CSE 5243 Homework 5: Clustering

Yiran Cao(cao.805)

1 Design and Implementation

1.1 Dataset Inspection and Data Preprocessing

Given the datasets, the first thing to do is checking for missing values and doing some exploratory data analysis to see if there is any necessity for data preprocessing.

Fortunately, the three datasets, 2 two-dimensional datasets and the wine dataset, are composed of of numeric attributes and with no missing values.

For the two dimensional datasets, since the two attributes of each data point can be regarded as a position vector, indicating the position of that point on a 2D plane, I did not do any transformation on these attributes. For the wine dataset, however, the attributes mean differently and probably have different scales, so I standardized each attribute of the wine dataset before clustering.

1.2 Clustering Function

The main functionality of this project is implemented in the clustering function, which takes two parameters, the dataset and the number of clusters. The following pseudo-code briefly explains how this function works.

Algorithm 1 Clustering

```
procedure CLUSTERING(dataset, k)
    Initialization:
    $centroids \leftarrow$ randomly select $k$ points from dataset
    while $dist > threshold$ and $iter < maxIter$ do
        for each centroid do
            assign points that's nearest to it to its cluster
        for each cluster do
            assign a new centroid, which is the mean of all points in this cluster
        $dist \leftarrow$ distance vector between the old and new centroids
        $iter \leftarrow iter + 1$
```

The clustering function initially assigns k random points as centroids, then keeps upating the centroids until the old and new centroids are close enough (distance less than a small threshold), or the maximum number of iterations is reached.

I apply this function to the two-dimensional datasets and also the wine dataset, with additional standardization on the wine dataset.

2 Results and Analysis on the Two-Dimensional Dataset

2.1 Dataset 1 – Two-Dimensional Data with 2 Clusters

2.1.1 Results

The clustering algorithm works well on dataset1, which is a two-dimensional dataset with two true clusters. With k=2, the k-means algorithm I implemented made 2 errors over the total number of

300 records.

The scatter plots, Figure 1 and Figure 2, demonstrate the clustering results and the comparison with true clusters.

Figure 1 shows the clustering result by the clustering algorithm I implemented compared with true clusters. . Figure 2 makes the comparison, where the crosses are the misclustered points.

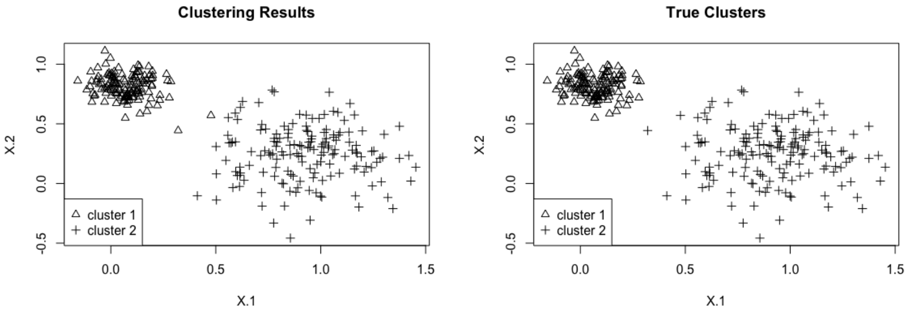

Figure 1: Clustering Results & True Clusters

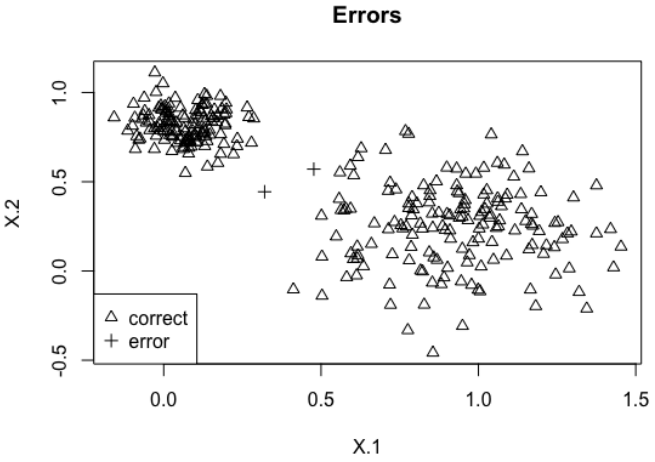

Figure 2: Clustering Errors

Table 1 gives an cross tabulation matrix comparing the actual and assigned clusters, from which we can see that only 2 points in true cluster 2 are misclustered into cluster 1 by the algorithm.

	true cluster 1	true cluster 2
cluster 1	138	2
cluster 2	0	160

Table 1: Cross Tabulation Matrix

2.1.2 Cluster Validation

I calculated the SSE and SSB for both true clusters and the clusters got from the algorithm. Results are shown in Table 2 and Table 3.

	true cluster	“predicted” cluster
1	2.36	2.78
2	17	16.3
overall	13.6943	19.35819

Table 2: SSE for True and “Predicted” Clusters

	true cluster	“predicted” cluster
SSB	77.76	78.08

Table 3: SSB for True and “Predicted” Clusters

2.2 Dataset 2 – Two-Dimensional Data with 4 Clusters

2.2.1 Results

For dataset2, the ture clusters are not so well-seperated as dataset1, with overlaps between clusters.

The scatter plots, Figure 3 and Figure 4, demonstrate the clustering results and the comparison with true clusters.

Figure 3 shows the clustering result by the clustering algorithm I implemented compared to the true clusters. Figure 4 makes the comparison, where the crosses are the misclustered points.

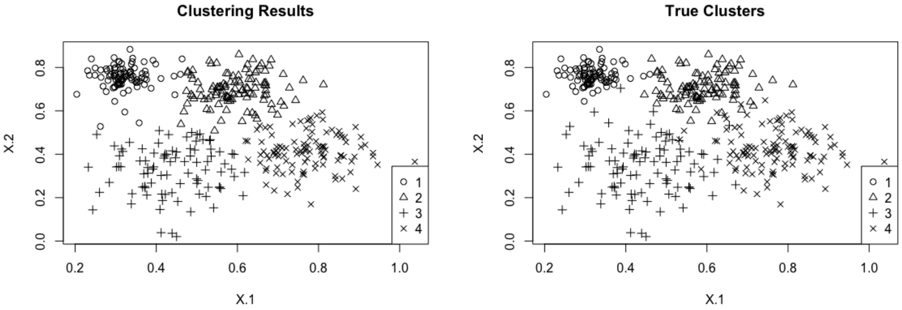

Figure 3: Clustering Results

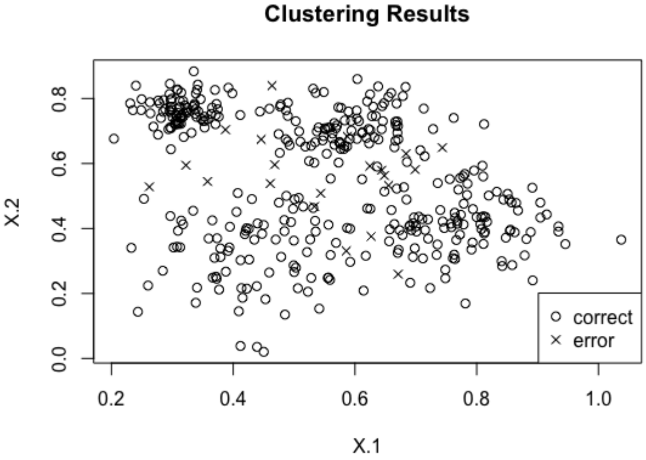

Figure 4: Clustering Errors

Table 4 gives an cross tabulation matrix comparing the actual and assigned clusters. The numbers on the diagonal are points which are correctly clustered, while others are error points. For dataset2, the algorithm made 21 mistakes in total.

	true cluster 1	true cluster 2	true cluster 3	true cluster 4
cluster 1	89	2	4	0
cluster 2	0	98	2	8
cluster 3	0	0	88	2
cluster 4	0	0	3	104

Table 4: Cross Tabulation Matrix

2.2.2 Cluster Validation

I calculated the SSE and SSB for both true clusters and the clusters got from the algorithm. Results are shown in Table 5 and Table 6.

	true cluster	“predicted” cluster
1	0.3128477	0.5004806
2	0.9025336	1.0764851
3	2.4301187	1.8446030
4	1.9107155	1.4705342
overall	4.892103	5.556216

Table 5: SSE for True and “Predicted” Clusters

	true cluster	“predicted” cluster
SSB	23.74813	24.41224

Table 6: SSB for True and “Predicted” Clusters

2.3 Changing the Value of k

2.3.1 Dataset 1

When changing k to$k = 3$, the clustering reults is as shown in Figure 5.

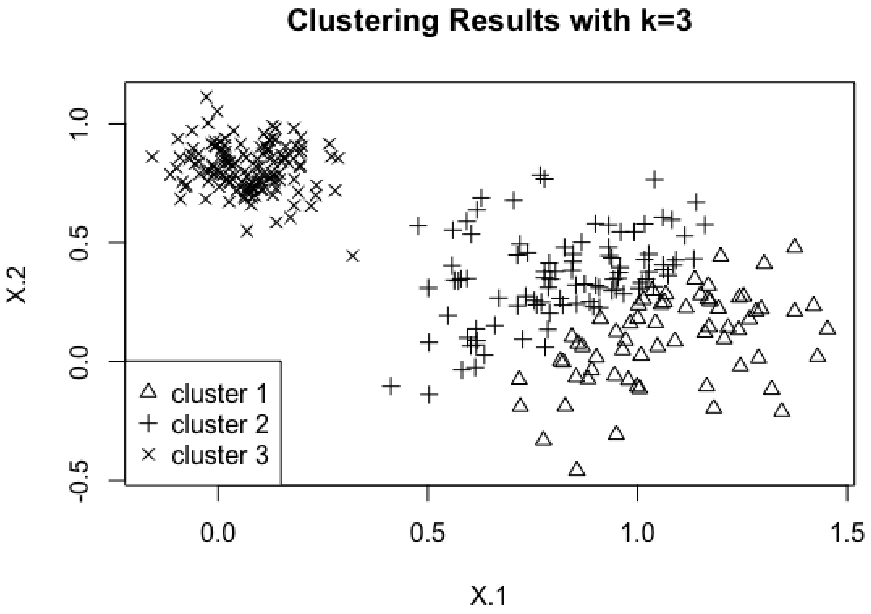

Figure 5: Clustering Results with $k = 3$

With $k = 3$, the SSE of each cluster, the overall SSE and SSB is shown in Table 7

	1	2	3	overall	overall(k=2)	overall(true)
SSE	1.178310	1.506164	15.283697	17.96817	19.358	13.694
SSB	-	-	-	79.1544	78.08	77.76

Table 7: SSE for Dataset 1 Clustering with $k = 3$

The tabulation matrix is shown in Table 8

	true cluster 1	true cluster 2
cluster 1	91	0
cluster 2	47	6
cluster 3	0	156

Table 8: Cross tabulation Matrix on Dataset 1 with $k = 3$

From Table 7 and Table 8 we can learn that, with k increasing from 2 to 3, the overall SSE decreases, which means the clusters are more tighly clustered, on average, than those when $k = 2$. While when $k = 3$, SSB increases, indicating that the clusters are better seperated than $k = 2$.

2.3.2 Dataset 2

When changing k to$k = 3$, the clustering reults for dataset 2 is as shown in Figure 6.

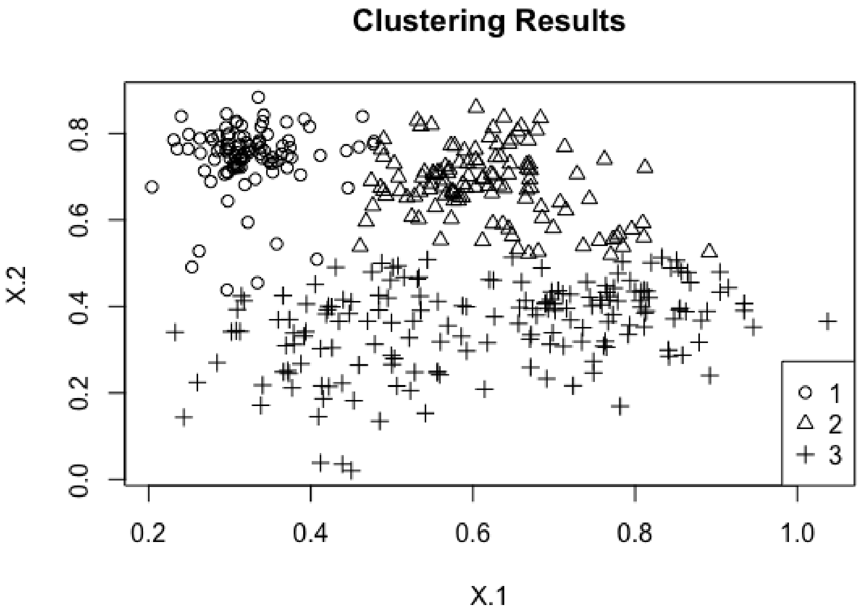

Figure 6: Clustering Results with $k = 3$

With $k = 3$, the SSE of each cluster, the overall SSE and SSB is shown in Table 9

	1	2	3	overall	overall(k=4)	overall(true)
SSE	0.8703093	1.6672667	7.8555954	10.393	5.556	4.892
SSB	-	-	-	18.91117	24.41224	23.74813

Table 9: SSE for Dataset 1 Clustering with $k = 3$

The tabulation matrix is shown in Table 10

	true cluster 1	true cluster 2	true cluster 3	true cluster 4
cluster 1	89	4	8	0
cluster 2	0	96	3	17
cluster 3	0	0	86	97

Table 10: Cross tabulation Matrix on Dataset 1 with $k = 3$

From Table 9 and Table 10 we can learn that, with k decreasing from 4 to 3, the overall SSE increases, which means the clusters are less tighly clustered than those when $k = 4$. While when $k = 3$, SSB decreases, indicating that the clusters are not so well-seperated as when $k = 4$.

3 Results and Analysis on the Wine dataset

3.1 Clustering with Different k values

According to the quality attribute in the wine dataset, which varies from 3 to 8, I vary the value of k from 2 to 9 while clustering the wine dataset. For each different value of k, I calculated the overall SSE and SSB to see which k performs better on this dataset.

The results are shown in Table 11.

	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8	k = 9
SSE	14321.978	12674.973	11452.660	10149.023	9427.378	8960.251	8307.952	8149.213
SSB	3256.022	4903.027	6125.340	7428.977	8150.622	8617.749	9270.048	9428.787

Table 11: SSE and SSB with Different k Values

To show the trends clearly, I plotted Figure 7 and Figure 8.

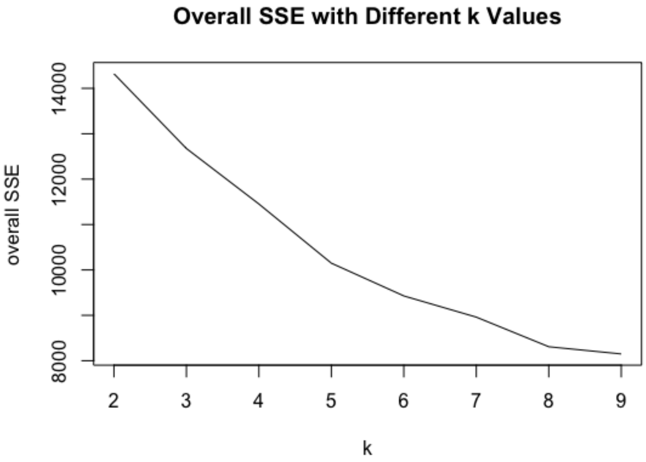

Figure 7:

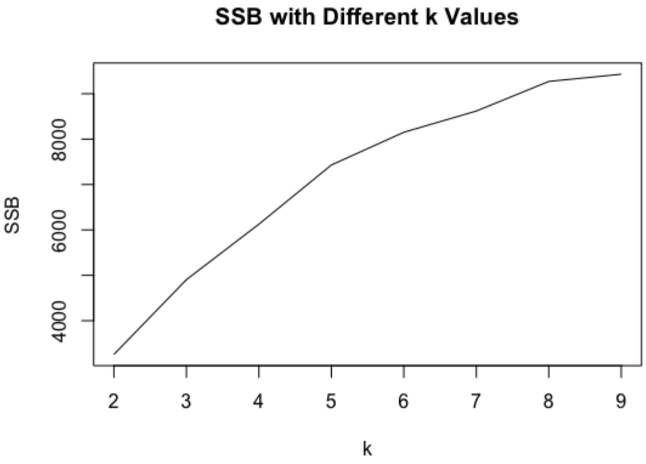

Figure 8:

Since SSE measures the cluster cohesion and SSB measures the cluster sepreation, we would always like to choose a k which gives relatively low SSE and high SSB. According to Figure 7 and Figure 8, when k = 9, the SSE is lowest and meanwhile the SSB is highest. So I would choose $k = 9$ for the wine dataset to do clustering.

	quality 3	quality 4	quality 5	quality 6	quality 7	quality 8
cluster 1	0	2	25	124	88	9
cluster 2	7	30	288	178	20	0
cluster 3	1	6	229	95	11	0
cluster 4	2	4	83	126	54	5
cluster 5	0	10	37	106	25	4
cluster 6	0	1	19	9	1	0

Table 12: Cross Tabulation Matrix with Quality Attribute

3.2 Clutser Validation with External Attribute

Since there are 6 different values for the quality attribute on wine, in this part I choose $k = 6$. However, as it's hard to relabel the clusters to make them agree to the true cluster label, the cross tabulation matrix (Table 12) is not as clear as those of the two-dimensional datasets.

The reason for this non-cleararity may come from the biased data. If looking into the wine dataset, a important discovery would be that, 75% of the records are of quality 5 or 6. So it's difficult for the data points which are not among quality 5 or 6 to cluster with k-means clustering algorithm. Meanwhile, this data is a 11-dimensional dataset. Such a high dimensionality considerably sparse the data, which also makes it difficult for k-means, which tends to form sepherical clusters.

4 Off-the-Shelf Clutering Method

In this section, I used the kmeans in Weka to run on the three datasets.

4.1 Dataset 1

For dataset 1, when $k = 2$, the kmeans function gives identical results with my method.

4.2 Dataset 2

For dataset 2, when $k = 4$, my method outperforms the Weka on SSE and SSB, which are shown in Table 13.

	Yiran	Weka
SSE	5.556	6.779
SSB	24.412	11.106

Table 13: Comparison with Weka on Dataset 2

4.3 Wine Dataset

For the wine dataset, Weka performs better than my method, as shown in Table 14

	Yiran	Weka
SSE	9427.378	158.78
SSB	8150.622	9123.09

Table 14: Comparison with Weka on Wine Dataset