732A75: Clustering Lab

Zijie Feng, Jiawei Wu

2019-2-7

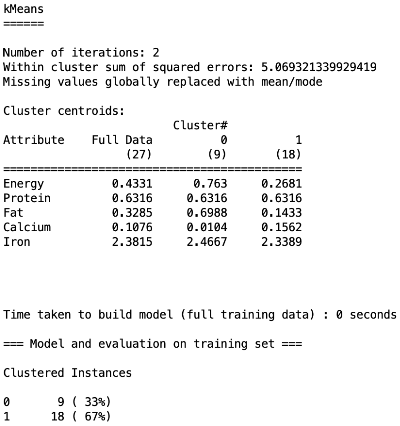
SimpleKmeans

1. Choose a set of attributes for clustering and give a motivation. (Hint: always ignore attribute "name". Why does the name attribute need to be ignored?)

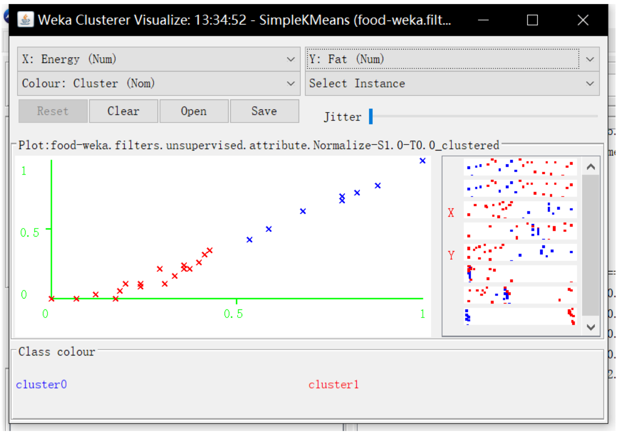
Since the attribute *name* is a nominal variable, it is only the lable of different things. So all the elements in such attribute are independent strings. In that case, it could not be useful for our clustering.

2. Experiment with at least two different numbers of clusters, e.g. 2 and 5, but with the same seed value 10.

number of clusrers:2



```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27)          (9)       (18)
=============================================
Energy          0.4331      0.763     0.2681
Protein         0.6316     0.6316     0.6316
Fat             0.3285     0.6988     0.1433
Calcium         0.1076     0.0104     0.1562
Iron            2.3815     2.4667     2.3389




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

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

Clustered Instances

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



Weka Clusterer Visualize: 13:34:52 - SimpleKMeans (food-weka.filt...

X: Energy (Num)

Y: Fat (Num)

Colour: Cluster (Nom)

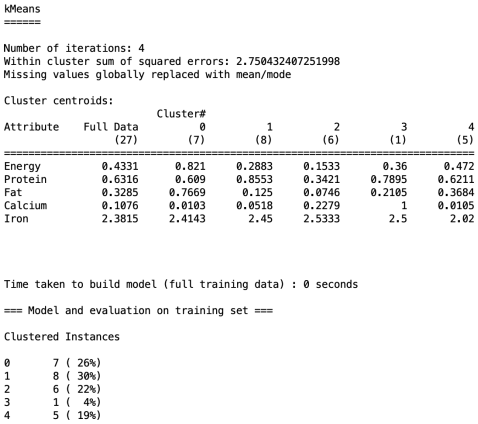
Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

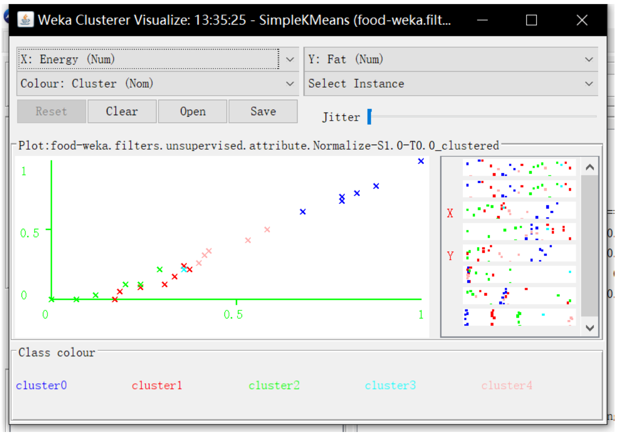
Class colour

cluster0 cluster1

number of clusrers:5



```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 2.750432407251998
Missing values globally replaced with mean/mode

Cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4
                (27)          (7)        (8)        (6)        (1)        (5)
================================================================================
Energy          0.4331      0.821     0.2883     0.1533       0.36      0.472
Protein         0.6316      0.609     0.8553     0.3421     0.7895     0.6211
Fat             0.3285     0.7669      0.125     0.0746     0.2105     0.3684
Calcium         0.1076     0.0103     0.0518     0.2279          1     0.0105
Iron            2.3815     2.4143       2.45     2.5333        2.5       2.02




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

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

Clustered Instances

0       7 ( 26%)
1       8 ( 30%)
2       6 ( 22%)
3       1 (  4%)
4       5 ( 19%)
```



Weka Clusterer Visualize: 13:35:25 - SimpleKMeans (food-weka.filt...

X: Energy (Num)

Y: Fat (Num)

Colour: Cluster (Nom)

Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

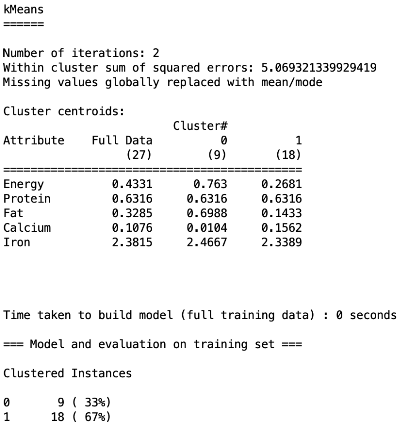
Class colour

cluster0 cluster1 cluster2 cluster3 cluster4

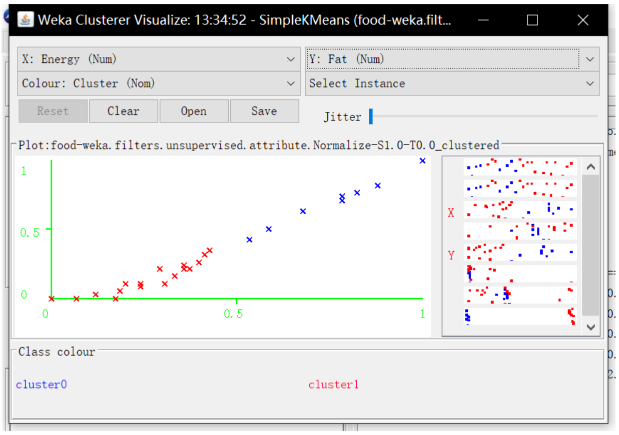
We scale all the data firstly. The cluster sum of squared errors is 5.07 when *k=2*, but 2.75 when *k=5*. It means that 5-mean algorithm might provide us a better clustering result. However, the plots with *k=5* seem more confused and unreadable sence there is over lap with cluster2, cluster1 and cluster4.

3. Then try with a different seed value, i.e. different initial cluster centers. Compare the results with the previous results. Explain what the seed value controls.

seed: 10 vs. 123456



```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                  (27)        (9)       (18)
============================================
Energy          0.4331      0.763     0.2681
Protein         0.6316     0.6316     0.6316
Fat             0.3285     0.6988     0.1433
Calcium         0.1076     0.0104     0.1562
Iron            2.3815     2.4667     2.3389




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

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

Clustered Instances

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



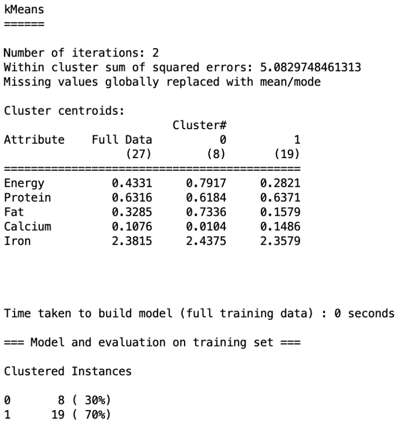
Weka Clusterer Visualize: 13:34:52 - SimpleKMeans (food-weka.filt...

X: Energy (Num)

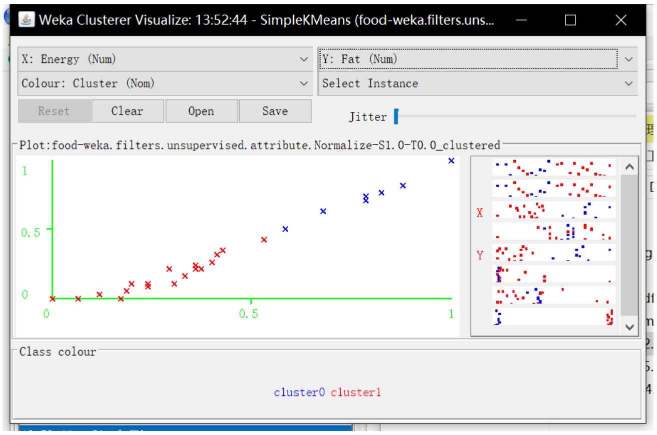
Y: Fat (Num)

Colour: Cluster (Nom)

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered



```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.0829748461313
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                  (27)        (8)       (19)
============================================
Energy          0.4331     0.7917     0.2821
Protein         0.6316     0.6184     0.6371
Fat             0.3285     0.7336     0.1579
Calcium         0.1076     0.0104     0.1486
Iron            2.3815     2.4375     2.3579




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

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

Clustered Instances

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



Weka Clusterer Visualize: 13:52:44 - SimpleKMeans (food-weka.filters.uns...

X: Energy (Num)

Y: Fat (Num)

Colour: Cluster (Nom)

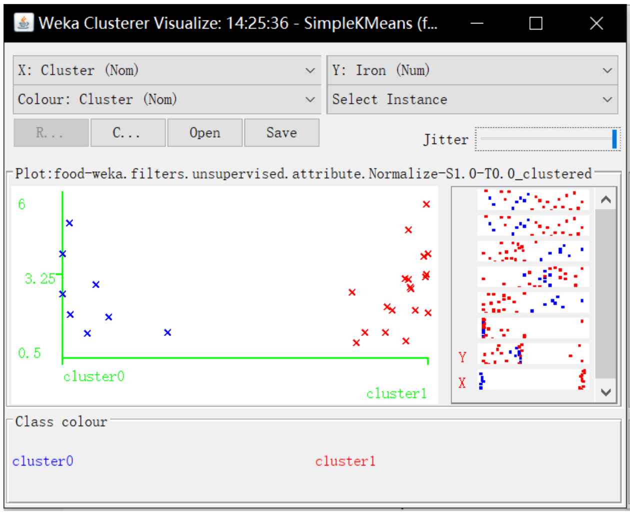
Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

As the plots and results shows, the seed doesn't change the within cluster sum of squared errors a lot, and the clustered instances are the same. So it won't change the result much.

K-mean algorithm is a method which could only find the local optimals, which means that it relies on the initial random clusters greatly. Different seeds represent different initial random clusters k-mean applies, the final clusters might thereby be different from time to time. This is the reason why we should run k-mean several times for best clustering.

4. Do you think the clusters are "good" clusters? (Are all of its members "similar" to each other? Are members from different clusters dissimilar?)

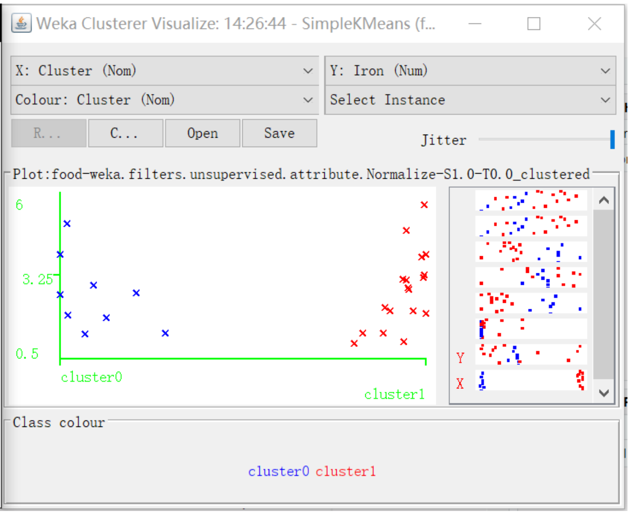
seed: 123456 vs. 10



Weka Clusterer Visualize: 14:25:36 - SimpleKMeans (f...

X: Cluster (Nom)
Y: Iron (Num)
Colour: Cluster (Nom)
Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered



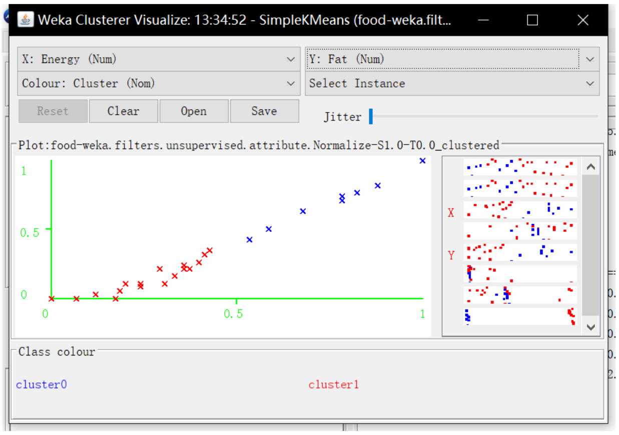
Weka Clusterer Visualize: 14:26:44 - SimpleKMeans (f...

X: Cluster (Nom)
Y: Iron (Num)
Colour: Cluster (Nom)
Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

Yes, the cluster is good enough. According to the plots, there is only one instance with different labels between two clustering (seed=123456 and seed=10), which means such kind of result might be the best clustering.

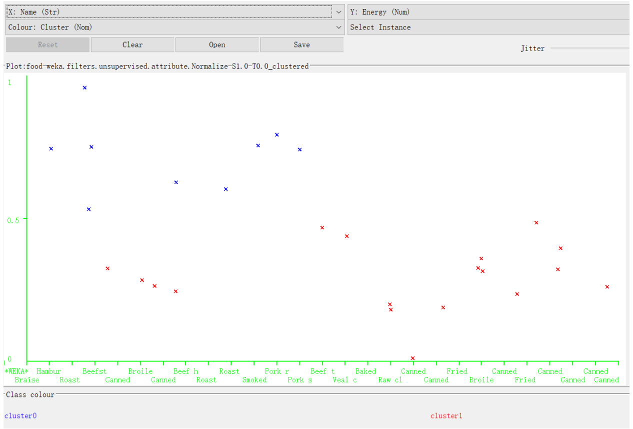
5. What does each cluster represent? Choose one of the results. Make up labels (words or phrases in English) which characterize each cluster.



Weka Clusterer Visualize: 13:34:52 - SimpleKMeans (food-weka.filt...

X: Energy (Num)
Y: Fat (Num)
Colour: Cluster (Nom)
Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered



X: Name (Str)
Y: Energy (Num)
Colour: Cluster (Nom)
Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

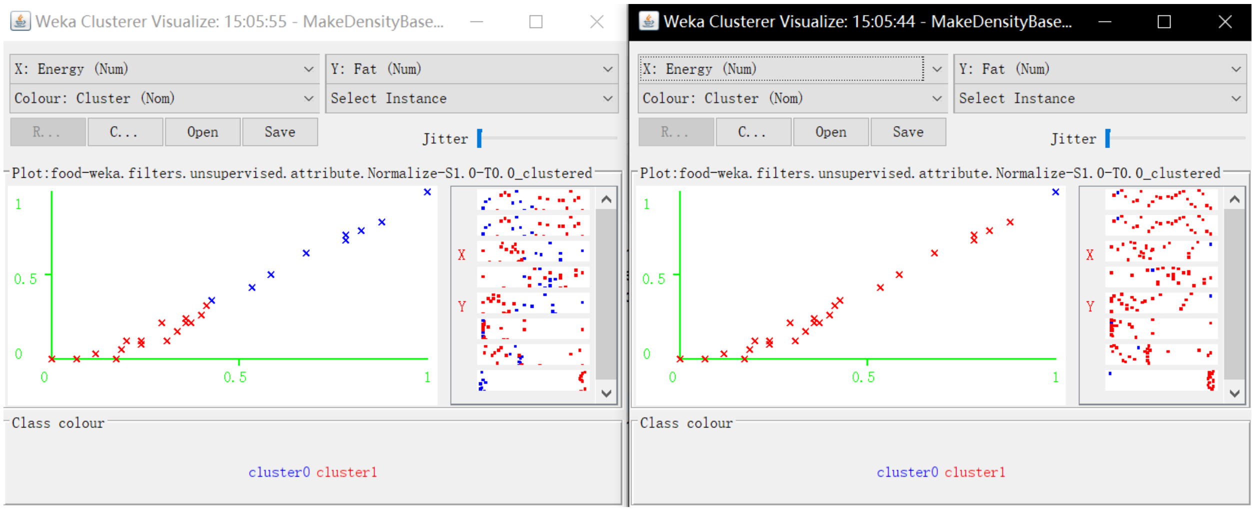
For example (seed=10 k=2), one of the most importances is that the elements in cluster0 (blue) the elements are having more energy and fat, including Hamburger, Roast lamb leg, Roast beef, etc. We can call this cluster as high fat and energy level foods.

Compared with roast and smoked food, the canned and fried food have less energy and fat. They are included in cluster1(red). We can call them low fat and energy level foods.

MakeDensityBasedClusters

Experiment with at least two different standard deviations. Compare the results. (Hint: Increasing the standard deviation to higher values will make the differences in different runs more obvious and thus it will be easier to conclude what the parameter does)

MinStdDev: 1 vs. 0.001



Weka Clusterer Visualize: 15:05:55 - MakeDensityBase...

X: Energy (Num)

Y: Fat (Num)

Colour: Cluster (Nom)

Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

Class colour

cluster0 cluster1

Weka Clusterer Visualize: 15:05:44 - MakeDensityBase...

X: Energy (Num)

Y: Fat (Num)

Colour: Cluster (Nom)

Select Instance

Plot:food-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0_clustered

Class colour

cluster0 cluster1

MinStdDev represents the minimum allowable standard deviation the clustering can bear. It can connect the close instances easily and find the clusters of different shapes and sizes. Since DBSCAN method is sensitive, the final clusters might conclude all the instance as a group besides outliers when MinStdDev is very large.