Homework 5

Jian Wang

Part 1.

A. Basically, the program was written by java. The overall design includes four classes. They are class Kmean, class Point, class Cluster, class Statistics. Class Kmean has all the information about k-mean algorithm which will be discussed later. It also includes the implementation of normalization, computing the label of each cluster and computing the confusion matrix. Data preprocessing in the kmean class includes read csv file from the disk. For data set 1 and 2, I eliminate some redundant symbols and for wine data, I use class label low or high to calculate the confusion matrix and use quality label to make validation in problem F. Overall, it deals with reading data and normalizing data using min-max normalization discussed in class. Then I implemented k-mean here. The algorithm is below.

1. Randomly select k points from the dataset as the initial centroids. Here I randomly choose k points from input dataset which I think is a reasonable way to initialize the k-mean.
2. Repeat
   a) From k clusters by assigning all points to the closest centroid
   b) Recompute the centroid of each cluster by computing the mean of all points in that cluster
3. Until the centroid do not change. I compute the Manhattan distance of the previous centroid and the current centroid and if the error distance is less than 10^-8 then I will stop my algorithm.

I select the majority label of all the points in that cluster as the label of that cluster which indicates that the predicted label of all the points of that cluster is the label of that cluster.

Point class is the class that described the points and its attributes like which cluster it belongs to and predicted and true label of that points.

Statistics class implements SSE, SSB and Average silhouette computation. Overall, I use Euclidean distance as described in the description.

Finally, cluster class is a class of cluster and has a label, all points in that cluster and centroid and updating the centroids methods. Change is used for determining if centroid changes

B. Here is the result for true clustering.

a. Dataset 1

True clustering

Cluster 1 SSE = 0.9343135635537495

Cluster 2 SSE = 6.723954566951231

Total k cluster SSE = 7.658268130504981

Total SSB = 30.453557381589476

b. Dataset 2

True clustering

Cluster 1 SSE = 0.43549863243055537

Cluster 2 SSE = 1.2587733447181861

Cluster 3 SSE = 3.3505361091694246

Cluster 4 SSE = 2.65147771492812

Total k cluster SSE = 7.696285801246287

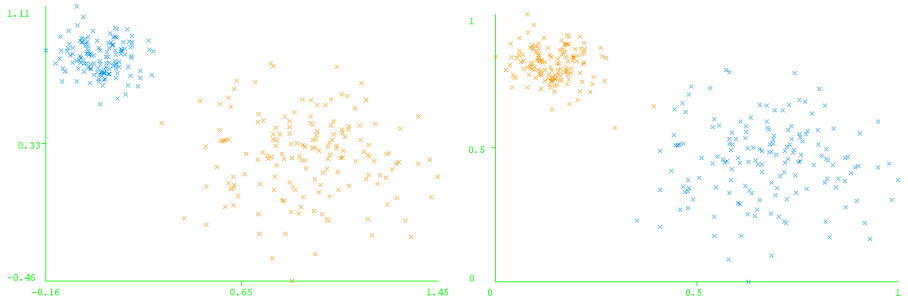
Total SSB = 32.971741043892074

C. Run k-means 3 different times using k = 2 and k = 4

For the 1st time

For dataset 1:

Cluster 1 SSE	1.101342	
Cluster 2 SSE	6.450015	
Total k cluster SSE	7.551357	
Total SSB	30.56047	
Silhouette in 1 cluster	0.829342	
Silhouette in 2 cluster	0.595814	
Total Silhouette	0.712578	
Confusion Matrix	Label 1	Label 2
Label 1	138	0
Label 2	2	160



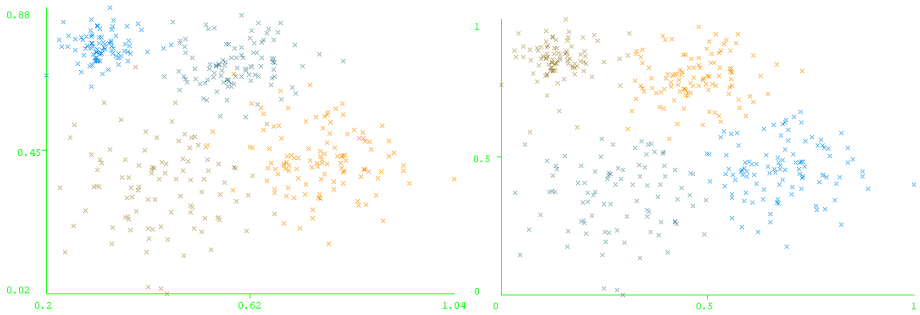
Dataset1 True clustering k-mean clustering k = 2

As you can see there are two points in two figures that is classified differently. The color of the plots means that cluster 1 has label 2 and cluster 2 has label 1. We can clearly see the two wrong prediction in the scatter plots. The scatter plots will not be changed for dataset 1 except the switch of color for the next two times, thus will not be presented.

For dataset 2:

Cluster 1 SSE	2.041977			
Cluster 2 SSE	2.549086			
Cluster 3 SSE	0.691742			
Cluster 4 SSE	1.496639			
Total k cluster SSE	6.779445			
Total SSB	33.88858			
Silhouette in 1 cluster	0.521091			
Silhouette in 2 cluster	0.445583			
Silhouette in 3 cluster	0.696055			
Silhouette in 4 cluster	0.514001			
Total Silhouette	0.544182			
Confusion Matrix	Label 1	Label 2	Label 3	Label 4

Label 1	89	0	0	0
Label 2	2	98	0	0
Label 3	4	2	88	3
Label 4	0	8	2	104



Dataset2 True clustering k-mean clustering k = 4

The difference of color indicates the representation of labels and clusters as discussed before. Generally for dataset 2, it is worse than data set 1, since some points are overlapping and for different initializations, there are different results. See the difference among three times.

For the 2nd time:

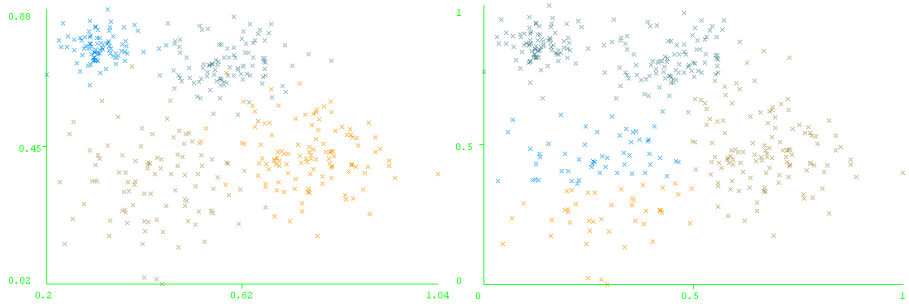
For dataset 1:

Cluster 1 SSE	1.101342	
Cluster 2 SSE	6.450015	
Total k cluster SSE	7.551357	
Total SSB	30.56047	
Silhouette in 1 cluster	0.829342	
Silhouette in 2 cluster	0.595814	
Total Silhouette	0.712578	
Confusion Matrix	Label 1	Label 2
Label 1	138	0
Label 2	2	160

For dataset 2:

Cluster 1 SSE	0.904036			
Cluster 2 SSE	6.00468			
Cluster 3 SSE	3.118141			
Cluster 4 SSE	0.837768			
Total k cluster SSE	10.86462			
Total SSB	29.8034			
Silhouette in 1 cluster	0.366216			
Silhouette in 2 cluster	0.442292			
Silhouette in 3 cluster	0.47976			
Silhouette in 4 cluster	0.322901			
Total Silhouette	0.402792			
Confusion	Label 1	Label 2	Label 3	Label 4

Matrix				
Label 1	0	89	0	0
Label 2	0	90	1	9
Label 3	0	3	91	3
Label 4	0	1	3	110



Dataset2 True clustering k-mean clustering k = 4

For the 3rd time:

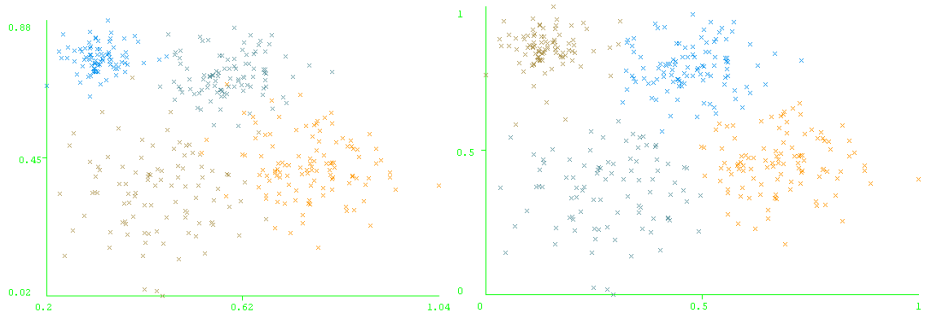
For dataset 1:

Cluster 1 SSE		1.101342	
Cluster 2 SSE		6.450015	
Total k cluster SSE		7.551357	
Total SSB		30.56047	
Silhouette in 1 cluster		0.829342	
Silhouette in 2 cluster		0.595814	
Total Silhouette		0.712578	
Confusion Matrix	Label 1		Label 2
Label 1	138		0
Label 2	2		160

For dataset 2:

Cluster 1 SSE		1.435868		
Cluster 2 SSE		2.609513		
Cluster 3 SSE		0.691742		
Cluster 4 SSE		2.041977		
Total k cluster SSE		6.779101		
Total SSB		33.88893		
Silhouette in 1 cluster		0.521462		
Silhouette in 2 cluster		0.439721		
Silhouette in 3 cluster		0.696179		
Silhouette in 4 cluster		0.520966		
Total Silhouette		0.544582		
Confusion Matrix	Label 1	Label 2	Label 3	Label 4
Label 1	89	0	0	0

Label 2	2	98	0	0
Label 3	4	1	89	3
Label 4	0	8	2	104



Dataset2 True clustering k-mean clustering k = 4

5 my observation on these metrics vary as the initial centroids are changed is that for data set 1, the metrics don't change when I change the initial centroids randomly. For data set 2, the metrics can change as the initial centroids change. We can see that for time 1 and 3, they are almost the same, but just some little difference in each Cluster SSE and confusion matrix. For time 2, it has much more different metrics from the other two. Each cluster SSE, total SSE, total SSB, each cluster Silhouette and confusion matrix are all very different compare to the others.

For both datasets, as I have mentioned before, the same clustering will have a different permutation of the others for different times and thus will have different colors of graphs as shown above.

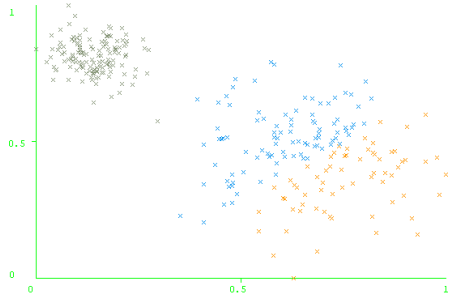
D. K = 3 for dataset 1 and 2:

For dataset 1:

Cluster 1 SSE	2.125326	
Cluster 2 SSE	1.224711	
Cluster 3 SSE	2.13544	
Total k cluster SSE	5.485477	
Total SSB	32.62635	
Silhouette in 1 cluster	0.3492	
Silhouette in 2 cluster	0.806816	
Silhouette in 3 cluster	0.275201	
Total Silhouette	0.477072	
Confusion Matrix	Label 1	Label 2
Label 1	138	0
Label 2	3	159

Cluster 1 SSE	2.583242
Cluster 2 SSE	1.014542
Cluster 3 SSE	1.821543
Total k cluster SSE	5.419328
Total SSB	32.6925

Silhouette in 1 cluster	0.298971	
Silhouette in 2 cluster	0.807747	
Silhouette in 3 cluster	0.299124	
Total Silhouette	0.468614	
Confusion Matrix	Label 1	Label 2
Label 1	138	0
Label 2	1	161



this figure is for dataset 1

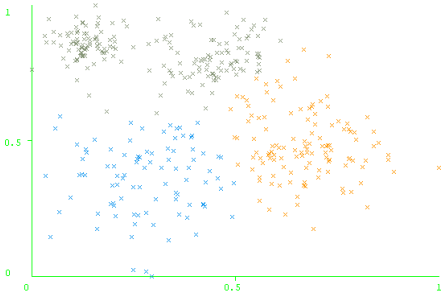
K mean clustering k = 3

For dataset 2:

Cluster 1 SSE	2.707295			
Cluster 2 SSE	6.180523			
Cluster 3 SSE	3.118141			
Total k cluster SSE	12.00596			
Total SSB	28.66207			
Silhouette in 1 cluster	0.458734			
Silhouette in 2 cluster	0.5023			
Silhouette in 3 cluster	0.504487			
Total Silhouette	0.488507			
Confusion Matrix	Label 1	Label 2	Label 3	Label 4
Label 1	0	89	0	0
Label 2	0	91	0	9
Label 3	0	5	89	3
Label 4	0	1	3	110

Cluster 1 SSE	10.8986
Cluster 2 SSE	1.620809
Cluster 3 SSE	2.149967
Total k cluster SSE	14.66938
Total SSB	25.99865
Silhouette in 1 cluster	0.289922
Silhouette in 2 cluster	0.629601
Silhouette in 3 cluster	0.491728
Total Silhouette	0.470417

Confusion Matrix	Label 1	Label 2	Label 3	Label 4
Label 1	89	0	0	0
Label 2	2	98	0	0
Label 3	11	3	0	83
Label 4	0	17	0	97



This figure is for dataset 2.

K mean clustering k = 3

As we can see, changing the number of clusters in this data set 1 will not change much and the performance could get better or worse. One reason could be the performance of this data set related to k-mean methods k = 2 is very good. Increasing 1 to k will not change much

As we can see, changing the number of clusters in this data set 2 could change the performance and metrics and the performance will get worse generally. One reason is because we have four labels but have only three clusters. It causes that no prediction could be made by one of the labels.

For wine data,

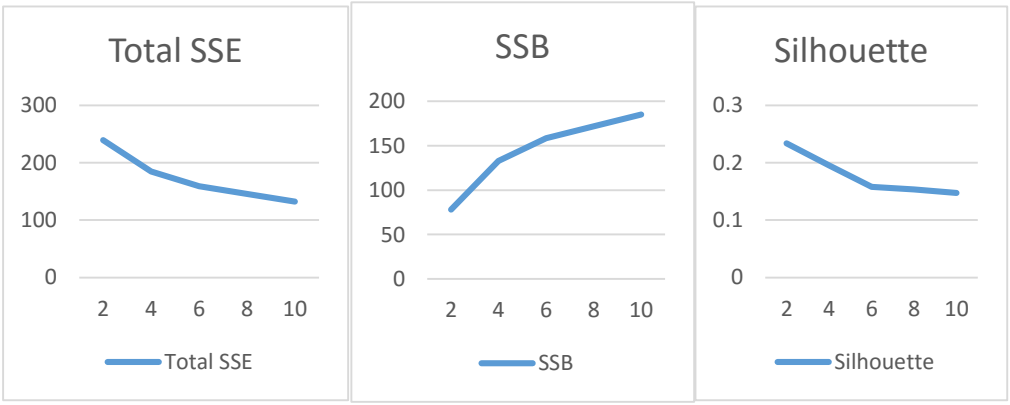
E. See table below

K	Total SSE	SSB	Silhouette
2	239.19	78.19	0.2338
4	184.47	132.90	0.1953
6	158.79	158.59	0.1580
8	145.67	171.69	0.1535
10	132.34	185.03	0.1475

We can see that as K increase, total SSE decreases, SSB increases and entire Silhouette decreases. However, this is because the number of clusters increases reduce the total SSE and Silhouette, but increase SSB. As we know that SSE + SSB will be a constant, therefore we can see the change.

F. Provide detailed analysis of the results

First I will show the graph of different k with respect to different SSE, SSB, and Silhouette.



Then Confusion matrices for different K (note I use low or high class to do the confusion matrix):

Confusion	(k=2, 8)L	(k=2, 8)H	(k=4)L	(k=4)H	(k=6, 10)L	(k=6, 10)H
(k=2, 4, 6)L	521	223	590	154	603	141
(k=2, 4, 6)H	422	433	382	473	366	489
(k=8, 10)L	566	178			602	142
(k=8, 10)H	326	529			355	500

Basically, In confusion matrix, we can observe that as the number of clusters increases from 2-6, the accuracy increases but after 6, see 6-10, the accuracy do not change much. However, k = 10 will have a little better performance than others so I prefer to choose k = 10.

G. Use the quality attribute for external validation.

Here is the confusion matrix I computed. I use quality attribute and use k = 10.

Confusion	3	4	5	6	7	8
3	0	0	5	5	0	0
4	0	0	30	23	0	0
5	0	0	520	161	0	0
6	0	0	275	363	0	0
7	0	0	23	176	0	0
8	0	0	1	17	0	0

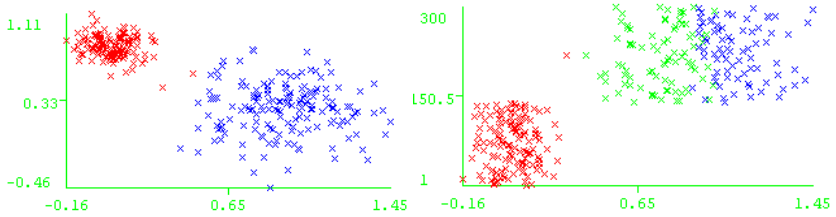
Thus, Accuracy is (520 + 363) / 1599 = 0.5522. Since there are 6 classes of quality, the performance is not bad. Since I use majority selection on assigning labels of clusters and 5, 6 are the most popular quality of the wine (in true clusters) thus all labels of the prediction will be chosen between 5 and 6. (note quality range from 3 to 8)

Part2

1. I use k-mean cluster in weka. There are many parameters in weka that we can control for example, number of iteration, canopyT1, canopyT2, number of clusters, and distance functions and so on. I make all default except the number of clusters. The distance function will be Euclidean like part 1. Since weka don't have SSB
2. Here is the table of results for each dataset

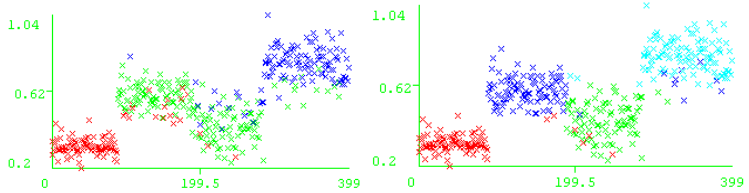
   For dataset 1:

K	SSE1	SSE2	SSE3	Total SSE	SSB	Silhouette1	Silhouette2	Silhouette3	Silhouette
2	6.4500	1.1013		7.5513	30.5605	0.5958	0.8293		0.7125
3	1.1015	1.7025	2.7842	5.5013	32.6104	0.8154	0.3813	0.2290	0.4753



For dataset 2:

K	SSE1	SSE2	SSE3		Total SSE	SSB	Silhouette1	Silhouette2	Silhouette3		Silhouette
3	1.5958	4.2843	10.8181		16.6983	23.9697	0.6487	0.4624	0.1370		0.4159
4	0.6917	1.4359	2.6095	2.0419	6.7791	32.5678	0.6961	0.5214	0.4397	0.5209	0.5446



For dataset 3 (k = 6):

Cluster 1 SSE	17.51564	Silhouette in 1 cluster	0.262559
Cluster 2 SSE	23.45758	Silhouette in 2 cluster	0.124627
Cluster 3 SSE	18.84035	Silhouette in 3 cluster	0.130337
Cluster 4 SSE	40.03864	Silhouette in 4 cluster	0.05104
Cluster 5 SSE	33.68411	Silhouette in 5 cluster	0.15475
Cluster 6 SSE	25.25051	Silhouette in 6 cluster	0.222715
Total k cluster SSE	158.7868	Total Silhouette	0.157671
Total SSB	158.5866		

3. I compare this result to my implementation of k mean in part 1. I found that for data set 1 and 3, the metrics just change a little. My implementation and weka's are very similar. All the similar values to my implementation of k mean are marked as red. For data set 2, the metrics change more relatively. But absolutely, the metrics do not change much, therefore the metrics of my implementation and weka's are still very similar. One reason could be that I have done some data preprocessing like normalization but weka does not. Generally, the results agree.