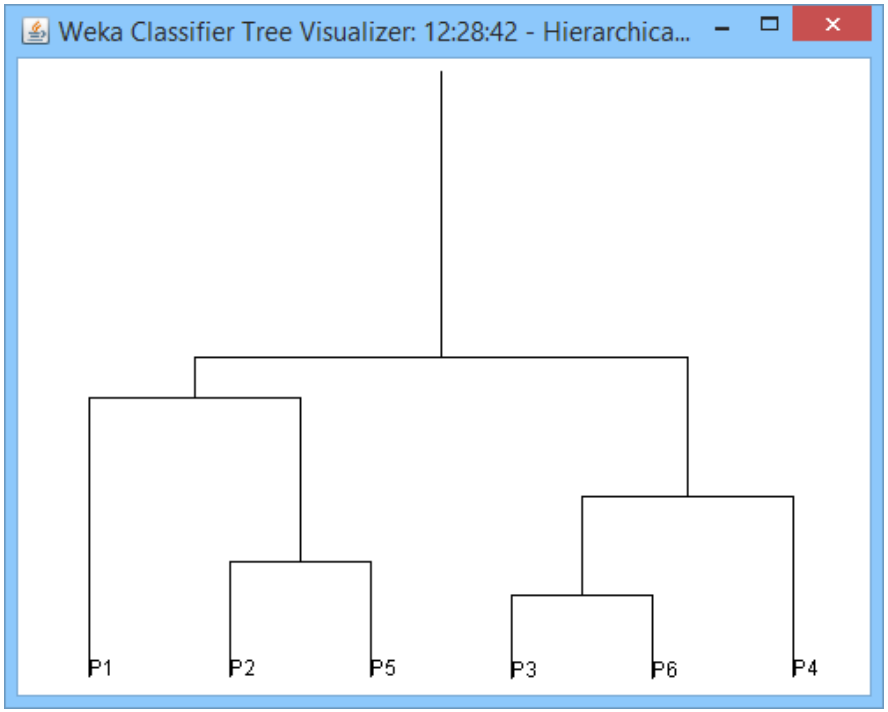
The results with SINGLE link

```
=== Run information ===
Scheme:       weka.clusterers.HierarchicalClusterer -N 1 -L SINGLE -P -A "weka.core.EuclideanDistance
              -R first-last"
Relation:     HierarchicalDataSetLibro-weka.filters.unsupervised.attribute.NominalToString-Cfirst
Instances:    6
Attributes:   3
              Point
              X
              Y
Test mode:    evaluate on training data

=== Clustering model (full training set) ===

Cluster 0
(P1:0.63226,(((P2:0.38853,P5:0.38853):0.00465,(P3:0.2766,P6:0.2766):0.11658):0.05999,P4:0.45317):0.17909)

Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0      6 (100%)
```

The resulting dendograms

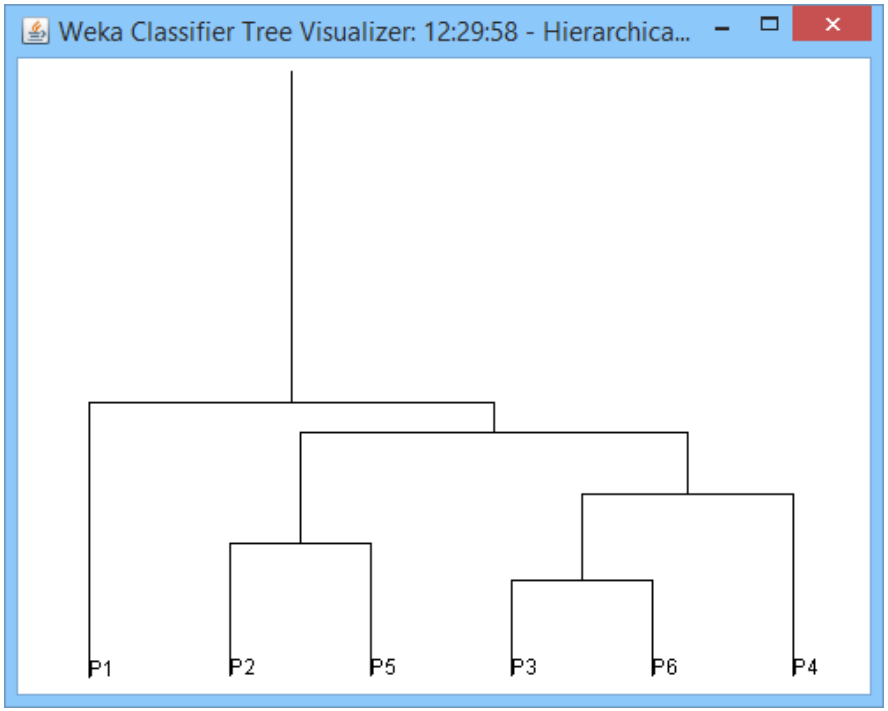


Single link

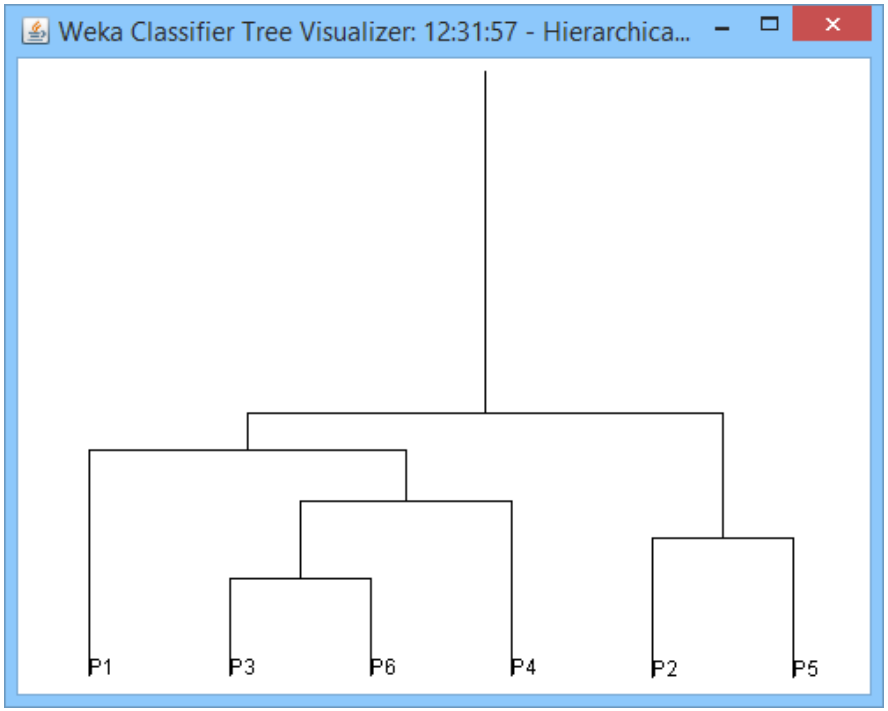


Complete link

The resulting dendograms



Average link



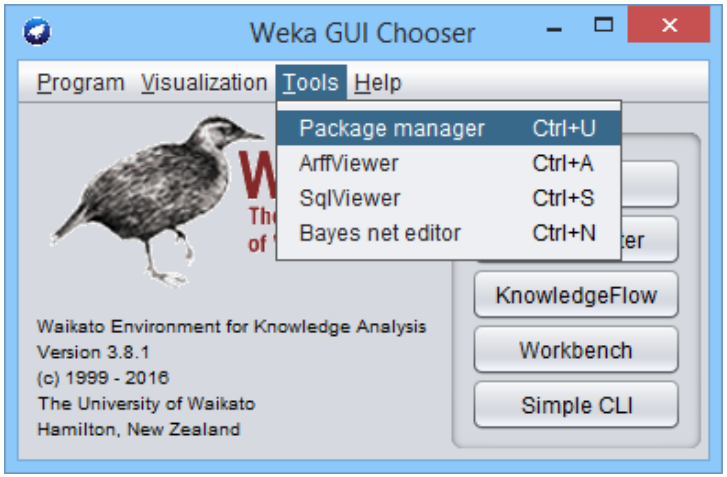
Ward link

Exercise

- Use `KEWA` agglomerative hierarchical clustering on the Iris data set
- Preprocess the data in order to apply the algorithm
- Try the algorithm by using several options:
  - select different clustering attributes
  - select different lynk type options
  - use different number of clusters
- Discuss the results

WEKA DBSCAN algorithm

- In the most recent version of `WEKA` the DBSCAN algorithm is not available in the basic version of the software
- The algorithm must be added from the *Package manager* menu (please search for `Optics_dbScan` algorithm and install it)



WEKA DBSCAN parameters

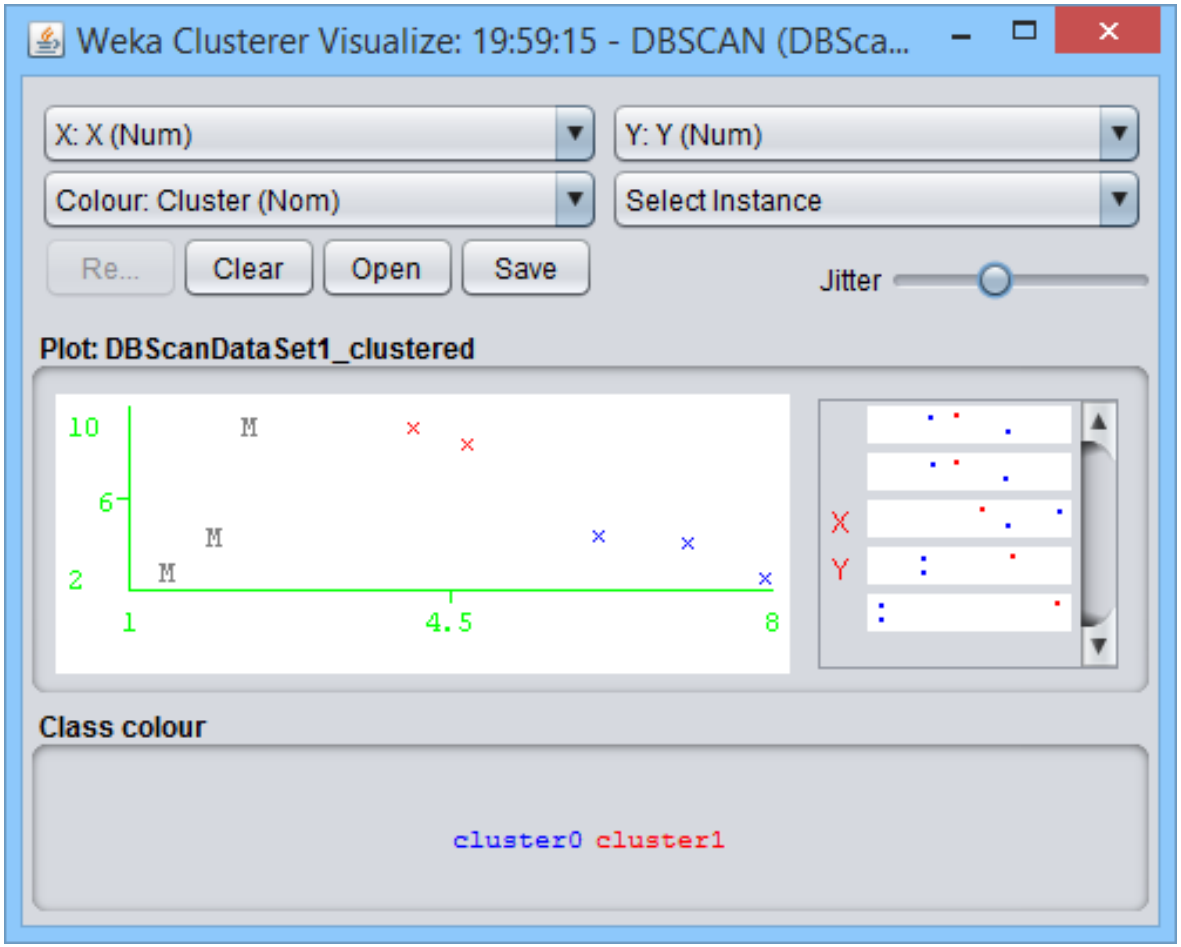
- Different distance functions, epsilon and minpoints can be used
- By default, `EuclideanDistance` normalizes attributes values to lie between zero and one, set `dontNormalize` to `True` for using more intuitive values for `epsilon` and `minpoints`.
- We work on the following small data set:

Point	X	Y
P1	2	10
P2	2	5
P3	8	4
P4	5	8
P5	7	5
P6	6	4
P7	1	2
P8	4	9

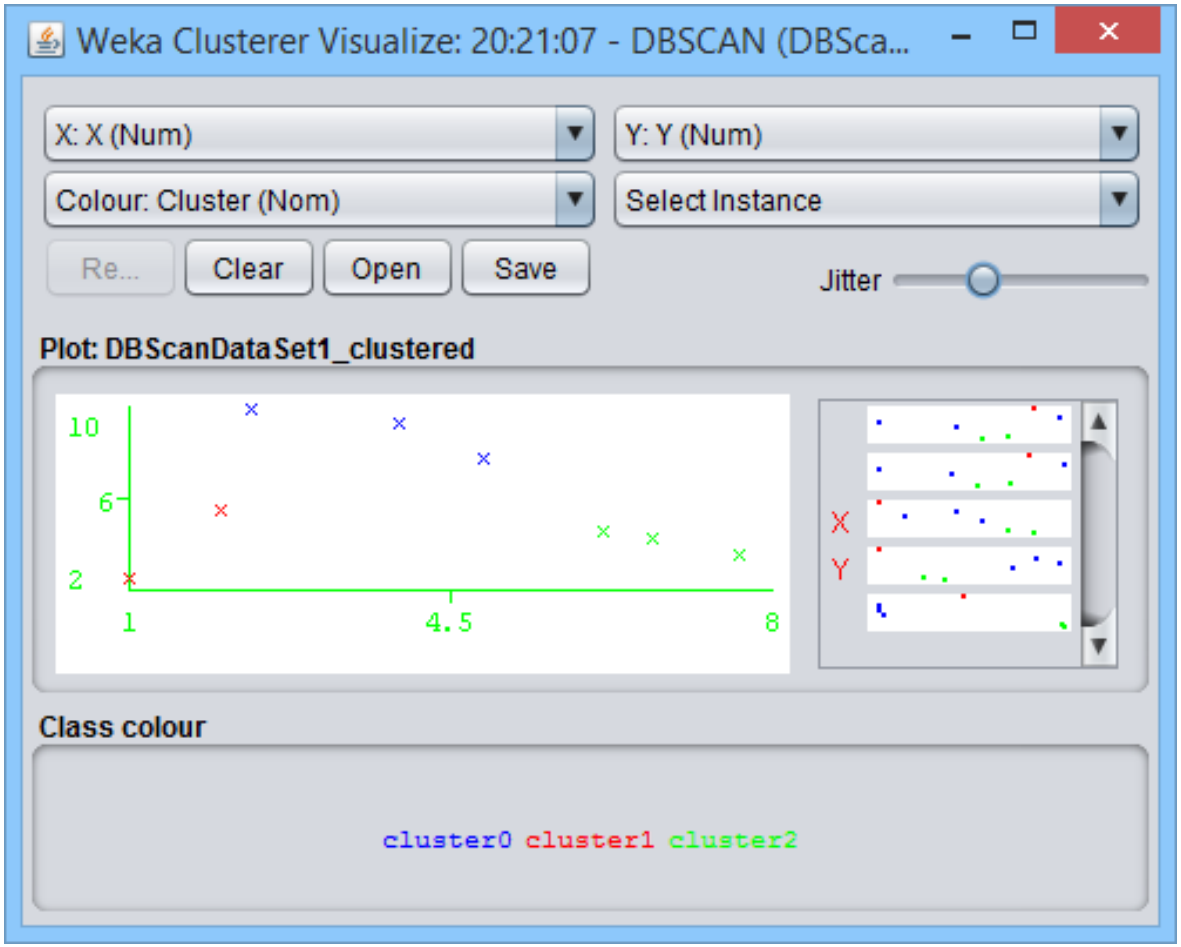
The results with $\epsilon = 2$ and minpoints=2

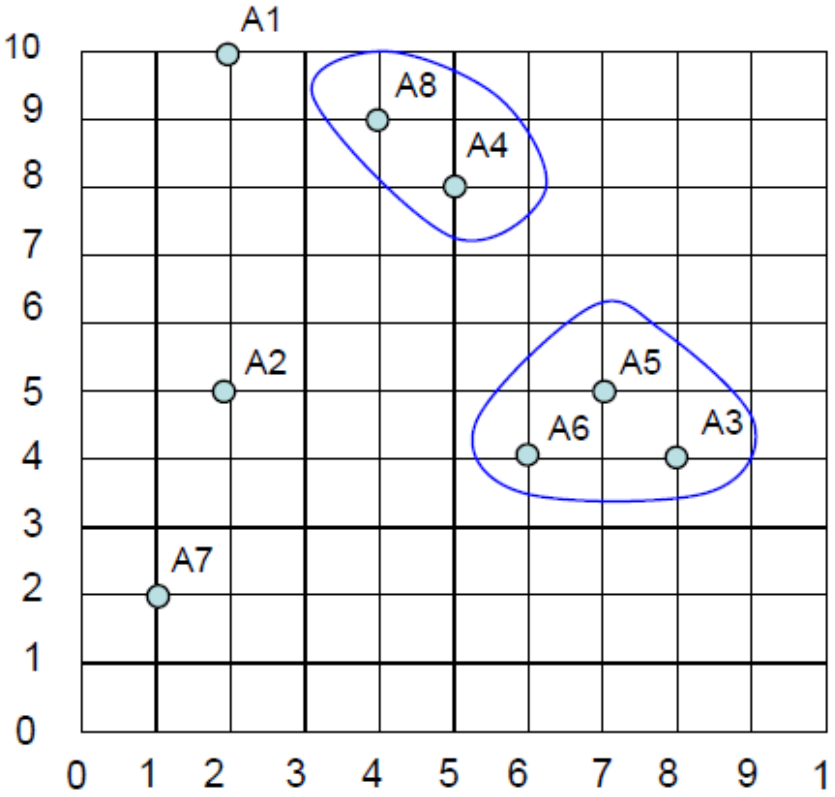
```
=== Clustering model (full training set) ===

DBSCAN clustering results
========================================================================================
Clustered DataObjects: 8
Number of attributes: 2
Epsilon: 2.0; minPoints: 2
Distance-type:
Number of generated clusters: 2
Elapsed time: .0
(0.) 2,10          -->  NOISE
(1.) 2,5           -->  NOISE
(2.) 8,4           -->  0
(3.) 5,8           -->  1
(4.) 7,5           -->  0
(5.) 6,4           -->  0
(6.) 1,2           -->  NOISE
(7.) 4,9           -->  1
Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===
Clustered Instances
0      3 ( 60%)
1      2 ( 40%)

Unclustered instances : 3
```

$\epsilon = 2$ and minpoints=2: 2 clusters and 3 outliers

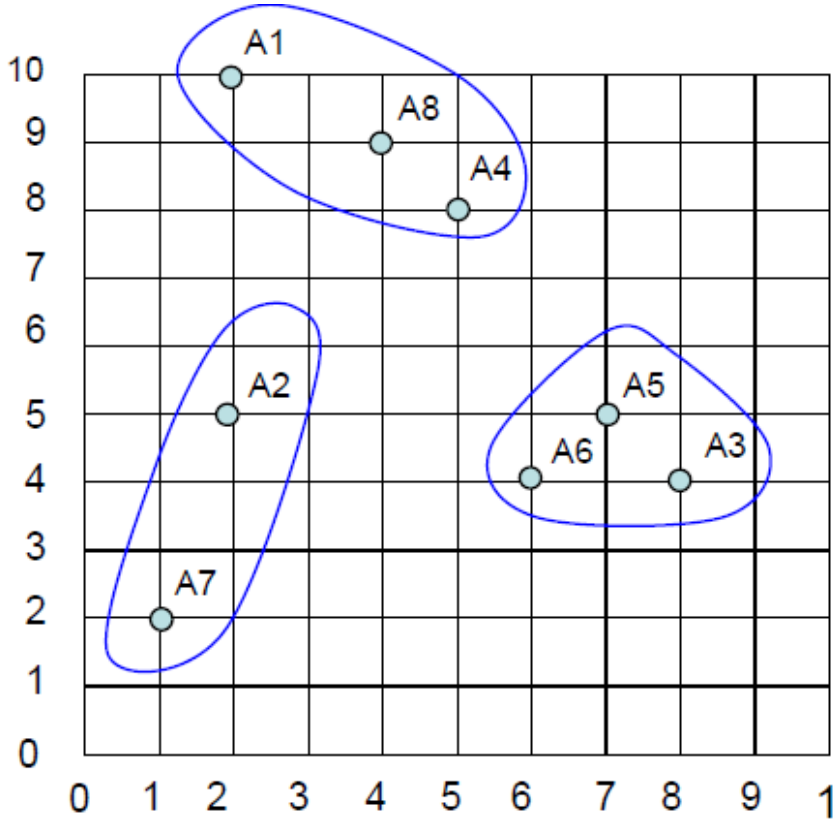


$\epsilon = 3.2$ and minpoints=2: 3 clusters and 0 outliers





Epsilon = 2



Epsilon = $\sqrt{10}$

Exercise

- Use KEWA DBSCAN algorithm on the Iris data set
- Preprocess the data in order to apply the algorithm
- Try the algorithm by using several options:
  - select different clustering attributes
  - select different epsilon and minpoints
- Discuss the results

Coordinates with DBSCAN

- Evaluate the result of the classification with DBSCAN
- Identify the correct values for epsilon and minpoints