K-Means on Implementation and Analysis

Pavani Komati | CSE 5243 | 28th Mar’18

Contents

[1.Introduction 1](#_Toc510614334)

[PART 1: Implementation k-Means clustering algorithm 1](#_Toc510614335)

[PART 2: Off-the-shelf Clustering Algorithm analysis 4](#_Toc510614338)

[Two Dim hard dataset Analysis 5](#_Toc510614339)

[Wine dataset data set Analysis 7](#_Toc510614340)

## 1.Introduction

The report summarizes about the implementation and analysis of the k-means clustering algorithm run on a two-dimensional dataset (TwoDimHard). The first part explains about the design choices and implementation of the K-means algorithm from scratch, which is tested on the current dataset. The second part concludes the previous analysis by comparing with the analysis of the dataset using off-the-shelf k-means algorithm from sklearn.

# PART 1: Implementation k-Means clustering algorithm

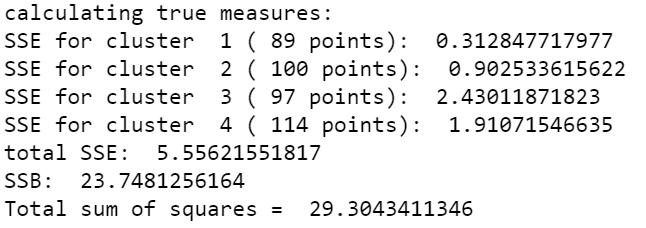
## Design Decision and Implementation

The K-means algorithm is implemented using the standard Euclidean distances measure. To compute the distances between the points, the values need to be numeric data type on the same scale. Hence normalization of values is required. (However, in the current dataset as the values as the X.1 and X.2 are on the same scale around (0-1) normalization wouldn’t be necessary in the current scenario).

The current K-means algorithm takes k value, dataset, columns, max\_iterations as parameters. Initially, k centroids are chosen randomly in the range of 0-1 and k clusters are computed by assigning each point with the cluster corresponding to the closest centroid. New centroids for the current clusters are found and the process is repeated until new centroids match with the current centroids or number of iterations exceeds the max\_iterations. This process confirms that the final centroids are stable and even after further iteration, the clusters remain constant.

After getting the stabilized clusters, the performance of the model is measured by calculating the SSE (sum of squared errors) within clusters, total SSE and SSB (sum of squared errors between clusters). SSE is obtained by calculating the sum of Euclidean distances between the centroid and points in each cluster. SSB is obtained by where mi is the number of points in each cluster, ci is the centroid of cluster i and c is the main centroid.

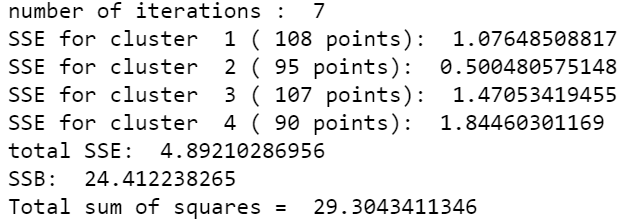
1. **True Cluster SSE, overall SSE, SSB**

True cluster SSE, overall SSE

and SSB are calculated using the true cluster value from ‘cluster’ column in the dataset.

**K-Means with K =4**

1. SSE for each cluster, the overall SSE and SSE between-cluster sum of squares (SSB) is calculated for the dataset with k=4. The current algorithm took 6-10 iterations on an average to compute the stabilized centroids. (Depending on how close the initial random centroids are, the number of iterations vary every time we run the algorithm)



1. Scatter plots for the true clusters and the clusters formed with the current algorithm (k=4) are plotted and shown as below. The legend on each plot shows the clusters with their corresponding numbers and colors. The marker ‘x’ represents the centroids of the clusters.

|  |  |
| --- | --- |
| K-Means clusters (k=4) | True Clusters |

1. Cross tabulation matrices are created to compare the actual and predicted clusters

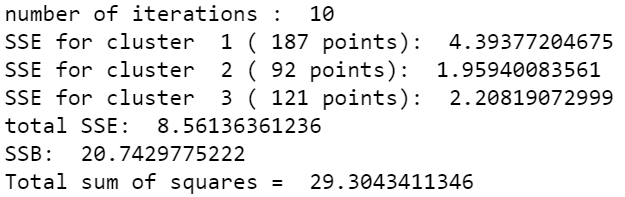
|  |  |
| --- | --- |
| **K=4** | **Predicted Clusters** |
| **Actual Clusters** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Cluster 4** | **Total** | | **Cluster 1** | 0 | 89 | 0 | 0 | 89 | | **Cluster 2** | 98 | 2 | 0 | 0 | 100 | | **Cluster 3** | 2 | 4 | 3 | 88 | 97 | | **Cluster 4** | 8 | 0 | 104 | 2 | 114 | | **Total** | 108 | 95 | 107 | 90 | 400 | |

From the table we can observe that, the original cluster 1 is predicted as 2 i.e 1🡪2, 2 🡪1, 3 🡪4 and 4🡪3. Total of 21(5.25%) records out of 400 have been misclassified.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SSE | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Total | SSB | Total SS |
| **Actual** | 0.313 | 0.903 | 2.430 | 1.910 | 5.556 | 23.748 | 29.304 |
| **Predicted** | 0.500 | 1.076 | 1.845 | 1.471 | 4.892 | 24.412 | 29.304 |

With the comparison of SSE and SSB of actual and predicted clusters, we can observe that the predicted clusters have higher value of sum of squared errors between the clusters. However, the error within in the clusters is lesser in the predicted model than the actual model. Total SSE +SSB remains constant as it is not dependent on clusters.

1. **K-Means with K =3**
2. SSE for each cluster, the overall SSE and SSE between-cluster sum of squares (SSB) is calculated for the dataset with k=3. The current algorithm took 9-10 iterations on an average to compute the stabilized centroids.



1. Scatter plots for the true clusters and the clusters formed with the current algorithm (k=3) are plotted and shown as below. The legend on each plot shows the clusters with their corresponding numbers and colors. The marker ‘x’ represents the centroids of the clusters.

|  |  |
| --- | --- |
| K-Means clusters (k=3) | True Clusters |

1. Cross tabulation matrices are created to compare the actual and predicted clusters

|  |  |
| --- | --- |
| **K=3** | **Predicted Clusters** |
| **Actual Clusters** | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Total** | | **Cluster 1** | 89 | 0 | 0 | 89 | | **Cluster 2** | 92 | 0 | 8 | 100 | | **Cluster 3** | 5 | 89 | 3 | 97 | | **Cluster 4** | 1 | 3 | 110 | 114 | | **Total** | 187 | 92 | 121 | 400 | |

From the table we can observe that, the original cluster 1 and 2 together are predicted as cluster 1, cluster 3 is predicted as cluster 2 and cluster 4 is predicted as cluster 3.

comparision between c and D results:

From the comparison of the plots for k=3, k=4 below, it is observed that clusters 1 and 2 in k=4 is merged as one in k=3 plot. These two clusters are clearly separable.

|  |  |
| --- | --- |
| K-Means clusters (k=3) | K-Means clusters (k=4) |

|  |  |  |  |
| --- | --- | --- | --- |
| SSE | Total SSE | SSB | Total SS |
| K = 3 | 8.561 | 20.7429 | 29.304 |
| K= 4 | 4.8921 | 24.4122 | 29.304 |

From the above table, SSE within the clusters is high for k=3 when compared to k=4. As we can observe from the k=3 plot, the points in cluster 1 are spread far away from its centroid. This is the reason for increase in SSE. Also, SSB value is reduced in k=3 due to less number of clusters and modified centroid. With the change in k value (number of clusters), the sum of squared error with in the clusters and between the clusters change accordingly. Hence with the high value of SSE and the plot analysis for clustering with k=3, we can clearly say that k=4 is the better choice.

# PART 2: Off-the-shelf Clustering Algorithm analysis

The off-the-shelf analysis for the datasets is performed using k-means algorithm from cluster in sklearn package. The k-means algorithm from sklearn takes various parameters. Few of the prominent parameters are:  
‘n\_clusters’ – number of clusters to be considered  
‘init’ – method to choose the centroids. (k-means++🡪optimised, random)   
‘n\_init’: Number of time the k-means algorithm will be run with different centroid seeds.   
‘max\_iter’: Maximum number of iterations of the k-means algorithm for a single run.

Two Dim hard dataset Analysis

1. **Model Parameters**

The current dataset is tested with varying the ‘n\_clusters’ i.e k= 3, 4. As the dataset size is very small (400 points) the number of iterations required to get the stabilized centroid is very less. Hence changing the ‘max\_iter’ parameter wouldn’t make any difference. Also, by changing the parameter values for ‘init’ and ‘n\_init’ did not show any observable difference to the current dataset.

1. **Cluster Analysis:**

|  |  |
| --- | --- |
| Off-the-shelf analysis with k=3 | Off-the-shelf analysis with k=4 |
|  |  |
| Scatter plots for the true clusters and the clusters formed with the off-the-shelf algorithm are plotted and shown as below. The legend on each plot shows the clusters with their corresponding numbers and colors. The marker ‘x’ represents the centroids of the clusters. | |
|  |  |
| |  |  | | --- | --- | | **K=3** | **Predicted Clusters** | | **Actual Clusters** | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Total** | | **Cluster 1** | 89 | 0 | 0 | 89 | | **Cluster 2** | 92 | 0 | 8 | 100 | | **Cluster 3** | 5 | 89 | 3 | 97 | | **Cluster 4** | 1 | 3 | 110 | 114 | | **Total** | 187 | 92 | 121 | 400 | | | |  |  | | --- | --- | | **K=4** | **Predicted Clusters** | | **Actual Clusters** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Cluster 4** | **Total** | | **Cluster 1** | 0 | 89 | 0 | 0 | 89 | | **Cluster 2** | 0 | 2 | 98 | 0 | 100 | | **Cluster 3** | 88 | 4 | 2 | 3 | 97 | | **Cluster 4** | 2 | 0 | 8 | 104 | 114 | | **Total** | 90 | 95 | 108 | 107 | 400 | | |
| From the table we can observe that, the original cluster 1 and 2 together are predicted as cluster 1, cluster 3 is predicted as cluster 2 and cluster 4 is predicted as cluster 3. | From the table we can observe that, the original cluster 1 is predicted as 2 i.e 1🡪2, 2 🡪3, 3 🡪1 and 4🡪4. Total of 21(5.25%) records out of 400 have been misclassified. |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | SSE | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Total | SSB | Total SS | | Actual | 0.313 | 0.903 | 2.430 | 1.910 | 5.556 | 23.748 | 29.304 | | Predicted | 4.3937 | 🡨 | 1.959 | 2.2081 | 8.561 | 20.743 | 29.304 | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | SSE | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Total | SSB | Total SS | | Actual | 0.313 | 0.903 | 2.430 | 1.910 | 5.556 | 23.748 | 29.304 | | Predicted | 0.500 | 1.076 | 1.845 | 1.471 | 4.892 | 24.412 | 29.304 | |
| Cluster 1 has abnormally high SSE indicating incorrect clustering. Hence  K=3 has higher total SSE indicating the poor performance of model. | K=4 has lower SSE compared to k=3 indicating better performance of the model |

With the comparison of SSE and SSB of actual and predicted clusters using off-the-shelf algorithm, we can conclude that k=4 model performs better.

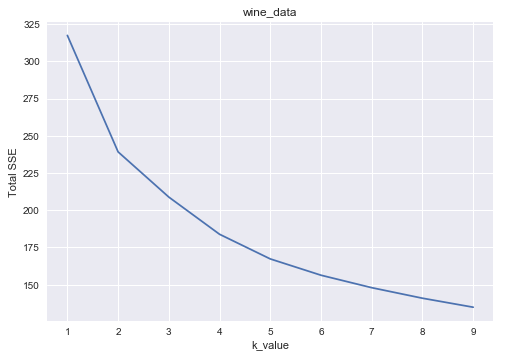
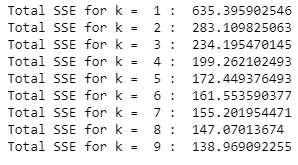
1. **Comparison of current k-means algorithm and off-the-shelf k-means algorithm.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Current k-means | | | | Off-the-shelf K-Means | | | | |
| COMPARING Models with K=4 | | | | | | | | |
| Comparing the performance measures | | | | | | | | |
|  | | | |  | | | | |
| We can observe that the only difference in both the measures is the cluster mapping  Cluster mappings : 1🡪3 , 2🡪2, 3🡪4, 4🡪 1 .  The total SSE and SSB values remain constant | | | | | | | | |
| Scatter plots for the true clusters and the clusters formed with both the algorithms are shown as below. We can notice that both the models have performed equally well. | | | | | | | | |
|  | | | |  | | | | |
| Comparing the Total SSE values of clusters (mapped to the original clusters) | | | | | | | | |
| SSE (K=4) | Cluster 1 | Cluster 2 | Cluster 3 | | Cluster 4 | Total | SSB | Total SS |
| **Current Algo** | 1.0764 | 0.500 | 1.470 | | 1.844 | 5.556 | 23.748 | 29.304 |
| **Off-the-shelf** | 1.0764 | 0.500 | 1.470 | | 1.844 | 5.556 | 23.748 | 29.304 |
| We can observe that all the SSE values obtained both the algorithms are exactly equal. | | | | | | | | |
| COMPARING Models with K=3 | | | | | | | | |
| Comparing the performance measures | | | | | | | | |
|  | | | |  | | | | |
|  | | | |  | | | | |
| Similar to the model with k=4, the performace measures and plots in both the models for k=3 are exactly equal. | | | | | | | | |

The current implementation works exactly like the off-the-shelf analysis with similar performance measures. Hence, it can be concluded that the current implementation from scratch performs exactly as expected without any issues.

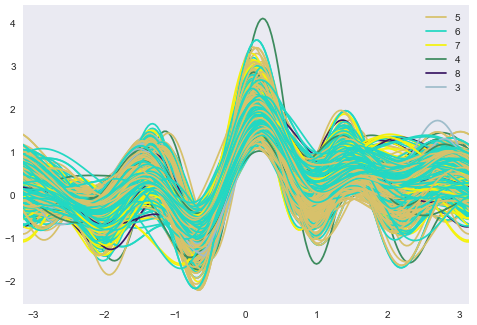
Wine dataset data set Analysis

# The wine-data set consists of 1599 records with 11 features. These features are spread out with varied distribution(range). The ‘quality’ column which gives the specific class label of the cluster is dropped and the features are normalized to perform the k-means clustering.

The off-the-shelf k-means algorithm from sklearn is run on the dataset with k values ranging from 1-10 and the total SSE of the clusters are calculated. With the reduction in k value, SSE always decreases. However, we need to find a point where the reduction in SSE is not significant. From the point plot with k vs total\_SSE reduces significantly from k = 1-5 and from k = 5-10 the reduction is not very high. Hence, we can confirm that the k value lies around 5-6.

To view the clustering for different k-values, Andrew plots (with legend showing clusters for the data), the cross tab between the true cluster and off-the-shelf result is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| K =3 | K =5 | K =6 | K =9 |
|  |  |  |  |

**True cluster**

With the cross tab and plots we can see that, k=5 and k=6 performs slightly better clustering than the other k values. However, even with k =5/6 the clusters are not formed perfectly. There is huge amount of misclassification in the clustering. Hence, the given wine dataset as such doesn’t perform well on the K-Means clustering.