lab01: Clustering Lab

Group 18

February 11, 2023

1 SimpleKmeans

1.1 Choose a set of attributes for clustering and give a motivation.

First we will work with the dataset "food" including nutrient levels of 27 kinds of food (see plot1). At the beginning we implement SimpleKmeans on it with 2 clusters and attributes "fat", "protein" 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.

No.	Name String	Energy Numeric	Protein Numeric	Fat Numeric	Calcium Numeric	**Iron** Numeric
1	Braised beef	340.0	20.0	28.0	9.0	2.6
2	Hamburger	245.0	21.0	17.0	9.0	2.7
3	Roast beef	420.0	15.0	39.0	7.0	2.0
4	Beefsteak	375.0	19.0	32.0	9.0	2.6
5	Canned beef	180.0	22.0	10.0	17.0	3.7
6	Broiled chicken	115.0	20.0	3.0	8.0	1.4
7	Canned chicken	170.0	25.0	7.0	12.0	1.5
8	Beef heart	160.0	26.0	5.0	14.0	5.9
9	Roast lamb leg	265.0	20.0	20.0	9.0	2.6
10	Roast lamb shoulder	300.0	18.0	25.0	9.0	2.3
11	Smoked ham	340.0	20.0	28.0	9.0	2.5
12	Pork roast	340.0	19.0	29.0	9.0	2.5
13	Pork simmered	355.0	19.0	30.0	9.0	2.4
14	Beef tongue	205.0	18.0	14.0	7.0	2.5
15	Veal cutlet	185.0	23.0	9.0	9.0	2.7
16	Baked bluefish	135.0	22.0	4.0	25.0	0.6
17	Raw clams	70.0	11.0	1.0	82.0	6.0
18	Canned clams	45.0	7.0	1.0	74.0	5.4
19	Canned crabmeat	90.0	14.0	2.0	38.0	0.8
20	Fried haddock	135.0	16.0	5.0	15.0	0.5
21	Broiled mackerel	200.0	19.0	13.0	5.0	1.0
22	Canned mackerel	155.0	16.0	9.0	157.0	1.8
23	Fried perch	195.0	16.0	11.0	14.0	1.3
24	Canned salmon	120.0	17.0	5.0	159.0	0.7
25	Canned sardines	180.0	22.0	9.0	367.0	2.5
26	Canned tuna	170.0	25.0	7.0	7.0	1.2
27	Canned shrimp	110.0	23.0	1.0	98.0	2.6

Figure 1: dataset:food

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

1.2 Experiment with at least two different numbers of clusters, e.g. 2 and 5, but with the same seed value 10.(Hint: always ignore attribute "name". Why does the name attribute need to be ignored?)

1.2.1 2 clusters:

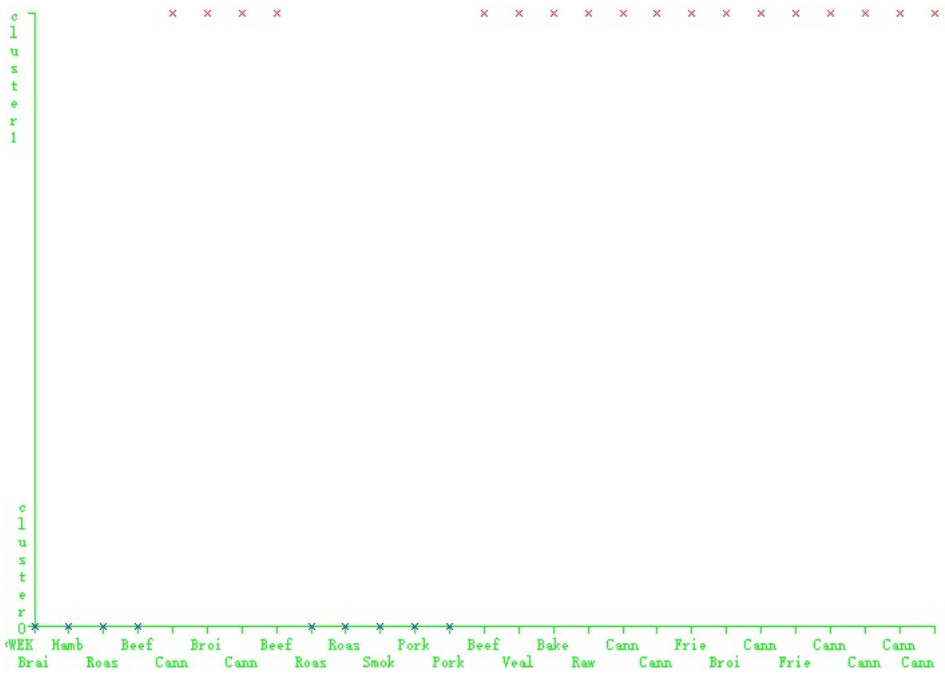

Figure 2: nametag vs cluster result

From the plot2 we can see that in general our goal is achieved but few kinds of meats with low fat&energy 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, see plot3 and 4).

Selected attribute

Name: Protein | Type: Numeric

Missing: 0 (0%) | Distinct: 14 | Unique: 7 (26%)

Statistic	Value
Minimum	7
Maximum	26
Mean	19
StdDev	4.252

Figure 3: attribute summary: protein

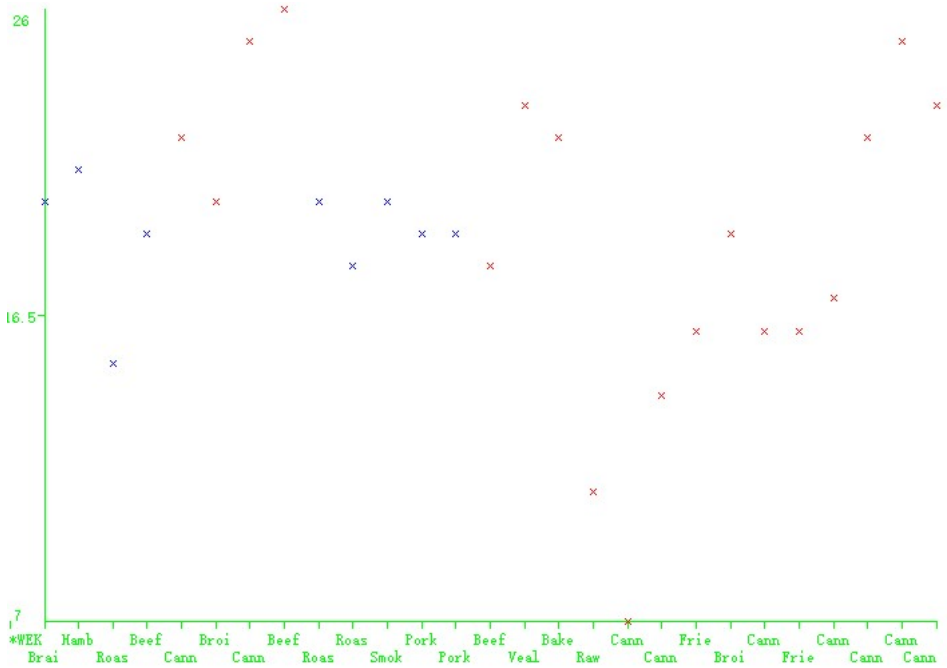

Figure 4: nametag vs protein

And from the energy-vs-fat plot7 with plot5 and 6 we can see that this 2 attributes are highly correlated so maybe we can consider that only keep 1 attribute of them as a choice

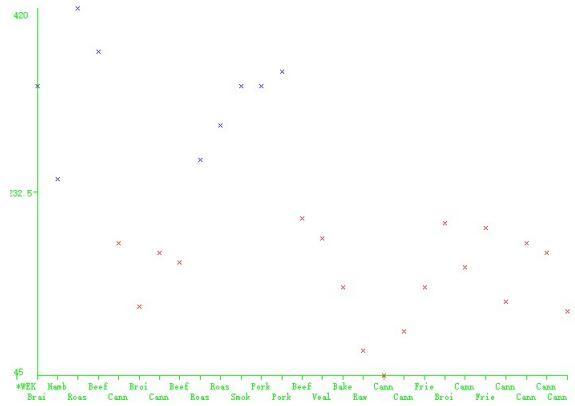

Figure 5: nametag vs energy

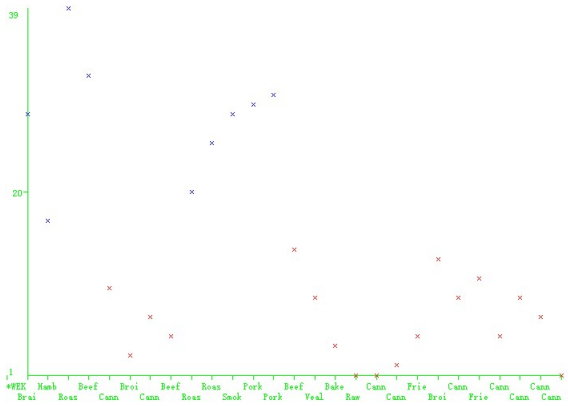

Figure 6: nametag vs fat

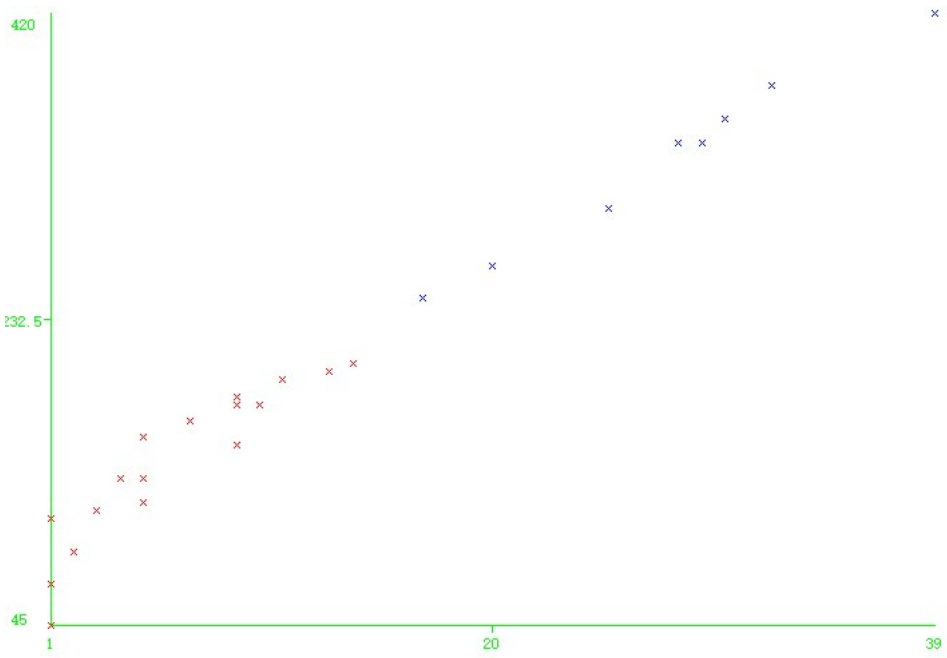

Figure 7: fat vs energy

1.2.2 5 clusters:

Now we try using cluster = 5 instead of 2:

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

Figure 8: centroids of 5 clusters

From the results8 we can see that cluster 1 and 3 have similar centroid values which indicates that it might be better to merge them together.

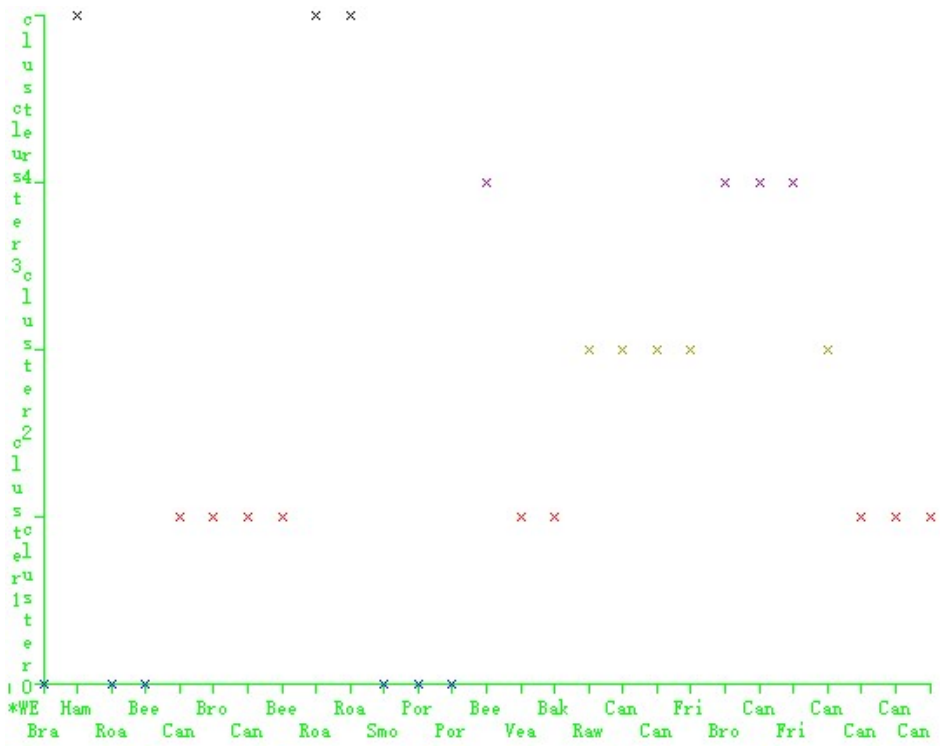

Figure 9: nametag vs cluster result under 5 clusters

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 other 1 attribute(protein) don't have significant differences in general. So it may yield a similar result as using only fat to divide 5 clusters.

1.3 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.

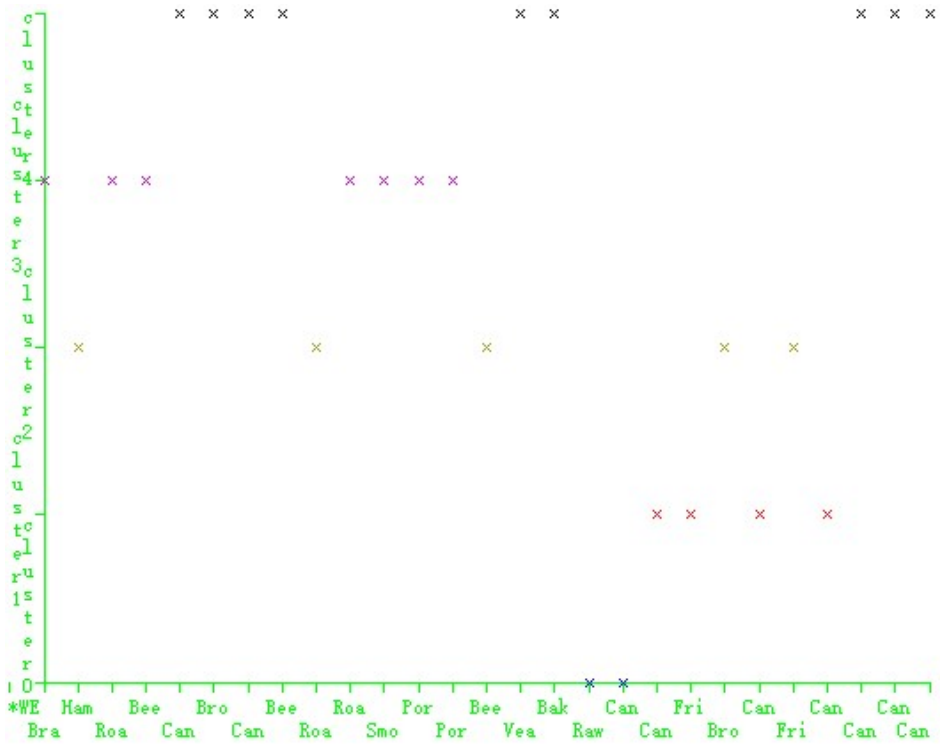

Figure 10: nametag vs cluster result under 5 clusters with seed 20

Now we try the same setting as cluster = 5 but set a different seed 20 instead of 10. Now we can see from the plotthat the result is completely different. The seed value control the randomness of the initial points choosing. Under kmeans we are using(with this dataset and attributes we chose) ,the initial points influences a lot when we have 5 clusters to divide into and obviously there isn't any same minimum which can easily be find by the algorithm. This can be reasoned by the conjecture we made, which is that out three attributes may not be able to provide more information than choosing solely 1 attribute energy/fat for classification so the lack of information provided brings the failure of effective clustering.

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

As discussed above ,clustering 2 clusters is relatively good because it generally divided data points into meat and seafoods as what we expected. But clusters = 5 doesn.t seems to be good and heavily susceptible to the changing of initial points which makes the clustering result highly randomly, both the result with different seed don't have much similarity within clusters.

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

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"

2 MakeDensityBasedClusters

2.1 Use the SimpleKMeans clusterer which gave the result you haven chosen in 5).

We implement MakeDensityBasedClusters with mindev = 1e-6 based on SimpleKMeans clusster = 2(the first clustering in this report).

The clustering result is shown by the plot11, and we can see that it's exactly the same with SimpleKMeans without MakeDensityBasedClusters, the clustering result hasn'e been changed at all.

Figure 11: name vs cluster, mindev = 1e-6

2.2 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)

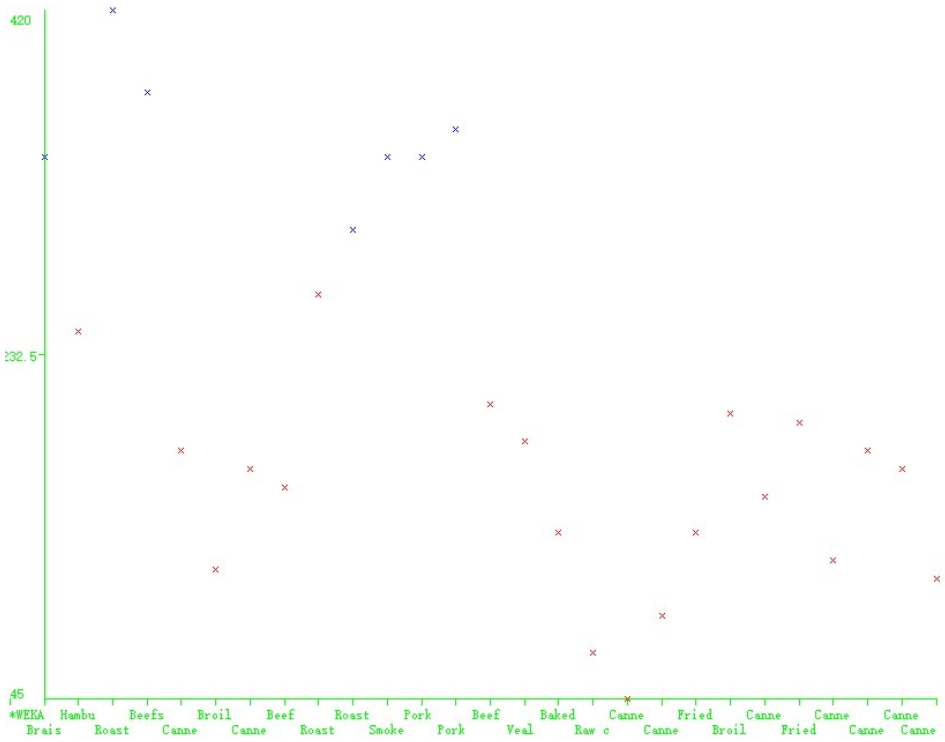

Figure 12: name vs cluster, mindev = 100

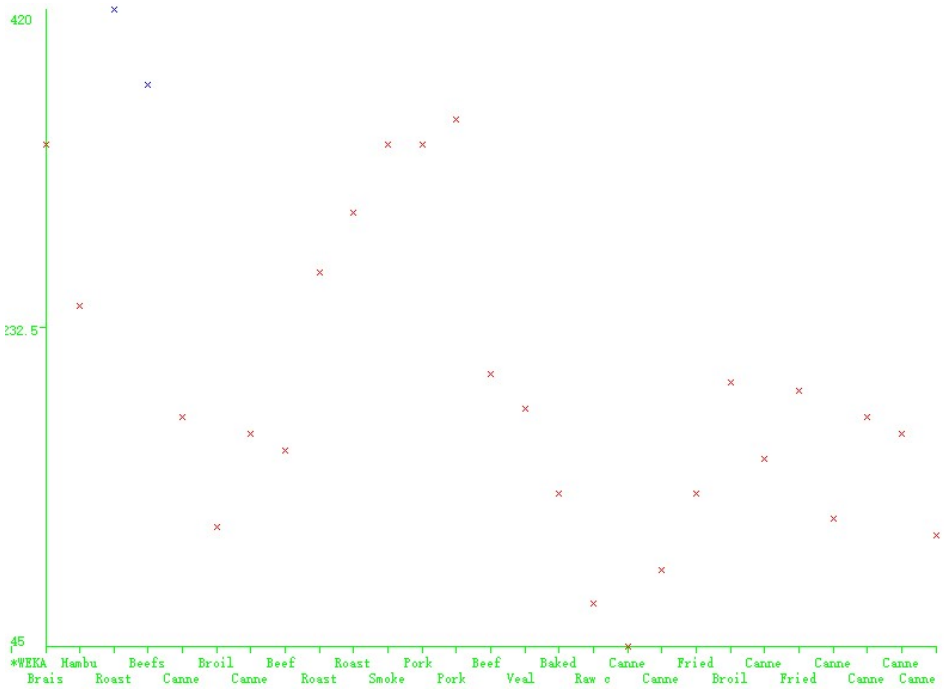

Figure 13: name vs cluster, mindev = 200

When we set mindev as 100(p12) and 200(p13) we can see that 1 of the cluster is became more and more comprehensive when mindev rises. Which we can conjecture that the large mindev value allow the cluster have larger range and finally 1 cluster will devour all of the data points if we set the mindev even higher.