University of Windsor

Collaborative Clustering C-16

06-88-590-03

Data Mining

Summer 2018

Prepared for: Dr. Roozbeh Razavi Far

Prepared by:

Md Ibtehajul Islam…….(104764046)
Abdul Shakib Billah…….(104874059)
Faysal Ahammed Shishir……(104942369)

July 31st, 2018

Acknowledgement

Our sincere acknowledgement to **Prof. Dr. Roozbeh Razavi Far**, the University of Windsor for his timely rendered guidance and help for the project. Without his assistance, this project works we did would not have been possible. It was a great experience doing the project under his guidance and support and has significantly added to the foundation of our career. Being an inspiration for us, we thank him once again for providing all that was needed to complete the project.

Additionally, we would also like to thank our **Ehsan Hallaji** ex. Student of our professor for his guidance and support we had for learning Python. Without his timely rendered help, coding in Python would have been extremely complex as we did not have any prior knowledge of Python libraries.

Executive Summary

Data Clustering is an essential task in the process of information extraction from a huge chunk of datasets that is targeted towards finding the underlying structures in an ensemble of objects by forming clusters that share similarities. There is a lot of different way clustering can be done and every new method is being published on every other day. So instead of writing a new algorithm, we tried to collaborate the existing ones in hopes of better results. The collaborative clustering makes different clustering methods collaborate which is usually unsupervised learning and takes the upper hand of different clustering results. The idea is to provide several local clustering results to reach an agreement on the partitioning of a common clustering of the and an iterative approach to improve the clustering process using an ensemble of clustering methods. As we all know different clustering methods can lead us to a different partitioning of the same dataset, finding a consensual clustering from these results is often a cumbersome task to reach the destination. The collaboration in-between the clustering aims to make the methods agree on the different clusters through a refinement of their results in an iterative form trying several algorithms with different parameter configurations. Through this iterative process, the results lean to become more similar.

Every new data sets introduced means that it is getting increasingly difficult for individual clustering algorithms to give good computational performances in a certain effective time frame. The main goal is to find a satisfying clustering often requires trying several algorithms with different parameter configurations. And clustering is a complex problem, once several results have been found, there is no one-way route to reach the objective way and to decide which one is the best.

Table of Contents

List of Figures

1. Introduction

From different types of machine learning techniques, clustering is an unsupervised learning. This method aims to partition a data set or object into ensembles named clusters. Clusters can be either reciprocally exclusive or overlapping on each other and it relies on the approach.

1.1. Clustering

Clustering is aggregation of a specific group of objects based on their characteristics and according to their similarities and use them according to their desired data analysis. From different types of machine learning techniques, clustering is an unsupervised learning. This method aims to partition a data set or object into ensembles named clusters. Clusters can be either reciprocally exclusive or overlapping on each other and it relies on the approach.

Information present in every huge data sets can be manipulated using this kind of analysis, this will eventually lead to unwrapping great results with many distinct types of data. Clustering analysis in general terms is broadly used in many applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others.

1.1.1. Collaborative clustering: Why & how?

The basic of collaborative clustering deals with revealing a structure that is common or similar structure within a wide range of data sets. The core principle of Collaborative Clustering is to collaborate and share some information within the local data to improve one single cluster with the information provided by the other collaborators to find optimal clusters. The aim is to work with different clustering methods collaborate each other so that an agreement on the partitioning of a common dataset is met, keeping in mind that different clustering methods can yield to different partitioning of the same dataset. The collaboration in between clusters helps the other methods to agree on the partitioning results through repetitive refinement.

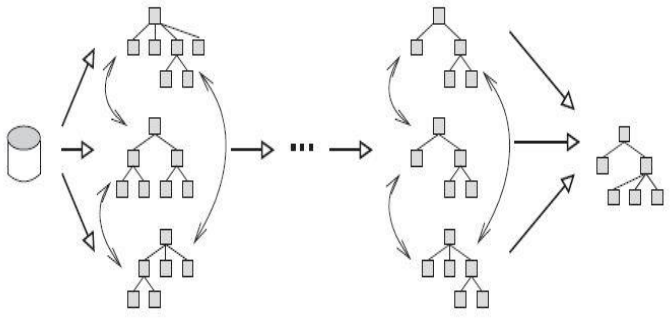

Figure 1: Collaborative clustering.

Figure 1 illustrates how different clustering algorithms are collaborating with each other and refining the outputs of each algorithm and generating a single pattern.

2. Implementation Procedure

To collaborate on several clustering a procedure needs to be followed where choosing the individual clustering algorithm plays a vital role. In this project, the k-means clustering algorithm has been chosen to solely cluster a dataset. K-means clustering result varies upon choosing the initial centroids coordinates and number of clusters to be formed. This idea has been used here to generate several clustering results from the same dataset by varying the number of predefined clusters.

In K-means clustering algorithm, several clustering centroids coordinate are chosen randomly to start the clustering procedure in the beginning. The number of final clusters depends on the chosen number of clustering centroids. Choosing the number of centroids can be pre-defined by the user or totally random. After this step, the Euclidean distance of each data is measured with the individual centroids. The data with the shortest distance with one centroid form one cluster. This procedure iterates for the whole data set. When the cluster is formed the algorithm chooses a new centroid by taking the mean of the data of one cluster. The whole process iterates until the centroid converges to the center of each cluster.

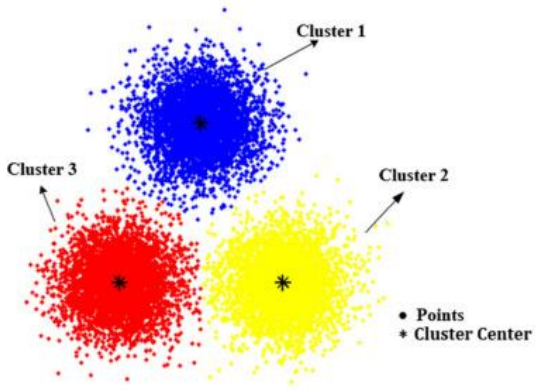

Figure 2: K-means clustering.

After generating multiple k-means clusters the target is to collaborate the clusters of different algorithms with each other. The collaboration gives us the conflicts between one cluster of different K-mean algorithms. The goal of this project is to iterate until the conflicts are resolved and make every cluster from every k-mean algorithm corresponding to each other.

3. Steps of collaborative Clustering Algorithm [1]

Collaborative clustering runs several K-means clustering algorithm and collaborates every cluster from one algorithm with all clusters of other k-mean algorithms and vice-versa. After collaboration, the algorithm refines the results by detecting and solving the conflicts. When a conflict is detected between two clusters of different algorithm, the conflict needs to be resolved is chosen. After that several operators are applied to create a local resolution according to the number of clusters involved in the conflict. In the global assessment, the local similarities between each couple are evaluated. The whole process iterates until all the conflicts are solved. Only one conflict

resolution is engaged at each iteration.

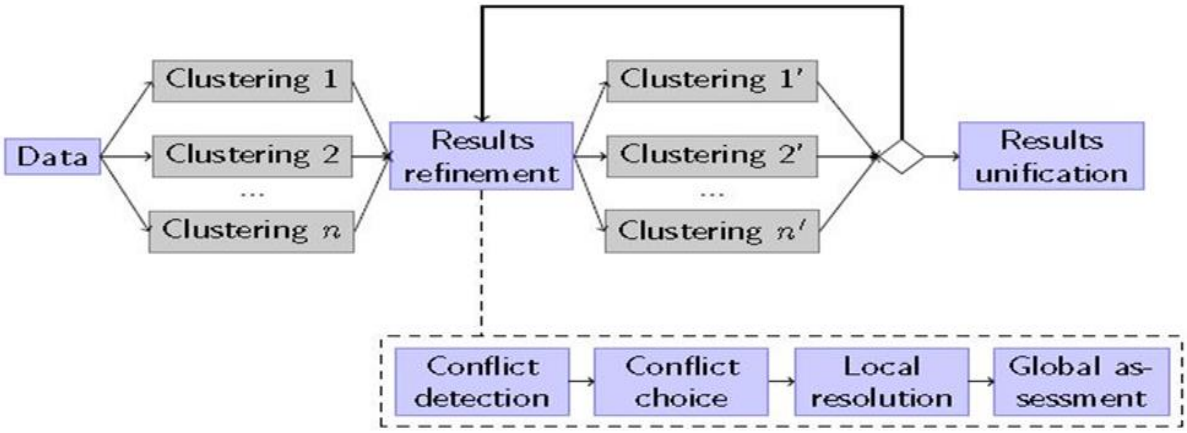

Figure 3: Collaborative clustering algorithm.

3.1. Taking Input

In this step, we read the data from the given data set on an excel data sheet. Initially, the python code reads the whole excel data sets and wants the user to give input of the desired sheet.

```
def read_data (filename):
  workbook = xlrd.open_workbook(filename)
  print('Reading Excel Data sheets')

  while True:
    names = workbook.sheet_names()
    print ("Please select a sheet from below name list : ")
    print (names)
    sheet_name = input('Please Enter the Sheet Name: ').strip()
    worksheet = workbook.sheet_names()
    if sheet_name in worksheet:
      n_worksheet = workbook.sheet_by_name(sheet_name)
      print ('The sheet is Available')

      total_rows = n_worksheet.nrows
      total_cols = n_worksheet.ncols
      table = list()
      record = list()
      for x in range (total_rows):
        for y in range (total_cols):
          record.append(n_worksheet.cell(x,y).value)
        table.append(record)
        record = []
        x += 1
```

```
            table = np.array(table)
                #print(table)
                if n_worksheet == workbook.sheet_by_name(sheet_name):
                    break
            else:
                print('The sheet is NOT Available. Please select from the below list
only.')

            return table
```

3.2. Running K-means

In this step, we need to cluster the given data using K-mean algorithm. At least 2 K-mean algorithms should be run with different values of K. Here, K is the value of number of clusters in one k-mean algorithm. K can be generated using random number generation method based on the length of given data sets. In the end of this step we will have the several clusters of the data from different multiple K-mean algorithms. The clusters are visualized using python package "matplotlib".

```
def run_KMean(number_of_class,dataX):
    cmap = plt.cm.get_cmap('hsv', number_of_class+1) # for color map on Plot
    kmeans = KMeans(n_clusters = number_of_class, init = 'k-means++', max_iter = 300,
n_init = 10, random_state = 0) # read sklearn library (kmean) for more details
    y_kmeans = kmeans.fit_predict(dataX) # Creating the model on dataset
    for i in range(0, number_of_class): # for plotting the dataset
        plt.scatter(dataX[y_kmeans == i, 0], dataX[y_kmeans == i, 1], s = 5, c = cmap(i+1),
label = 'cluster_'+str(i+1))
    plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1], s = 50, c =
'black', label = 'Centroide', marker = '*', alpha=0.5) # for plotting the calculated centroids
    # plt.legend()
    plt.title('Cluster_'+str(number_of_class))
    plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
    plt.savefig('figure/fig_cluster_'+str(number_of_class)+'.png',     bbox_inches="tight")
#saving the figures into a folder named "figure" -need to create this folder manually
    plt.clf()
    plt.show()
    return y_kmeans
```

Assuming 3 K-mean clustering algorithms: algorithm A, algorithm B and algorithm C. Algorithm A has 3 clusters in it, algorithm B has 4 clusters and algorithm C has 7 clusters inside it.

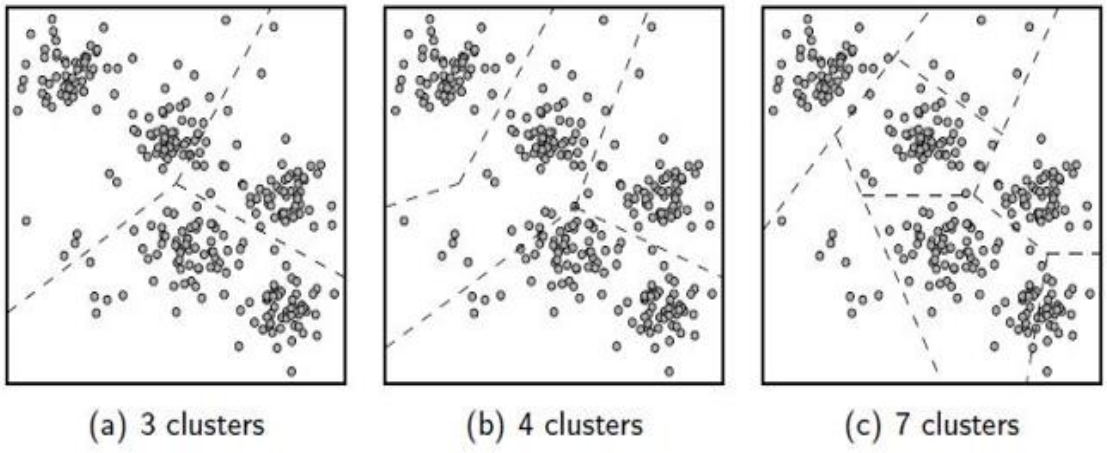

Figure 4: Example of 3 different clustering results.

3.3. Result Refinement

After getting the clusters from different algorithms, the next step is collaboration and the refinement of the results.

3.3.1. Conflict Detection

In this step we measure the similarity between Different clusters to identify the conflict between them. To identify the similarities a confusion matrix is used.

Let, C^1, C^2, C^3 represent the 3 clustering Algorithm A, B & C. And Inside C^1 we have 3 clusters C_1, C_2 & C_3. Similarly, Inside C^2 & C^3 we have C_1, C_2, C_3 & C_4 and C_1, C_2, C_3, C_4, C_5, C_6, C_7 clusters respectively. To compute the similarity between two results, the intersections between each couple of clusters ($C_k{}^i$, $C_l{}^j$), from two results $C^{(i)}$and $C^{(j)}$ are computed in the **confusion matrix**:

$$\Omega^{(i,j)} = \begin{pmatrix} \alpha_{1,1}^{(i,j)} & \cdots & \alpha_{1,K^{(j)}}^{(i,j)} \\ & \vdots & \\ \alpha_{K^{(i)},1}^{(i,j)} & \cdots & \alpha_{K^{(i)},K^{(j)}}^{(i,j)} \end{pmatrix} \text{ where } \alpha_{k,l}^{(i,j)} = \frac{\left| C_k^{(i)} \bigcap C_l^{(j)} \right|}{\left| C_k^{(i)} \right|}$$

Similarity must be identified using above confusion matrix: initially considering cluster C^1 from clustering algorithm C_1 with every clusters of algorithms C_2 and vice-versa. In short, from these two matrices ($\Omega^{(I,j)}$ and $\Omega^{(j,I)}$), a similarity measure is computed to compare two clusters of two different results.

$$S\left(C_k^{(i)}, C_l^{(j)}\right) = \rho_k^{(i,j)} \alpha_{l,k}^{(j,i)}$$

where

$$\rho_k^{(i,j)} = \sum_{r=1}^{n_j} (\alpha_{k,r}^{(i,j)})^2$$

Then, this similarity measure is used to compute the corresponding cluster of each cluster, which is the most similar cluster in another result:

$$\psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right) = \arg \max_{C_l^{(j)} \in \mathcal{C}^{(j)}} S\left(C_k^{(i)}, C_l^{(j)}\right)$$

Here, by aggregating the maximum similarity we are finding out how many maximum numbers of clusters from algorithm 2/1 are subset or correspond to the whole clustering algorithm 2/1. This correspondence between the clusters of the different results is used to identify the conflicts between the different results.

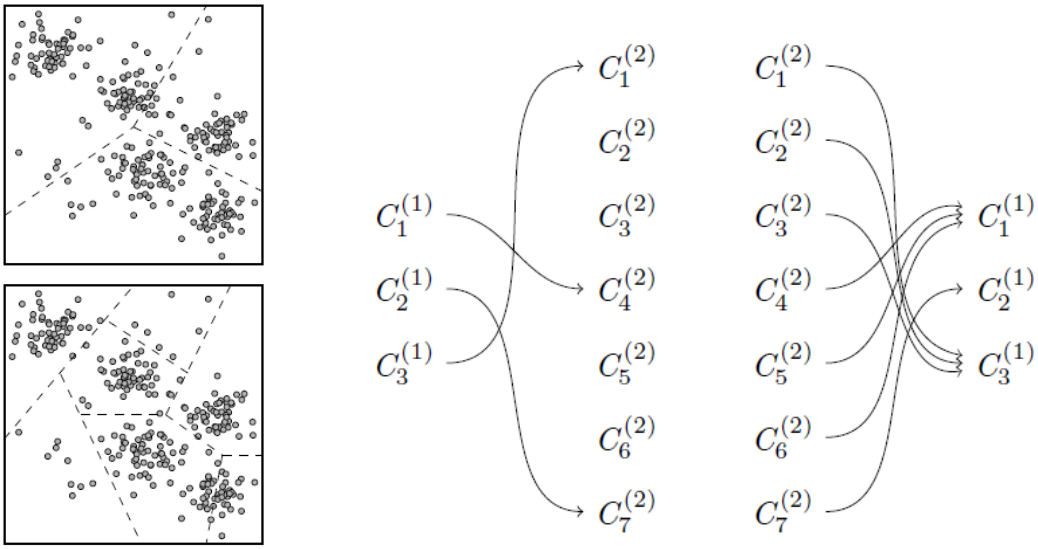

Figure 5: Correspondence between clusters.

For detecting the conflicts we again need to measure the similarity between $C_k{}^I$ and $\breve{\sigma}(C_k{}^i, C^j)$, where "*i*" is not equals to *"j"*.

If, $S(C_k{}^I, \breve{\sigma}(C_k{}^i, C^j)) < 1$, the cluster $C_k{}^I$ cannot be found in the result C^j. So, the expression can be written as-

$$\texttt{conflicts}(\mathbb{C}) = \left\{(C_k^{(i)}, \mathcal{C}^{(j)}) : i \neq j, S\left(C_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right) < 1\right\}$$

Each conflict $K_k^{(i,j)}$ is identified by one cluster $C_k^{(i)}$ and one result $C^{(j)}$. Its importance, $CI(K_k^{(i,j)})$, is computed according to the inter cluster similarity.

$$CI\left(\mathcal{K}_k^{(i,j)}\right) = 1 - S\left(C_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right)$$

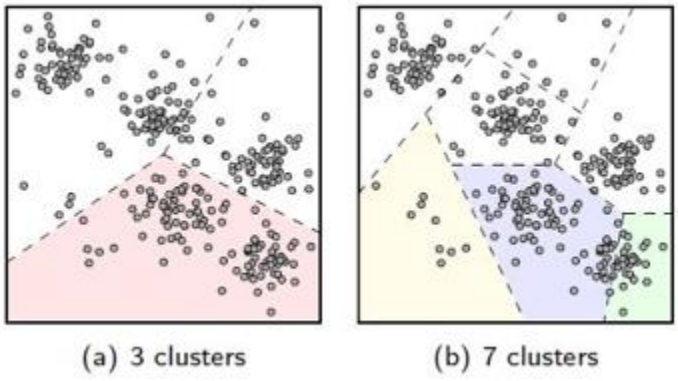

Figure 6: Example of conflicts between two results.

Here, $\texttt{conflicts}(\mathbb{C}) = \ CI\left(\mathcal{K}_k^{(i,j)}\right)$

3.3.2. Conflict Choice

After detecting the conflict, the next step is to choose a cluster from the couple which needs to be solved to improve similarity between the results. The strategy of choosing most important conflict can be random, conflict weighting, rank selection, tournament selection, etc.

Selection of most important conflict:

$$\mathcal{K} := \arg \max_{\mathcal{K}_{(i)} \in \check{\mathcal{K}}} CI\left(\mathcal{K}_{(i)}\right)$$

3.3.3. Local Resolution

The modified cluster, for improving the similarity by resolving the conflict is called the local resolution. Some operators are required to apply to get the local resolution. Assuming a local resolution of a conflict $K_k^{(I,j)}$, where the conflict is in between two clusters $C^{(i)}$ and $C^{(j)}$ where the similarity needs to be improved. For the improvement, operators such as merging, splitting and reclustering are used.

To evaluate the similarity between two results a local similarity criterion Υ is defined.

$$\gamma^{(i,j)} = \frac{1}{2}\left(p_s.\left(\frac{1}{n_i}\sum_{k=1}^{n_i}\omega_k^{(i,j)} + \frac{1}{n_j}\sum_{k=1}^{n_j}\omega_k^{(j,i)}\right) + p_q.\left(\delta^{(i)} + \delta^{(j)}\right)\right)$$

Where,

$$\omega_k^{(i,j)} = S\left(C_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right)$$

And, here P_q and P_s are given by the user ($P_q + P_s = 1$).

Let, $C^{(i')}$ (resp. $C^{(j')}$) be the result $C^{(i)}$ (resp. $C^{(j)}$) after having applied the operators. The local similarity criterion is computed on each of the 4 couples of results: *(($C^{(i)}$, $C^{(j)}$), ($C^{(i')}$, $C^{(j')}$), ($C^{(i')}$, $C^{(j)}$), ($C^{(i)}$, $C^{(j')}$))*. The best couple is accepted as the local solution of the conflict.

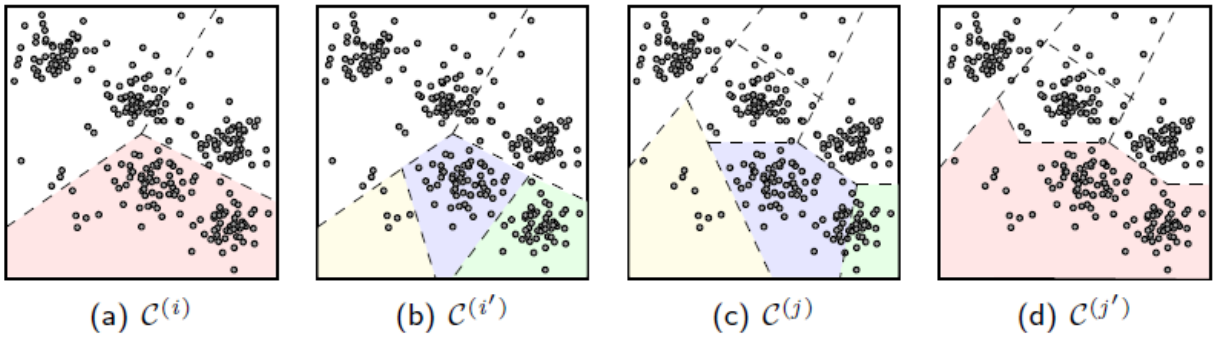

Figure 7: The four results obtained after an operator application.

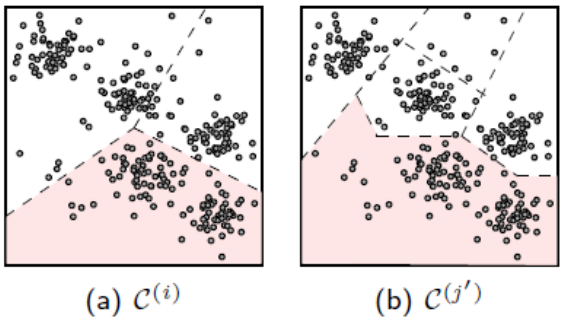

Figure 8: The two kept results.

3.3.4. Global Assesment

In the global assessment, the local similarities between each couple are evaluated using a global agreement coefficient Γ. The whole process iterates until all the conflicts are solved. Only one conflict resolution is engaged at each iteration.

$$\Gamma = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} \Gamma^{\,i}$$

Where,

$$\Gamma^{\,i} = \frac{\mathbf{1}}{\boldsymbol{m-1}} \sum_{\substack{\boldsymbol{j=1} \\ \boldsymbol{j \neq 1}}}^{\boldsymbol{m}} \boldsymbol{\gamma^{(i,j)}}$$

More the Γ, more similarity between the cluster couple.

4. Experimental Results

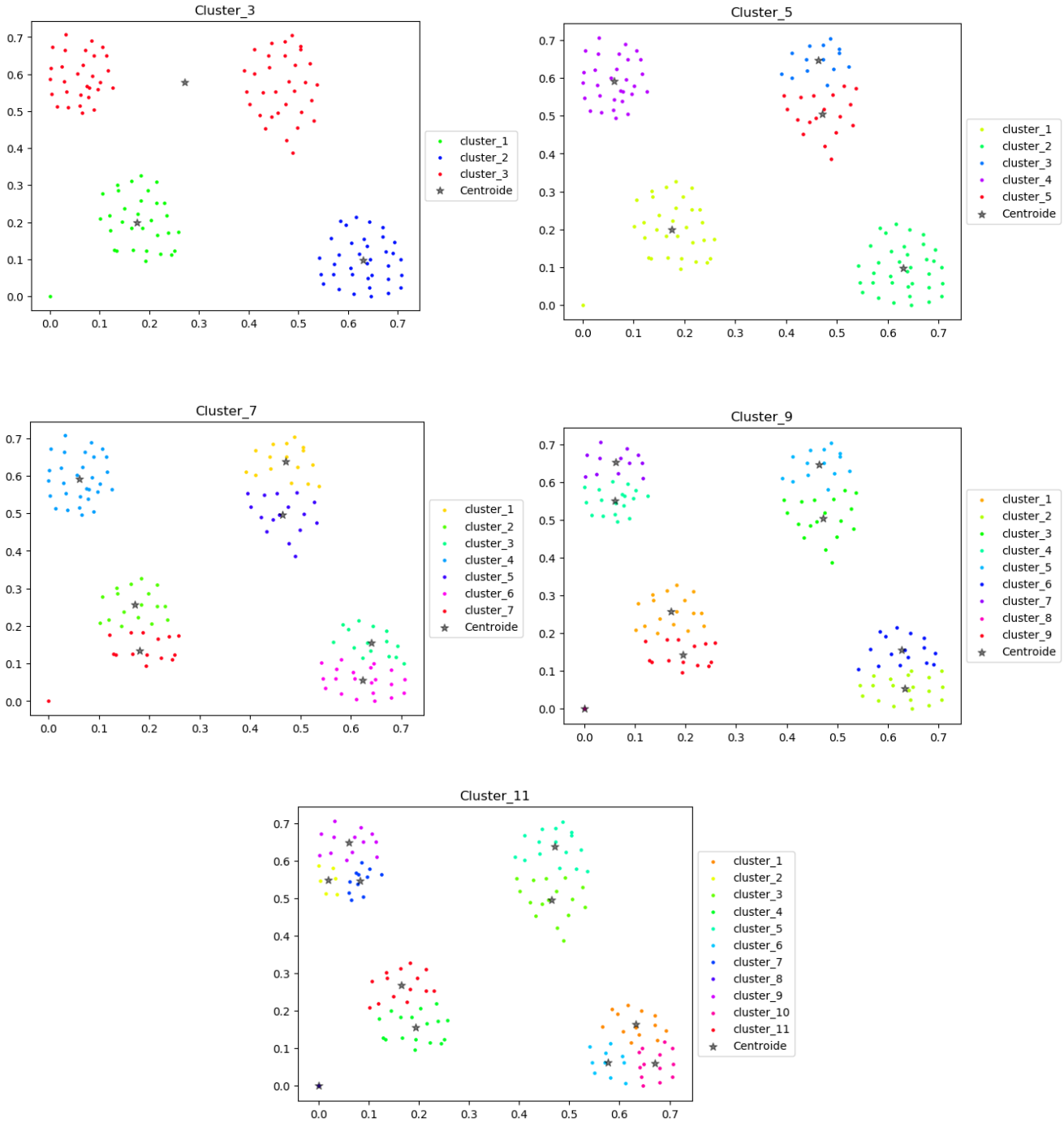

Figure 9: Several K-mean clustering using given dataset.

In this project, a dataset of 23 excel sheet was given. The above figures show the k-mean clustering results of first excel sheet form the dataset.

5. References

[1] G. Forestier, “Collaborative clustering: introduction, knowledge integration and application”, University of Strasbourg, 2009.

[2] A. Cornuéjols, C. Wemmert, P. Gançarski and Y. Bennani, "Collaborative clustering: Why, when, what and how", Information Fusion, vol. 39, pp. 81-95, 2018.

[3] J. Sublime, N. Grozavu, G. Cabanes, Y. Bennani and A. Cornuéjols, "From horizontal to vertical collaborative clustering using generative topographic maps", International Journal of Hybrid Intelligent Systems, vol. 12, no. 4, pp. 245-256, 2016.