Lab 1
732A61 Data Mining - Clustering and Association Analysis

Carles Sans Fuentes

February 8, 2017

Assignment 1

The main goals of this assignment are:

- Gaining familiarity with the data mining toolkit Weka
- Learning to apply clustering algorithms using Weka
- Understanding outputs produced by clustering tools in Weka

For this, two clustering algorithms must be evaluated in Weka on the food.arff data (a data set providing nutrient levels of 27 kinds of food. The mounts of energy, protein, fat, calcium and iron have been measured in a 3 ounce portion of the various foods):

- the SimpleKmeans implemented in euclidian distance
- MakeDensityBasedClusters, an implementation of a density-based method

Exercise 1 (SimpleKmeans)

Apply "SimpleKMeans" to your data. In Weka euclidian distance is implemented in SimpleKmeans. You can set the number of clusters and seed of a random algorithm for generating initial cluster centers. Experiment with the algorithm as follows:

The set of attributes chosen for clustering in my study are going to be fat and energy. Inductively, I have always believed that to some extent, the fatter some food is, the more energy it provides. For this reason, trying to separate data on these two variables may give us some other relations the clusters that were not expected.

When doing clustering, the attribute name is not used since this attribute is not observable but just used as an identity key.

In order to do clusters I have chosen the variables previously mentioned for classifying each variable in 2 or 5 clusters: fat and energy. The results for all of them are provided below:

————————-seed 10 (2 clusters)————————-

```
1  === Run information *seed 10 cluster 2*===
2
3  Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000
       -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I
       500 -num-slots 1 -S 10
4  Relation:     food
5  Instances:    27
6  Attributes:   6
7                Energy
8                Fat
9  Ignored:
10               Name
11               Protein
12               Calcium
```

```
              Iron
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 0.8481897660818714

Initial staring points (random):

Cluster 0: 340,28
Cluster 1: 170,7

Missing values globally replaced with mean/mode

Final cluster centroids:
                          Cluster#
Attribute    Full Data          0          1
                  (27)        (8)       (19)
==========================================
Energy         207.4074    341.875   150.7895
Fat             13.4815     28.875          7



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

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

Clustered Instances

0       8 ( 30%)
1      19 ( 70%)
```

-seed 99 (2 clusters)-

```
=== Run information *seed 99 cluster 2*===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000
    -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I
    500 -num-slots 1 -S 99
Relation:     food
Instances:    27
Attributes:   6
              Energy
              Fat
Ignored:
              Name
              Protein
              Calcium
              Iron
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 0.854594931773879

Initial staring points (random):

Cluster 0: 195,11
Cluster 1: 185,9

Missing values globally replaced with mean/mode

Final cluster centroids:
                          Cluster#
Attribute    Full Data          0          1
                  (27)        (9)       (18)
==========================================
Energy         207.4074   331.1111   145.5556
Fat             13.4815    27.5556     6.4444

```

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

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

Clustered Instances

0       9 ( 33%)
1      18 ( 67%)
```

-seed 10 (5 clusters)-

```
=== Run information *seed 10 (5 clusters)* ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000
    -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I
    500 -num-slots 1 -S 10
Relation:     food
Instances:    27
Attributes:   6
              Energy
              Fat
Ignored:
              Name
              Protein
              Calcium
              Iron
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 0.20447197056969912

Initial staring points (random):

Cluster 0: 340,28
Cluster 1: 170,7
Cluster 2: 90,2
Cluster 3: 180,9
Cluster 4: 300,25

Missing values globally replaced with mean/mode

Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4
                  (27)        (6)        (7)        (5)        (6)        (3)
============================================================================
Energy         207.4074   361.6667   149.2857         86   190.8333        270
Fat             13.4815         31          6        1.6         11    20.6667



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

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

Clustered Instances

0       6 ( 22%)
1       7 ( 26%)
2       5 ( 19%)
3       6 ( 22%)
4       3 ( 11%)
```

-seed 99 (5 clusters)-

```
=== Run information *seed 99 (5 clusters)*===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000
    -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I
    500 -num-slots 1 -S 99
```

```
Relation:     food
Instances:    27
Attributes:   6
              Energy
              Fat
Ignored:
              Name
              Protein
              Calcium
              Iron
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 0.21739706869512965

Initial staring points (random):

Cluster 0: 195,11
Cluster 1: 185,9
Cluster 2: 180,10
Cluster 3: 375,32
Cluster 4: 265,20

Missing values globally replaced with mean/mode

Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4
                  (27)        (3)        (8)        (7)        (6)        (3)
============================================================================
Energy         207.4074        200      102.5   171.4286   361.6667        270
Fat             13.4815    12.6667       2.75          8         31    20.6667



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

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

Clustered Instances

0       3 ( 11%)
1       8 ( 30%)
2       7 ( 26%)
3       6 ( 22%)
4       3 ( 11%)
```

Comparison

When doing classification, results must be evaluated taking into consideration different seeds in order to see whether output differs from one seed to another. The seed accounts for the way random numbers are generated. If the seed is fixed,then even a randomized algorithm will be deterministic, starting always from same initial cluster center. Since KMeans is not deterministic, but we want repeatable results - you fix a seed in order your experiment to be repeatable and proved by other scientists, and then you perform different outputs to assess whether your output depends on the way random numbers are generated.

When comparing the number of clusters equal to 2, it can be seen that no matter the seed (in our case 10 and 11) the output reached is the same. This can mean that it exists an optimal classification for our data when classifying it across these two classes. The same does not happen when classifying the data on 5 different clusters taking into account the same attributes. Results differ quite a lot depending on the seed. The exact results can be seen in the lists above, whereas a visualization of the clusters can be seen below in the figures 1 to 4. ———————— 2 Clusters———————

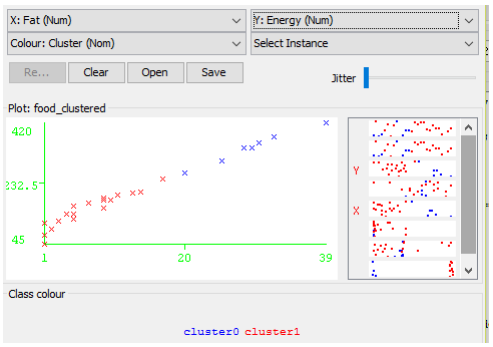

Figure 1: 2 Cluster classification from energy and fat attributes (energy fat), seed 10

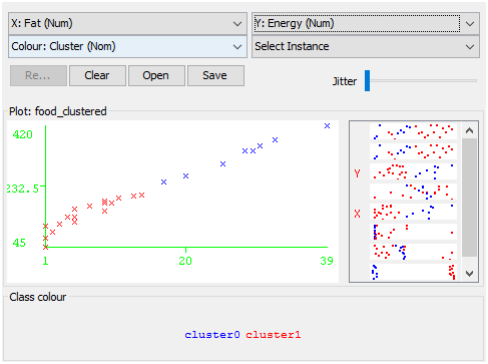

Figure 2: 2 Cluster classification from energy and fat attributes (energy fat), seed 99

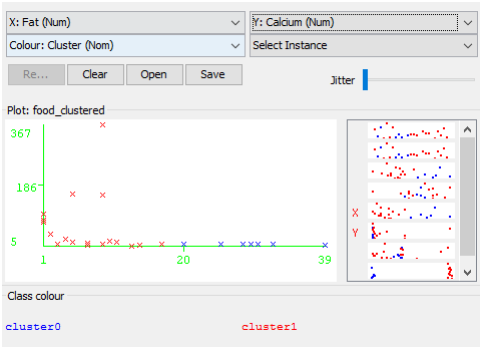

Figure 3: 2 Cluster classification from energy and fat attributes (calcium fat), seed 10

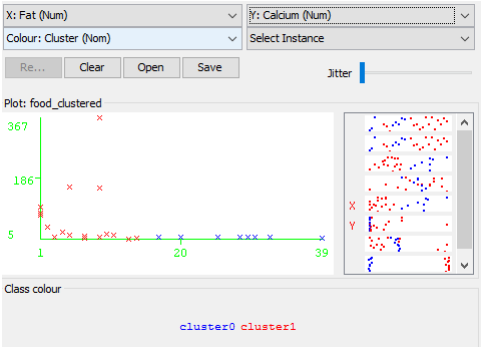

Figure 4: 2 Cluster classification from energy and fat attributes (calcium fat), seed 99

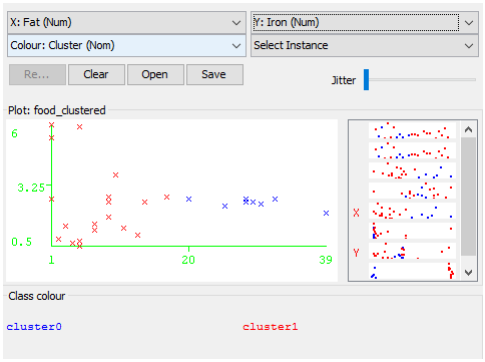

Figure 5: 2 Cluster classification from energy and fat attributes (iron calcium), seed 10

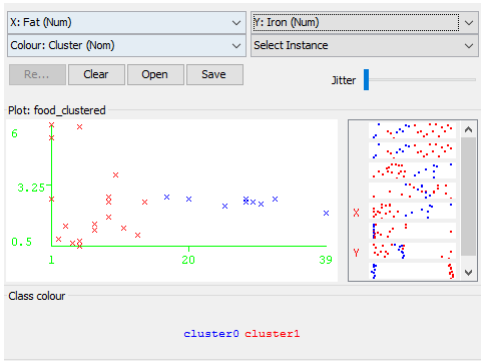

Figure 6: 2 Cluster classification from energy and fat attributes (iron calcium), seed 99

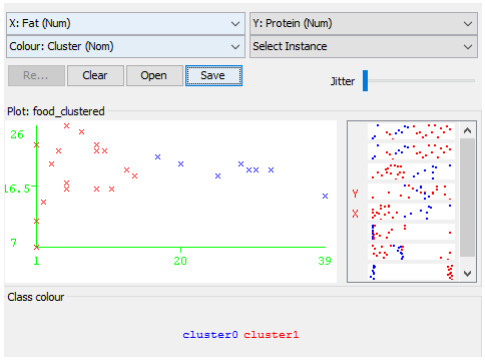

Figure 7: 2 Cluster classification from energy and fat attributes (Protein calcium), seed 10

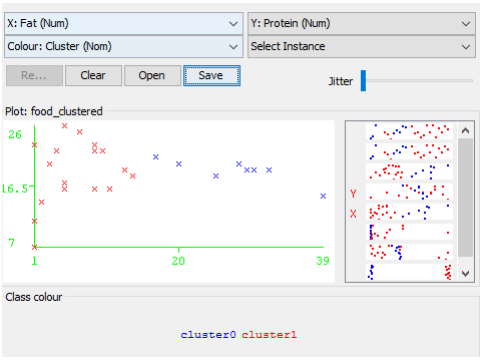

Figure 8: 2 Cluster classification from energy and fat attributes (Protein calcium), seed 99

5 Clusters

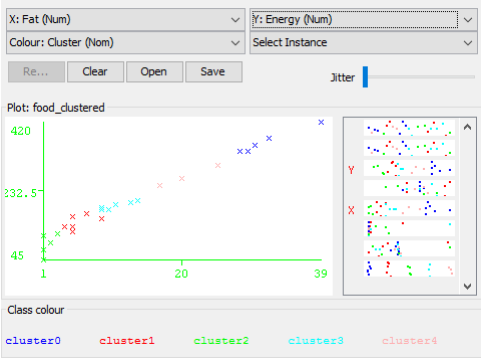

Figure 9: 2 Cluster classification from energy and fat attributes (energy fat), seed 10

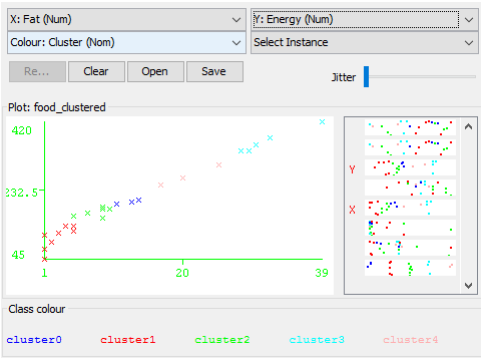

Figure 10: 2 Cluster classification from energy and fat attributes (energy fat), seed 99

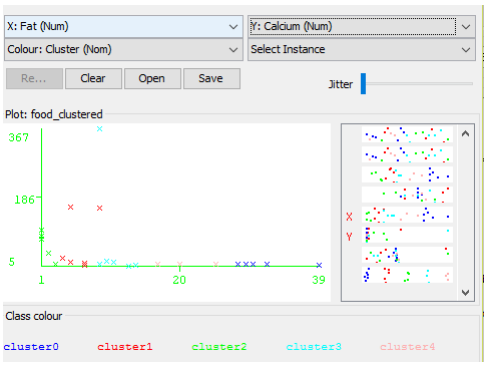

Figure 11: 2 Cluster classification from energy and fat attributes (energy calcium), seed 10

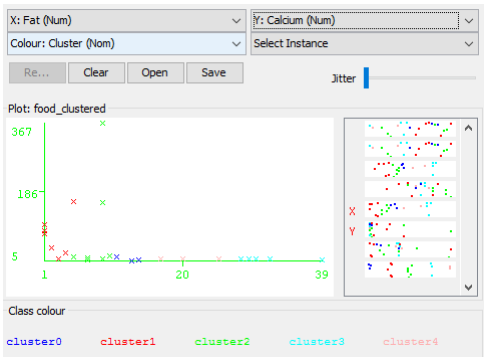

Figure 12: 2 Cluster classification from energy and fat attributes (energy calcium), seed 99

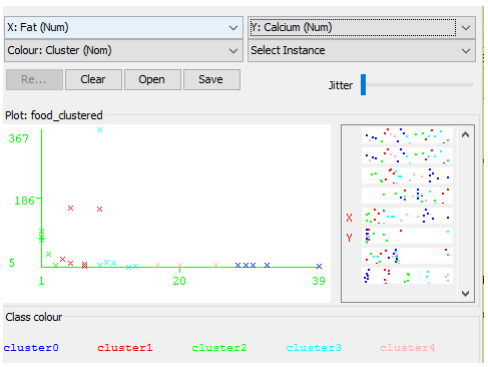

Figure 13: 2 Cluster classification from energy and fat attributes (iron calcium), seed 10

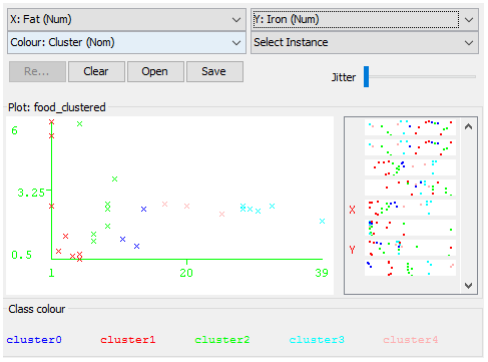

Figure 14: 2 Cluster classification from energy and fat attributes (iron calcium), seed 99

Figure 15: 2 Cluster classification from energy and fat attributes (Protein calcium), seed 10

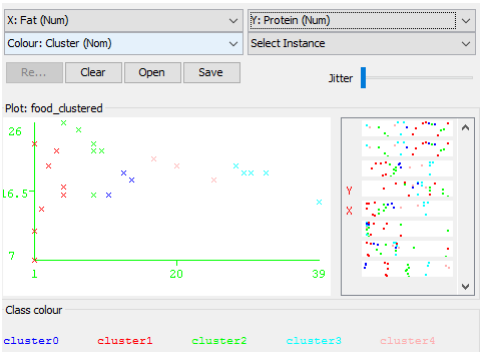

Figure 16: 2 Cluster classification from energy and fat attributes (Protein calcium), seed 99

When trying to assess my answer on whether the clusters are good, I will base my answer on the fact that (1) if graph clustering and size of the clusters do not differ a lot across seeds, then the classification is good, and (2) how classification is found together for given levels of an attribute (in this case regressing all the variables on fat) such that if all the cluster found in a 2D graph is easily separated by a line, then the classification would be good.

I think that the output when separating only across two clusters is quite good even though the results for both seeds differ a little bit numerically and in the classification of one variable but that is not much (see figure 1 and 2. In order to assess whether this classification is good or not, the following graphics (figure 3-8) show that the regression of each variable with fat in the X axis, and it can be seen that it can be easily drawn a line that separates the data.

I think that the classification when separating across five clusters is not acceptable given that the two clusters accounting for high values of fat and energy do not change but for the other 3 they do change a quite a bit (clusters from 0-2). In this case, it is less sure that the clusters are good and further research should be done across seeds (see 9 and 10 for the graph classification). Nevertheless, it cannot be said that it is a really bad classification since there seem to exist some line on each case when showing the different variables with all the attributes in the Y-axis while having on the X-axis the fat attribute (see figure 11-16).

When clustering, we try to find similarities within the same cluster and dissimilarities across clusters. From the clusters got, good enough similarities have been founded to say that the cluster is good, at least for the case of two clusters. On the case of having 5 clusters there are also similarities between clusters, preferably for those with high fat and high energy, but dissimilarities across the lower part are not that clear.

To see what each cluster represents I will take as an example figure 1. Cluster 0 represents as label "high fat high energy", with fat higher than 20 and energy higher than 132 calories approximately, while Cluster 1 will represent the opposite: "low fat low energy", with fat lower than 20 and energy

lower than 132 calories approximately. This will work for all the figure with seed 10 and number of clusters equal to 2.

Exercise 2 (MakeDensityBasedClusters)

Now with MakeDensityBasedClusters, a SimpleKMeans is turned into a densitiy-based cluster. The cluster chosen in part five for the labels, so the one in Figure 1 where there are 2 clusters: cluster0 being "high fat high energy", and cluster 1 being being "low fat low energy".

When clustering with this new method, two different standard deviations are used: 0.1 on the first one and 300 on the second one. Results for both clusters are provided below. The main conclusion driven by the results is that by assigning higher standard deviations the algorithm tries to force everything into the first cluster (in this case cluster 0) if the variable is inside this standard deviation distance from the central point. When the standard deviation is 0.1, same results are reached than with the Kmeans method, whereas when standard deviation is settled by myself to be 300, everything is given the same class. Nevertheless, the loglikelihood reached in the latter case is lower, such that thanks to it we can see that when standard deviation is 0.1 better likelihoods and better results are got.

```
---*minstdev = 0.1----
Normal Distribution. Mean = 28.875 StdDev = 5.1097

Cluster: 1 Prior probability: 0.6897

Attribute: Energy
Normal Distribution. Mean = 150.7895 StdDev = 49.0505
Attribute: Fat
Normal Distribution. Mean = 7 StdDev = 4.5422


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

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

Clustered Instances

0       8 ( 30%)
1      19 ( 70%)


Log likelihood: -8.84267
```

```
---*minstdev = 300----
Attribute: Fat
Normal Distribution. Mean = 28.875 StdDev = 300

Cluster: 1 Prior probability: 0.6897

Attribute: Energy
Normal Distribution. Mean = 150.7895 StdDev = 300
Attribute: Fat
Normal Distribution. Mean = 7 StdDev = 300


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

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

Clustered Instances

1      27 (100%)


Log likelihood: -13.33942
```