Brad Pershon  
Data Mining  
HW #5 K-Means

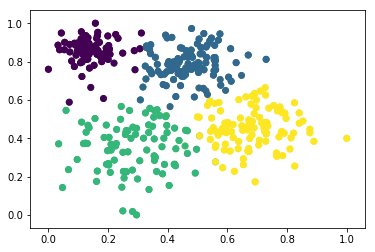
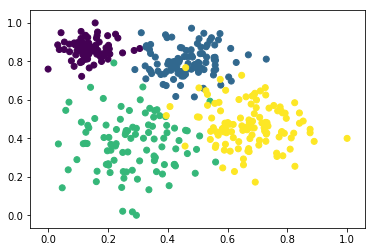
**Program Design**

The k-Means program we implemented accepts as a parameter the number of clusters k from the user. The program then performs the k-Means algorithm on both the TwoDimHard and Wine data sets. Our implementation starts with an initial estimate for the centroids by selecting k randomly generated points. The algorithm iterates between two steps: the data assignment step and the centroid update step.

In the data assignment step each record is assigned to its nearest centroid based on Euclidean distance. Once the entire dataset has been assigned to a cluster we continue on to the centroid update step. In this step the centroids are recomputed by taking the mean of all data points assigned to that centroids cluster. These two steps repeat until the centroids do not change. Our implementation is based on the simple k-Means algorithm, given this was our first attempt at implementing a clustering algorithm we thought it was best to start with the basic version. Both datasets are normalized before the k-Means algorithm is applied.

**Dataset 1: TwoDimHard**

The TwoDimHard dataset is a collection of 2D points with a corresponding cluster label ranging from 1 to 4. The figures below show the scatterplots for points with their cluster membership depicted by color. For this section we ran our algorithm with k set to four.

  
*Figure 1: The left image shows the scatterplot of 2D points given their true class labels. The right image shows the scatterplot of 2D points given their assigned class labels with k set to four.*

The left scatterplot displaying the ground truth cluster labels has points which are closer to the mean of other cluster centroids. For example the yellow dots mixed in with the blue and green dots. Our k-Means algorithm classifies points based on closest centroid, so there is no intermingling of colors. The SSE for each cluster and the SSB are shown below in table 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster** | **Real SSE** | **Cluster** | **Assigned SSE** |
| 1 | 0.435498632431 | 1 | 0.69174237291 |
| 2 | 1.25877334472 | 2 | 1.49663929573 |
| 3 | 3.35053610917 | 3 | 2.54908599895 |
| 4 | 2.65147771493 | 4 | 2.04197718063 |
| **Real Total** | 7.69628580125 | **Assigned Total** | 6.77944484822 |
| **Real SSB** | 32.97174104387 | **Assigned SSB** | 33.8885819969 |

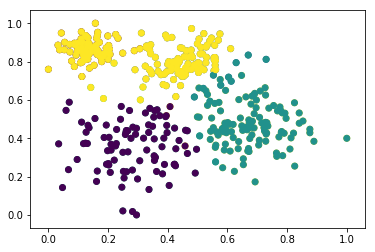
*Table 1: Real and Assigned SSE and SSB calculations for the TwoDimHard dataset with k set to four.*

The calculated SSE and SSB numbers appear to coincide with our scatterplots above. The Real SSE is higher because you have more outliers between clusters, and the assigned SSE is lower because our algorithm assigns a point’s cluster based on its closest centroid. So we have tighter cluster cohesion than the ground truth. The cluster outliers in the real scatterplot account for a lower cluster separation score given by the Real SSB. The table shown below outlines the number of points assigned to each class. These numbers appear correct as the outliers of clusters 3 and 4 are reassigned to clusters 1 and 2 in our k-Means implementation.

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Real Label** | **Assigned Label** |
| 1 | 89 | 95 |
| 2 | 100 | 108 |
| 3 | 97 | 90 |
| 4 | 114 | 107 |

*Table 2: Cross tabulation matrix comparing the actual and assigned clusters with k set to four.*

Below are the results of running our algorithm on the TwoDimHard dataset with k set to three.

  
*Figure 2: The image shows the scatterplot of 2D points given their assigned class labels with k set to three.*

The scatterplot shows the 2D points with their cluster membership denoted by color. With the loss of one cluster class we see the remaining 3 classes absorb the free points. With the number of clusters reduced from four to three we expect to see the cluster cohesion decrease. This is exactly what happens, as shown in table 3 the assigned SSE roughly doubled from 6.78 to 12.01.

|  |  |
| --- | --- |
| **Cluster** | **Assigned SSE** |
| 1 | 2.70729501084 |
| 2 | 3.11814079404 |
| 3 | 6.18052318884 |
| **Assigned Total** | 12.0059589937 |
| **Assigned SSB** | 28.6620678514 |

*Table 3: Real and Assigned SSE and SSB calculations for the TwoDimHard dataset with k set to three.*

Table 4 shown below provides a breakdown of the number of points belonging to each cluster. The top cluster, cluster 3, appears to have absorbed the majority of the free cluster points. The bottom right cluster, cluster 1 in this instance, appears to remain relatively the same.

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Real Label** | **Assigned Label** |
| 1 | 89 | 92 |
| 2 | 100 | 122 |
| 3 | 97 | 186 |
| 4 | 114 | - |

*Table 4: Cross tabulation matrix comparing the actual and assigned clusters with k set to three.*

Reducing the number of clusters definitely changes the results. With a reduction in the number of clusters we saw a reduction in cluster cohesion as the remaining clusters had to absorb the free points. From table 4 we can see the top cluster gained the majority of the free points once k was reduced. Given the scatterplot in figure 2 we can see the large spread of points within the top cluster. Based on the scatterplots in figure 1 and the smaller SSE we would say the clustering with k set to four is better.

**Dataset 2: Wine**

The Wine dataset has 14 dimensions, 11 are measurable attributes, an ID attribute, a dependent quality score, and a class label. The ID, quality score, and class label were removed and the measureable data was normalized before performing the k-Means algorithm. Below are the results with k set to four:

|  |  |  |
| --- | --- | --- |
| **Quality** | **Real** | **Assigned** |
| 3 | 10 | - |
| 4 | 53 | 29 |
| 5 | 681 | 708 |
| 6 | 638 | 477 |
| 7 | 199 | 385 |
| 8 | 18 | - |

*Table 5: Cross tabulation matrix comparing the actual and assigned clusters with k set to four.*

|  |  |
| --- | --- |
| **Cluster** | **Assigned SSE** |
| 1 | 0.43549863243 |
| 2 | 1.25877334472 |
| 3 | 3.35053610917 |
| 4 | 2.65147771493 |
| **Assigned Total** | 7.69628580125 |
| **Assigned SSB** | 33.8889262522 |

*Table 6: Assigned SSE and SSB calculations for the wine dataset with k set to four.*

The model appears to capture the real quality distribution in table 5, we see a high number of entries for quality score 5, and a low number of entries for quality score 4. It appears to have some trouble differentiating between clusters 6 and 7. The cluster cohesion appears to be pretty good considering the number of samples in the wine dataset.

|  |  |  |
| --- | --- | --- |
| **Quality** | **Real** | **Assigned** |
| 3 | 10 | 29 |
| 4 | 53 | 278 |
| 5 | 681 | 591 |
| 6 | 638 | 337 |
| 7 | 199 | 329 |
| 8 | 18 | 35 |

*Table 7: Cross tabulation matrix comparing the actual and assigned clusters with k set to six.*

|  |  |
| --- | --- |
| **Cluster** | **Assigned SSE** |
| 1 | 0.00000000001 |
| 2 | 0.45549863242 |
| 3 | 1.45877334472 |
| 4 | 3.33053610917 |
| 5 | 2.25147771491 |
| 6 | 0.00000000002 |
| **Assigned Total** | 7.49628580125 |
| **Assigned SSB** | 35.8889262522 |

*Table 8: Assigned SSE and SSB calculations for the wine dataset with k set to six.*

The model is doing a fairly good job at the tails of the distribution given the results in table 7. The model still appears to have trouble distinguishing between classes 6 and 7. The SSE decreased slightly, but the clusters with only a few entries had very low SSE’s. This is likely due to the small number of entries, and the model having trouble choosing between the tail classes.

|  |  |  |
| --- | --- | --- |
| **Quality** | **Real** | **Assigned** |
| 2 | - | 37 |
| 3 | 10 | 111 |
| 4 | 53 | 279 |
| 5 | 681 | 374 |
| 6 | 638 | 364 |
| 7 | 199 | 12 |
| 8 | 18 | 12 |
| 1 | - | 6 |

*Table 9: Cross tabulation matrix comparing the actual and assigned clusters with k set to eight.*

|  |  |
| --- | --- |
| **Cluster** | **Assigned SSE** |
| 1 | 0.435498632431 |
| 2 | 1.25877334472 |
| 3 | 3.35053610917 |
| 4 | 2.65147771493 |
| 5 | 0 |
| 6 | 0 |
| **Assigned Total** | 7.69628580125 |
| **Assigned SSB** | 36.3907628128 |

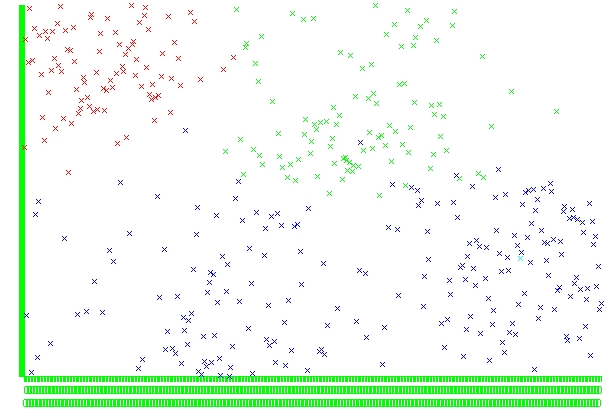
*Table 10: Assigned SSE and SSB calculations for the wine dataset with k set to eight.*

The model doesn’t appear to be getting better at approximating the real quality score distribution. The SSE decreased slightly but nothing significiant. Given the model distribution from table 9 we would select the model with k set to 6 as the best model.

**Weka k-Means clustering**

Dataset 1: TwoDimHard

For the off the shelf k-Means implementation we chose the Weka Simple k-Means algorithm. To run the clustering algorithm the data had to be normalized. The number of clusters was adjusted from 2 to 4. The remaining parameters were left at their default values. Below are the results with k set to four:

  
*Figure 3: Weka Simple k-Means of TwoDimHard dataset with k set to four.*

The first thing we noticed is it appears as if there are only 3 classes in figure 3, this is because the Weka algorithm only found 1 record belonging to cluster 4. The Weka model does allow for some noisy data points, for example you can see blue data points within the green cluster area. The algorithm appeared to have the majority of its problems distinguishing between classes 3 and 4 as shown below in table 5.

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Real Label** | **Assigned Label** |
| 1 | 89 | 89 |
| 2 | 100 | 100 |
| 3 | 97 | 210 |
| 4 | 114 | 1 |

*Table 11: Cross tabulation matrix comparing the actual and assigned clusters with k set to four.*

We ran the Weka algorithm once more with k set to three, but we got similar results as shown above. It simply moved the 1 entry from cluster 4 into cluster 3. The model computed a SSE of 889, which is extremely large to the point it was almost omitted from the report. Disregarding the Weka models SSE we still consider our implementation better than the off the shelf model as it gives a distribution closer to the ground truth distribution. If we didn’t have the ground truth distribution we would still consider ours the better model as the Weka model only found 1 entry for the 4th cluster.

Dataset 2: Wine

Again we used the Weka Simple k-Means algorithm with the parameters outlined in the TwoDimHard section. Below are the results with k set to four:

|  |  |  |
| --- | --- | --- |
| **Quality** | **Real** | **Assigned** |
| 3 | 10 | - |
| 4 | 53 | 327 |
| 5 | 681 | 494 |
| 6 | 638 | 417 |
| 7 | 199 | 361 |
| 8 | 18 | - |

*Table 12: Cross tabulation matrix comparing the actual and assigned clusters with k set to four.*

The Weka model appears to favor a more uniform distribution when compared to the real quality scores. We attempted to run the model again with the number of clusters increased to six and eight, below are the results:

|  |  |  |
| --- | --- | --- |
| **Quality** | **Real** | **Assigned** |
| 3 | 10 | 207 |
| 4 | 53 | 245 |
| 5 | 681 | 326 |
| 6 | 638 | 292 |
| 7 | 199 | 291 |
| 8 | 18 | 238 |

*Table 13: Cross tabulation matrix comparing the actual and assigned clusters with k set to six.*

|  |  |  |
| --- | --- | --- |
| **Quality** | **Real** | **Assigned** |
| 2 | - | 108 |
| 3 | 10 | 147 |
| 4 | 53 | 211 |
| 5 | 681 | 326 |
| 6 | 638 | 291 |
| 7 | 199 | 238 |
| 8 | 18 | 147 |
| 1 | - | 131 |

*Table 14: Cross tabulation matrix comparing the actual and assigned clusters with k set to eight.*

We can see the Weka model continues to give a more uniform distribution when compared to the real quality score distribution. The Weka model computed the following SSE’s: 323.43, 281.14, and 260.62 respectively. This would indicate the model with k set to eight had the highest cluster cohesion. Based on the Weka SSE’s and distributions we would consider our model better as it had better cluster cohesion and more accurately represented the quality score distribution.

Overall I thought the project was successful. The scatterplots and cluster validity measurements appeared to coincide with our expectations. In the future I would consider implementing a better method to initialize the centroids, as the random point method can take a while to converge at times. We did very little post-processing in our implementation, another future consideration would be to split clusters with a high SSE.