First we will work with the dataset “food” including nutrient levels of 27 kinds of food.

At the beginning we implement SimpleKmeans on it with 2 clusters and attributes ”fat”, “protain” and “energy” . The motivation is that we can see from the dataset that there are meats and seafoods, which meats may have a higher fat and energy and seafood may have a higher protein, hope those attributes can help us classify them.

Attribute “name” is always ignored because it is merely a tag and don’t provide actual information to help us classify.

Result:

=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Relation: food

Instances: 27

Attributes: 6

Energy

Protein

Fat

Ignored:

Name

Calcium

Iron

Test mode:evaluate on training data

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

kMeans

======

Number of iterations: 2

Within cluster sum of squared errors: 2.156533989945624

Missing values globally replaced with mean/mode

Cluster centroids:

Cluster#

Attribute Full Data 0 1

(27) (9) (18)

============================================

Energy 207.4074 331.1111 145.5556

Protein 19 19 19

Fat 13.4815 27.5556 6.4444

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

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

Clustered Instances

0 9 ( 33%)

1 18 ( 67%)

图形用户界面, 应用程序, Word

描述已自动生成

From the plots we can see that in general our goal is achieved but few meats with low fat are classified into another cluster which contains more seafoods, such as chicken, beef heart etc. And the protein doesn’t seems to work as we expected because most samples we have they have a similar protein level whether they are meat or fish(as we can see the low stddev of protein).

And from the energy-vs-fat plot we can see that this 2 attributes are highly correlated so maybe we can consider that only keep 1 of them as a choice

Now we try using cluster = 5 instead of 2

Result:

=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Relation: food

Instances: 27

Attributes: 6

Energy

Protein

Fat

Ignored:

Name

Calcium

Iron

Test mode:evaluate on training data

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

kMeans

======

Number of iterations: 5

Within cluster sum of squared errors: 0.6428477815396192

Missing values globally replaced with mean/mode

Cluster centroids:

Cluster#

Attribute Full Data 0 1 2 3 4

(27) (6) (9) (5) (4) (3)

=============================================================================

Energy 207.4074 361.6667 156.1111 92 188.75 270

Protein 19 18.6667 23.1111 13 17.25 19.6667

Fat 13.4815 31 6.1111 2.8 11.75 20.6667

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

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

Clustered Instances

0 6 ( 22%)

1 9 ( 33%)

2 5 ( 19%)

3 4 ( 15%)

4 3 ( 11%)

From the results we can see that cluster 1 and 3 have similar centroid values which indicates that it might be better to merge them together

From the plot we can see that there isn’t much useful information explicitly shown. Seems only 3 attributes we chose don’t works well for clustering into 5 clusters, for the reason that 2 attributes(fat and energy) are correlated and the left 1(protein) don’t have significant differences in general. So it may yield a similar result as using only fat to divide 5 clusters.

Now we try the same setting as cluster = 5 but set a different seed 20 instead of 10. Now we can see that the result is completely different. The seed value control the randomness of the initial points choosing. Under kmeans we are using ,the initial points influents a lot when we have 5 clusters to divide into, may be 5 clusters are too much under this dataset and the attributes we chose.

Result:

=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 20

Relation: food

Instances: 27

Attributes: 6

Energy

Protein

Fat

Ignored:

Name

Calcium

Iron

Test mode:evaluate on training data

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

kMeans

======

Number of iterations: 3

Within cluster sum of squared errors: 0.543344174568976

Missing values globally replaced with mean/mode

Cluster centroids:

Cluster#

Attribute Full Data 0 1 2 3 4

(27) (2) (4) (5) (7) (9)

=============================================================================

Energy 207.4074 57.5 125 222 352.8571 156.1111

Protein 19 9 15.75 18.8 18.5714 23.1111

Fat 13.4815 1 5.25 15 30.1429 6.1111

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

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

Clustered Instances

0 2 ( 7%)

1 4 ( 15%)

2 5 ( 19%)

3 7 ( 26%)

4 9 ( 33%)

4. as explained above ,clustering 2 clusters is relatively good because it generally divided data points into meat and seafoods, but clusters = 5 doesn.t seems to be good and heavily susceptible to the changing of initial points.

5. in out first attempt with 2 clusters preset. We can label 1 as meats and another as seafoods nonetheless we have a considerable misclassification rate in label “seafoods”

MakeDensityBasedClusters:

Mindev = 1.0E-6

Result:

=== Run information ===

Scheme:weka.clusterers.MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Relation: food

Instances: 27

Attributes: 6

Energy

Protein

Fat

Ignored:

Name

Calcium

Iron

Test mode:evaluate on training data

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

MakeDensityBasedClusterer:

Wrapped clusterer:

kMeans

======

Number of iterations: 2

Within cluster sum of squared errors: 2.156533989945624

Missing values globally replaced with mean/mode

Cluster centroids:

Cluster#

Attribute Full Data 0 1

(27) (9) (18)

============================================

Energy 207.4074 331.1111 145.5556

Protein 19 19 19

Fat 13.4815 27.5556 6.4444

Fitted estimators (with ML estimates of variance):

Cluster: 0 Prior probability: 0.3448

Attribute: Energy

Normal Distribution. Mean = 331.1111 StdDev = 50.9781

Attribute: Protein

Normal Distribution. Mean = 19 StdDev = 1.633

Attribute: Fat

Normal Distribution. Mean = 27.5556 StdDev = 6.0939

Cluster: 1 Prior probability: 0.6552

Attribute: Energy

Normal Distribution. Mean = 145.5556 StdDev = 44.9348

Attribute: Protein

Normal Distribution. Mean = 19 StdDev = 4.9777

Attribute: Fat

Normal Distribution. Mean = 6.4444 StdDev = 3.9892

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

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

Clustered Instances

0 9 ( 33%)

1 18 ( 67%)

Log likelihood: -11.49331

When we set mindev as 100 and 200 we can see that 1 of the cluster is became more and more comprehensive. Which we can conjecture that the large mindev value allow the cluster have larger range and finally 1 cluster devoured all of the data points.