Introduction to Machine Learning
Lab Session 5

Group 39
Lubor Budaj (s4167376) & Gasan Rzaev (s3553213)

October 2, 2022

1. Introduction

The goal of this assignment is to implement the agglomerative hierarchical clustering algorithm. using three different linkage measures: *Single*, *Average*, *Complete* and *Ward's* linkage functions. For the distance measurements we will use simple (squared) Euclidean distance. We will provide the results for the division 2, 3 and 4 clusters ($K \in \{2, 3, 4\}$).

The dimensions of the data set are 200 by 2 - it contains 200 samples. For each sample contains x feature and y feature. The data is unlabeled.

In this report we will first introduce the agglomerative hierarchical clustering in the method section. Next we will provides the result of our experiments on the given data set in the Results section. Lastly, in the discussion section we will interpret the results.

2. Method

Agglomerative hierarchical algorithm is a bottom-up clustering algorithm, which means each cluster in the beginning consists of only one data point (singleton), then it computes the proximity matrix (matrix that consists of all distances between clusters) using simple Euclidean distance (in this implementation) and then enters a loop of merging two closest clusters and updating the proximity matrix until there is only one cluster left. However, it is important to mention the way the distance between clusters is computed, since clusters have more than one data points in the later iterations of the algorithm. We will consider 4 ways to compute the distance between the clusters (linkage measures) namely, *Single linkage*, *Complete linkage*, *Average linkage* and *Ward's linkage*.

- **Single linkage:** Evaluates the distance between the 2 closest data points from 2 different clusters, in other words the smallest distance between two data points of two clusters.
- **Complete linkage:** Evaluates the distance between the 2 furthest data points from 2 different clusters, which is the opposite of Single linkage measure.
- **Average linkage:** Evaluates the sum of all pairs of data points from each cluster and then divides it by the amount of pairs, in other words it takes mean of distances between all pairs of two clusters.
- **Ward's linkage:** Instead of evaluating the distance directly, this method analyzes the variance of clusters and keeps the increase in sum of squares when merged as small as possible. In other words, it combines clusters, such that the increase within cluster variance is minimal.

Our implementation is modular. It works for different number of data-points, linking measures and clusters. Apart from using the built-in function to calculate the silhouette scores, we

implemented our own method to calculate the silhouette score. It is based on the silhouette score formula.

3. Results

In this section the results of the implemented algorithm are presented in two different ways: **Qualitative** and **Quantitative** results.

Qualitative results consist of dendrograms with dashed horizontal lines that indicate the cut-off threshold for different amount of clusters, scatter plot of original data points before the clustering and the scatter plot of data points after the hierarchical clustering. All the results will be provided for all linkage measures used in our implementation and different choices of the number of clusters K.

Quantitative results will show a table listing the computed silhouette score for all different linkage measures and different choices of the number of clusters.

3.1. Qualitative results

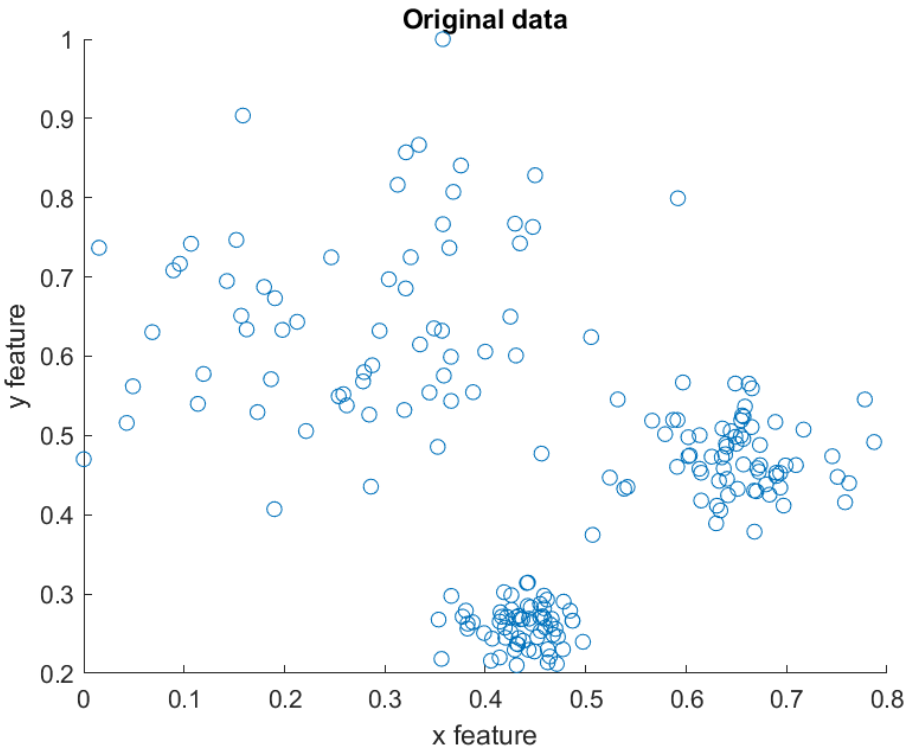

Figure 1

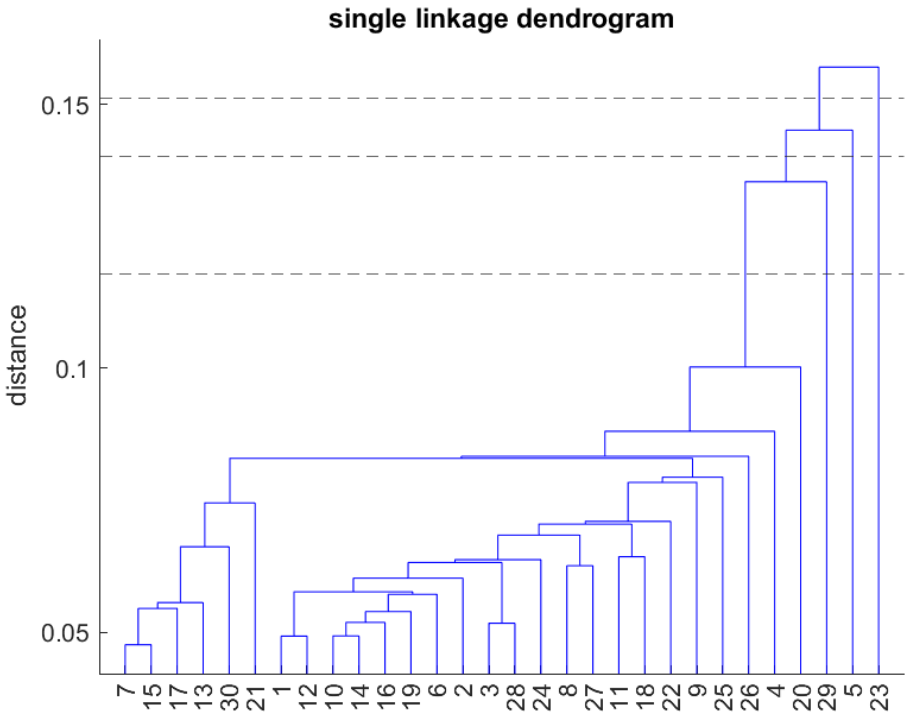

Figure 2

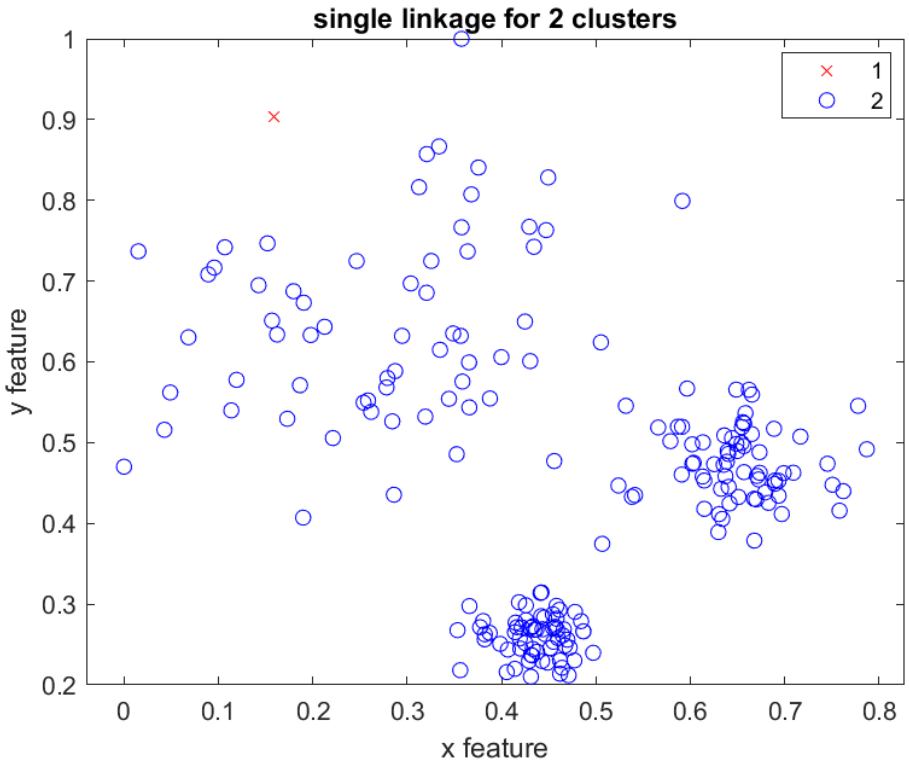

Figure 3

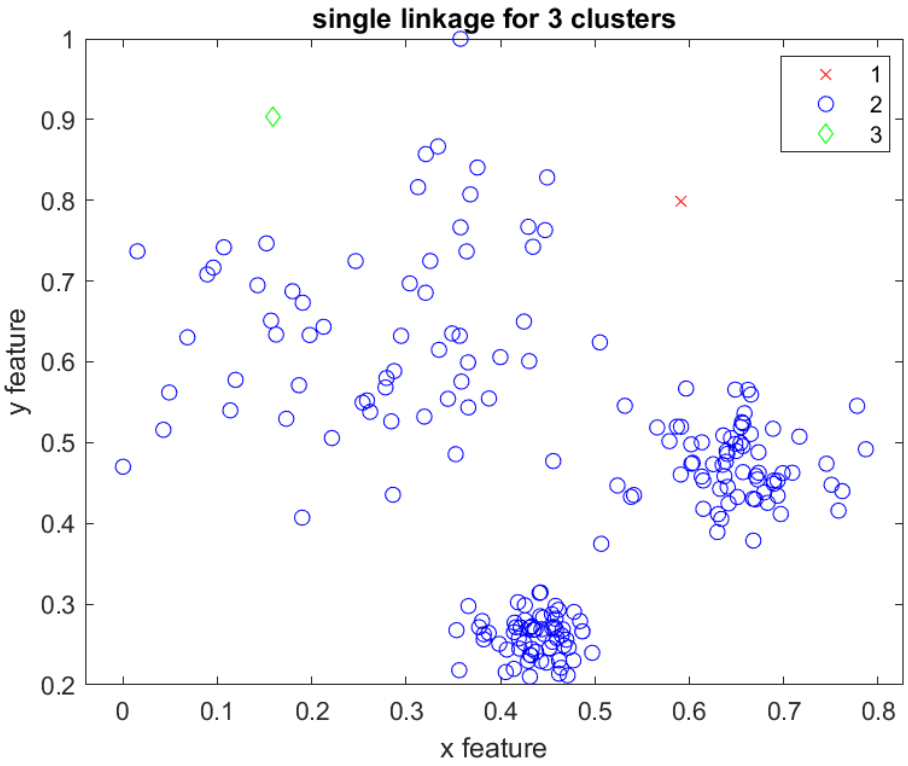

Figure 4

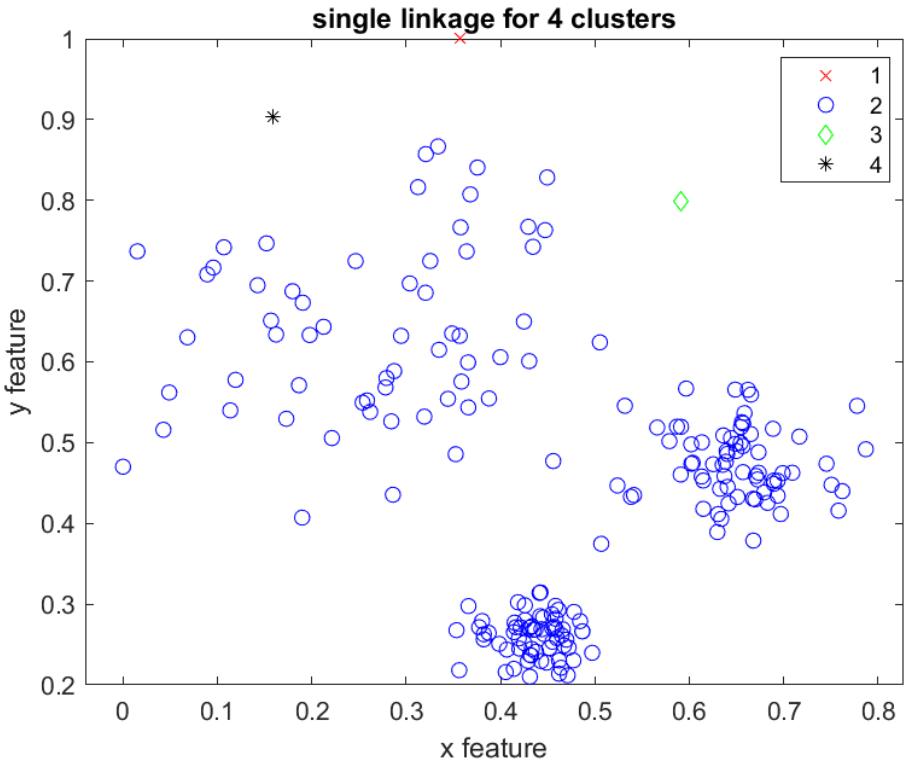

Figure 5

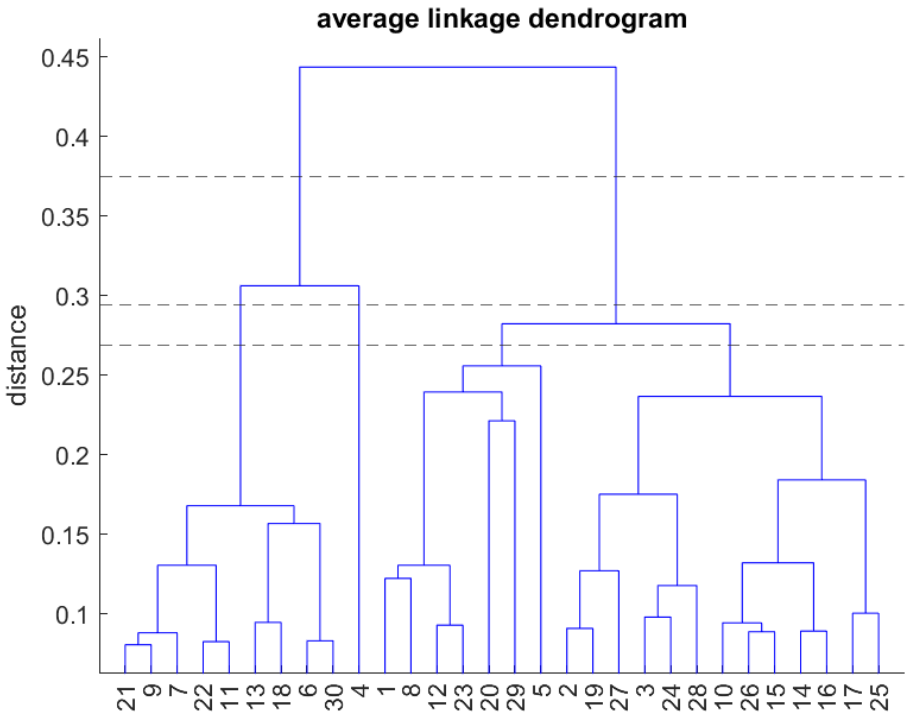

Figure 6

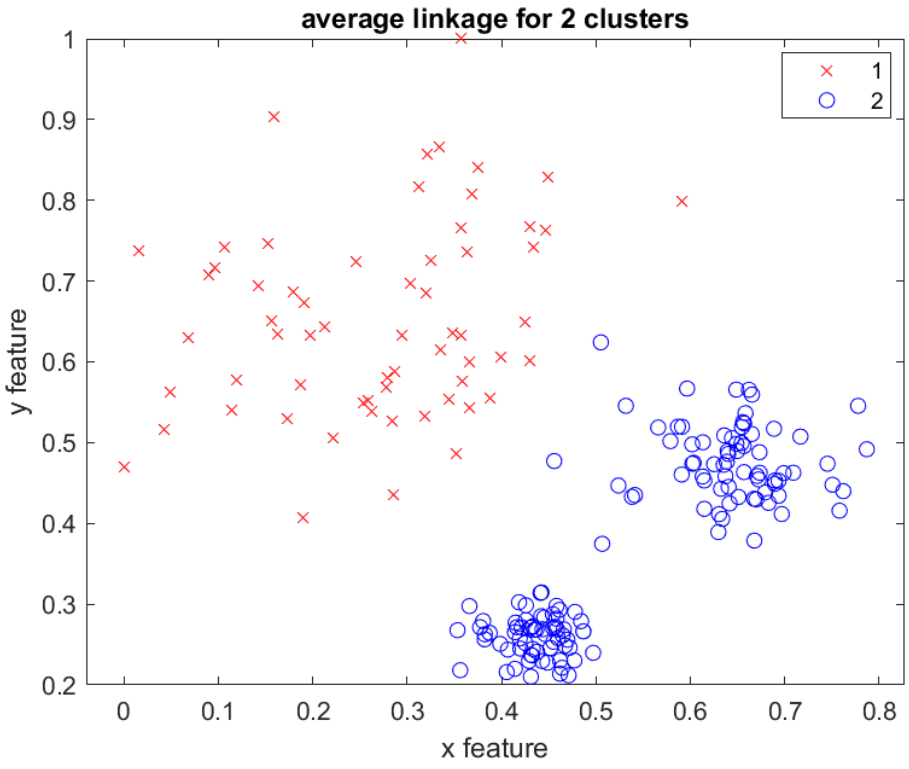

Figure 7

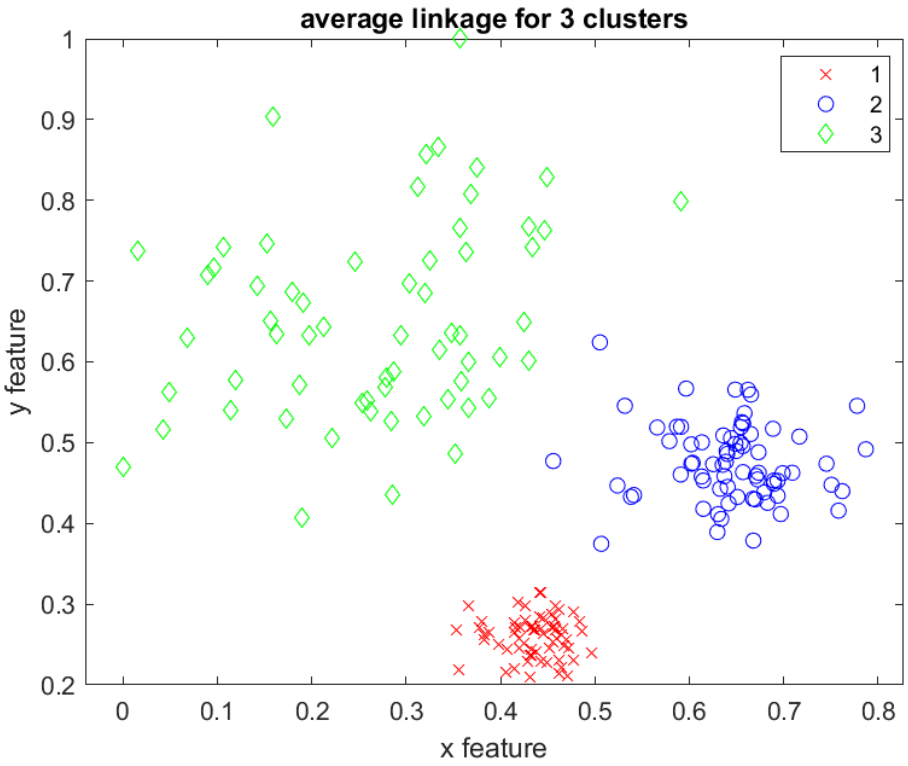

Figure 8

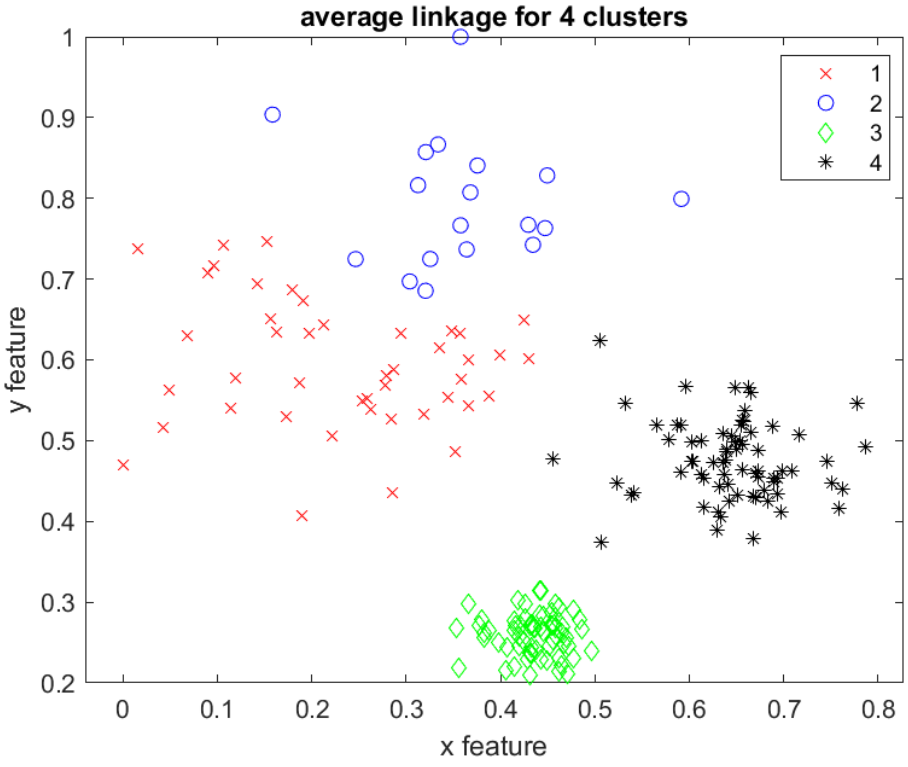

Figure 9

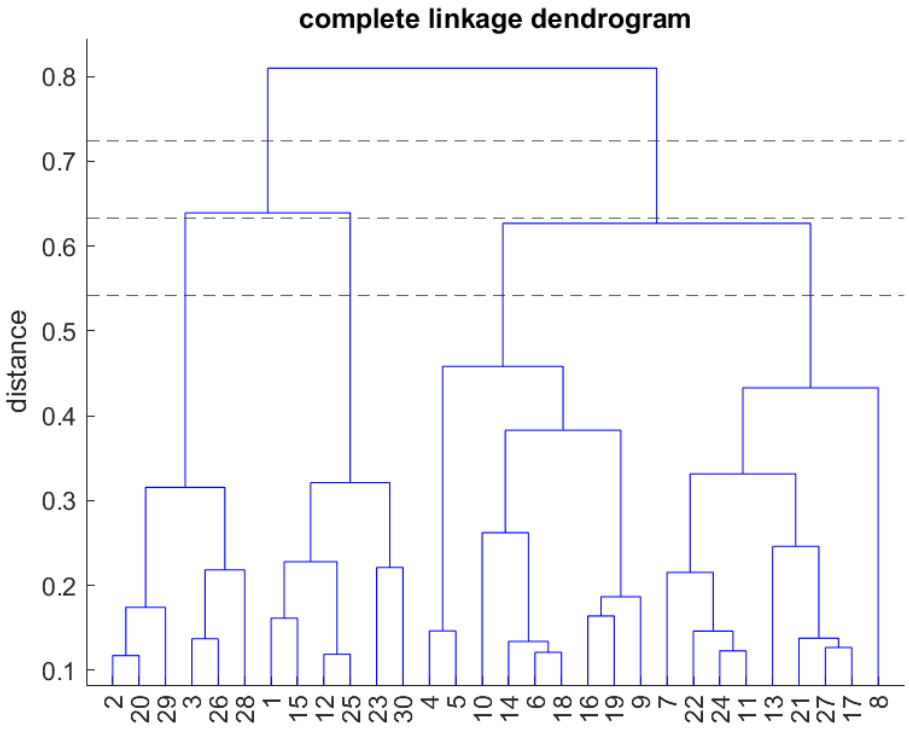

Figure 10

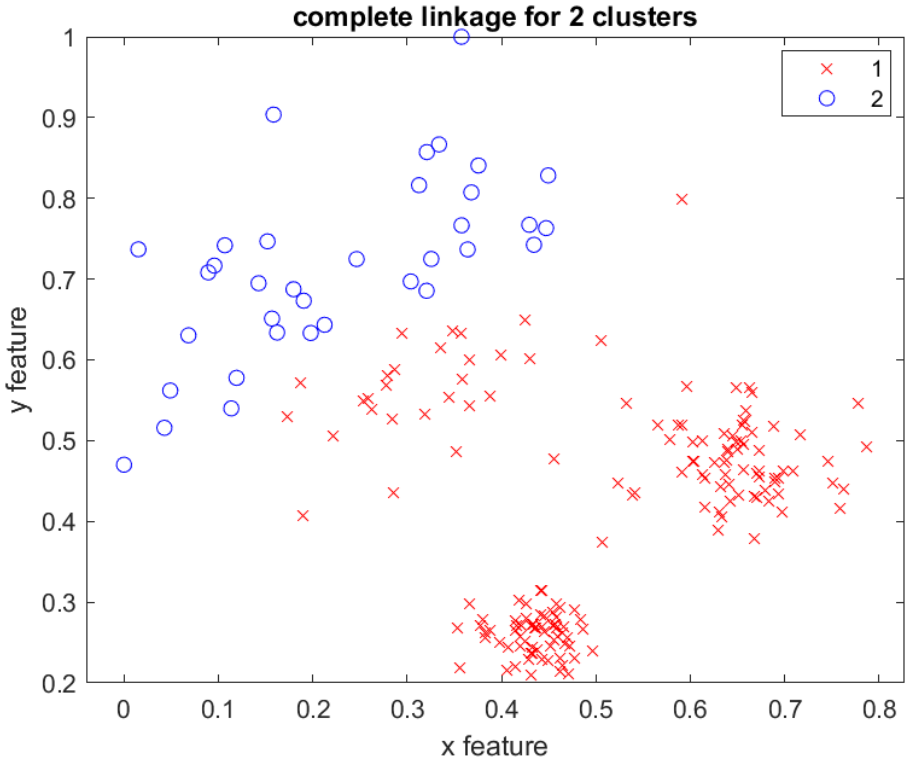

Figure 11

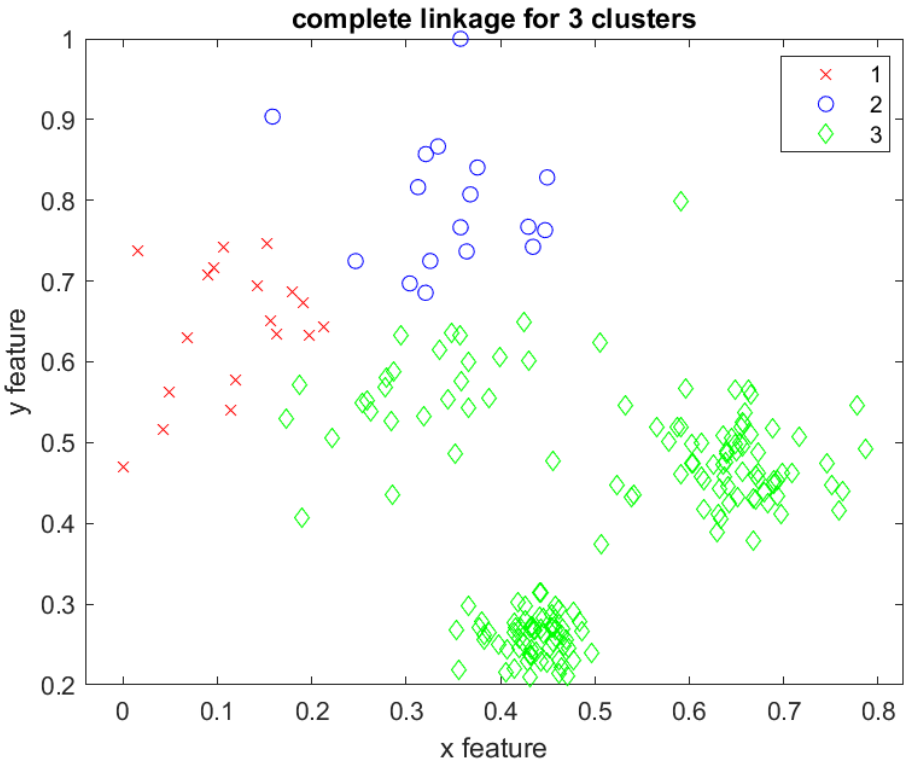

Figure 12

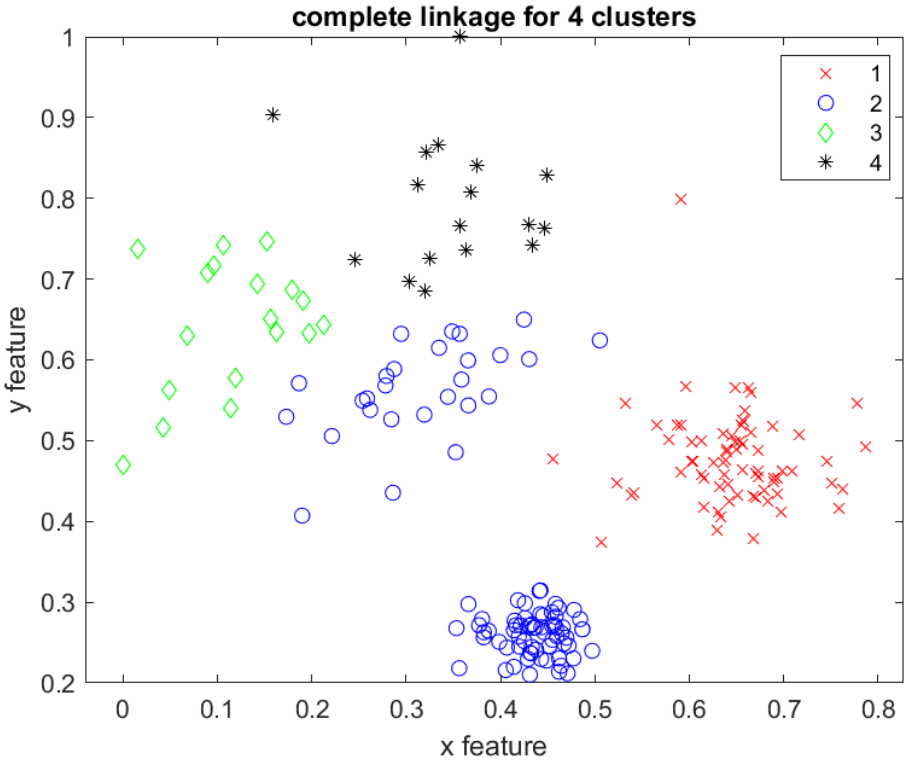

Figure 13

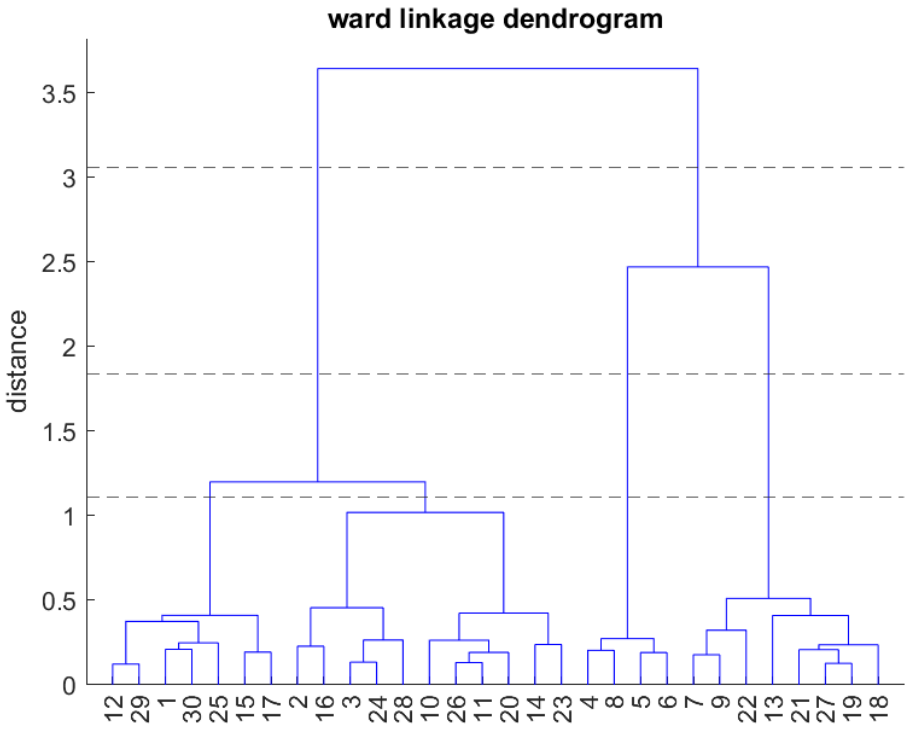

Figure 14

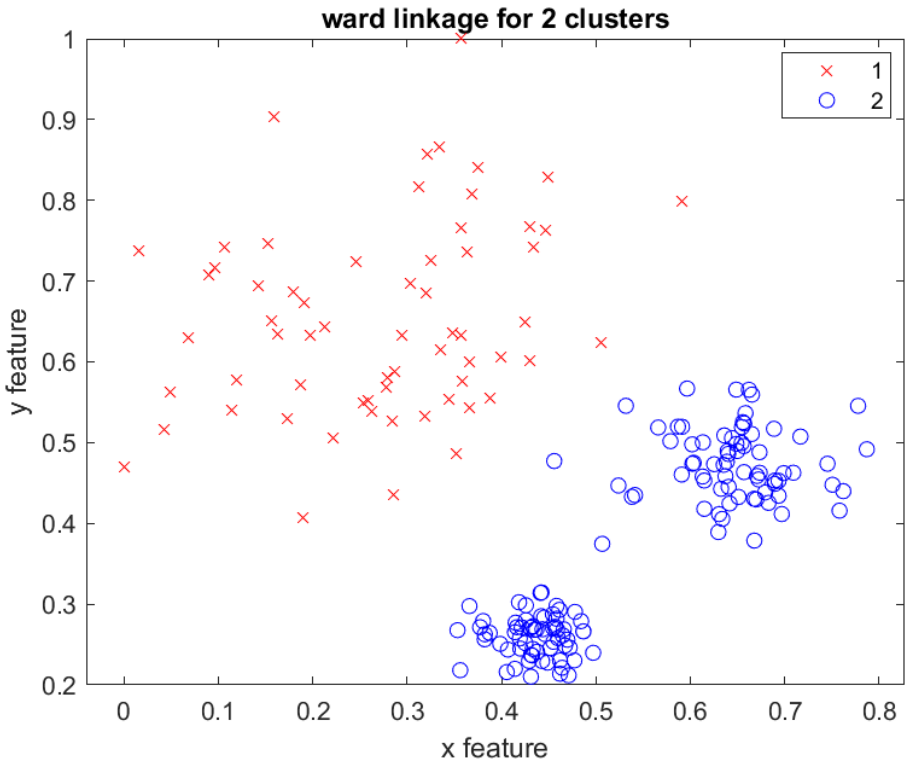

Figure 15

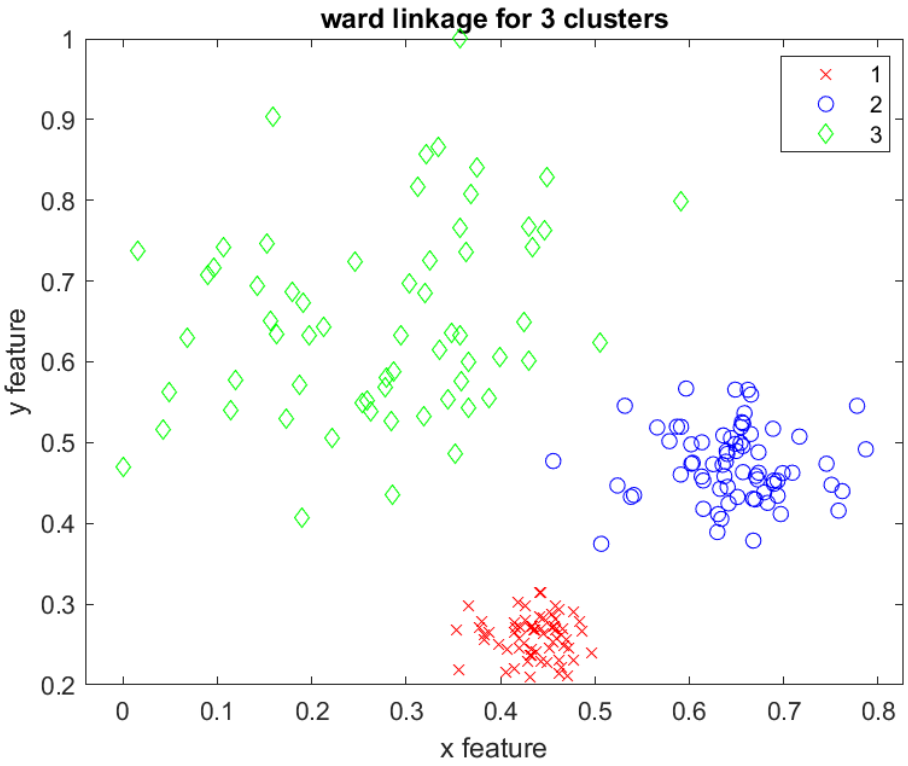

Figure 16

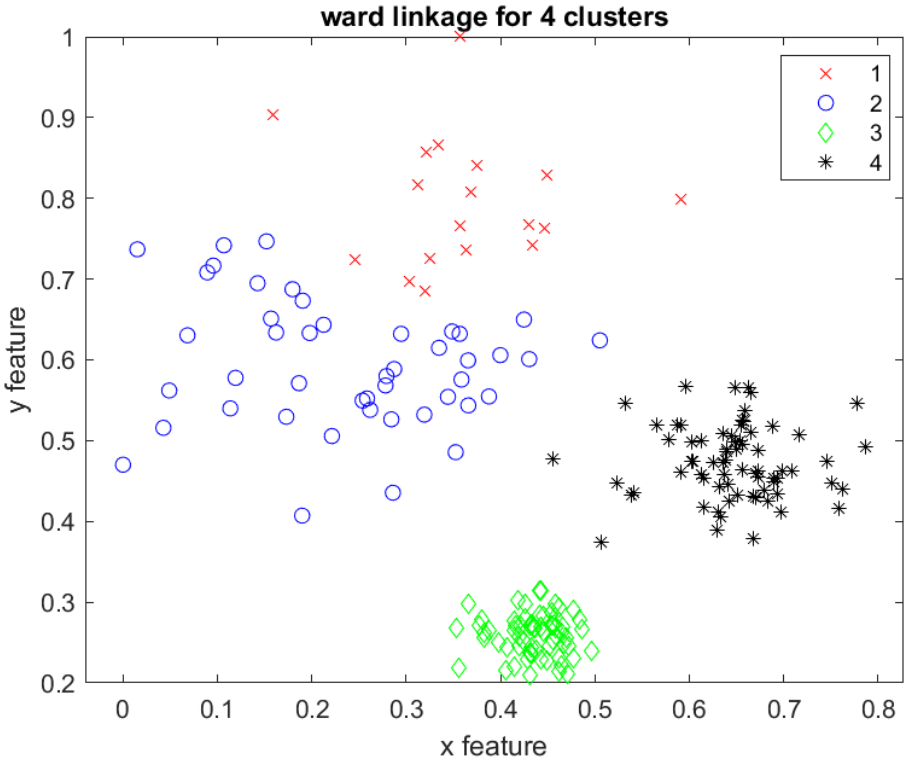

Figure 17

3.2. Quantitative results

clusters/linkage measure	Single	Average	Complete	Ward's
2	0.4711	0.7207	0.6101	0.7207
3	0.1669	0.8049	0.5500	0.8027
4	0.1816	0.7789	0.5852	0.7744

Figure 18: *Silhouette scores for different number of cluster and different linkage measures using built-in function*

clusters/linkage measure	Single	Average	Complete	Ward's
2	0.4661	0.7207	0.6101	0.7207
3	0.1569	0.8049	0.5500	0.8027
4	0.1666	0.7789	0.5852	0.7744

Figure 19: *Silhouette scores for different number of cluster and different linkage measures using our own implementation for Silhouette scores function*

4. Discussion

First of all, let's consider the qualitative results. In Figure 1 we can see the scatter plot for original data points. Observing this plot it seems that there appear to be 2 clusters of data, one on the right and one on the bottom. Other than that there are other data points close to the centre, top, and left parts of the plot. We think that there should be another cluster for these data points, but it is not as distinct as first 2 clusters.

Next, we will compare the dendrograms for different linking measures. In the dendrogram for *Single* linking measure (Figure 2), we can see that the distribution of data points among any number of clusters (2-4) is unequal. For any number of clusters (2-4), all except one data point per cluster seem to be in only one cluster. This doesn't match our intuition described in the previous paragraph, therefore we don't have a high expectations from using of this linkage method. Other linkage methods provides more evenly spread participation among clusters. When comparing the remaining 3 dendograms (Figure 6, Figure 10 and Figure 14), based on the height of vertical lines, the biggest dissimilarity between different clusters before joining them together is for *Ward's* linkage for 2 and 3 clusters, therefore we expect these 2 combinations of linkage and number of clusters to provide the best results. Using, the same metric, for *Complete* linkage we expect the best result for 2 clusters and using *Average* linkage for 2 clusters.

Next we will consider the scatter plots with all data-points assigned to a given number of cluster using *Single* linkage. In figures (Figure 3, Figure 4 and Figure 5) we can see the data points assigned to cluster with single linkage for 2, 3 and 4 clusters. In all 3 plots, there are is one big clusters containing almost all data points and 1, 2 or 3 clusters containing only 1 data-point. This is caused by the outliers in data (these 3 data points), as *Single* linkage is based on the closest distance between 2 data points from different clusters. Our expectation from the dendrogram for *Single* linkage is indeed the case. Based on this allocation of data points to clusters we don't expect silhouette scores to be high neither for *Single* linkage.

Considering the scatter plots for *Average* linkage (Figure 7, Figure 8 and Figure 9), we can see that the distribution of the data points to the clusters is not as extreme as in the case of *Single* linkage. For each number of clusters there appear to be a distinct bound between different clusters. Especially the division into 3 clusters using *Average* linkage matches our intuition for the

distribution into the clusters, as described in the first paragraph of the discussion. We expect high silhouette scores for all cases of *Average* linkage.

In the scatter plots (Figure 11, Figure 12 and Figure 13) we can see data points assigned to 2,3 and 4 clusters using *Complete* linkage. For all 3 numbers of clusters, there seem to be a bound between different clusters, but these bounds don't match our expectation. Therefore we expect silhouette scores for this type of linkage to be higher than for *Single* linkage, but small than for *Average* linkage.

Lastly, in the scatter plots (Figure 15, Figure 16 and Figure 17) we can see the clusters for *Ward's* linkage. The clusters are very similar to the results of *Average* linkage and match our expectation about the clusters, especially the division into 3 clusters.

4.1. Optimal linkage measures and number of clusters

In Figure 18 and Figure 19, there can be seen the silhouette scores for different linkage measures and for different number of clusters. The highest silhouette scores were achieved by the combination of using 3 clusters and the *Average* or *Ward's* linkage measures. The worst result we achieved by using 3 or 4 clusters with *Single* linkage measure. This confirms that are initial expectation about the division of data point into the clusters - 3 clusters, a dense one on the bottom of the plot, another dense cluster on the right and the sparse cluster with the rest of the data-points - is indeed the the best clustering achieved in this experiment. Our other expectations, from the previous subsection, about the silhouette scores for different different linkage measures were also met. *Single* linkage measure is the worst performing one, although we expected even lower score for 2 clusters. The relatively high score of 0.4711 is probably caused by only having 2 clusters, in which one contains only 1 data-point. As we expected, the complete linkage measure results are better than for *Single* linkage measure, but smaller than for *Average* and *Ward's* linkage measure.

4.2. Comparing built-in and our own silhouette score function

In Figure 18 there are silhouette scores calculated using the built-in silhouette functions. In Figure 19 there are silhouette scores calculated using our won implementation based on the formula for silhouette score. The silhouette scores are identical for *Average*, *Complete* and *Ward's* linking measures, however they are slightly different *Single* linking measure. We don't know what caused the difference between the 2 results.

5. Work Distribution

The work among the group members was distributed in a following way:

- Code: 50% done by Lubor, 50% by Gasan
- Report: 50% done by Lubor, 50% by Gasan
- Graphs: 50% done by Lubor, 50% by Gasan

The code used to complete this assignment is presented in Listing 1 in Appendix A.

A. Appendix

Listing 1: *This is the file* `code.m`

```
%import
data = readmatrix("data_clustering.csv");

%initialization of dimensions of the data
dimensions = size(data);
N = dimensions(2);
P = dimensions(1);

%number of clusters
K_min = 2;
K_max = 4;

%linkage measures
linkage_measures = categorical(["single", "average", "complete", "ward
    "]);
M = length(linkage_measures);

%initialization of silhouette scores matrix
s_scores = zeros(K_max - K_min + 1, M);

%scatter plot
figure(1);
scatter(data(:,1),data(:,2));
xlabel("x feature");
ylabel("y feature");
title("Original data");

fig_num = 2;
for K = K_min:K_max
    for i = 1:M
        s_scores(K-K_min+1, i) = hierarchical_clustering(data, char(
    linkage_measures(i)), K, fig_num, K == K_max, false);
        fig_num = fig_num + 2;
    end
end


function s_score = hierarchical_clustering(data, link, K, fig_num,
    dendro, buildin_sil)

    %clustering
    Z = linkage(data, link);
    T = cluster(Z, "maxclust", K);

    %scatter plot
    figure(fig_num);
    gscatter(data(:,1),data(:,2),T,'rbgk','xod*');
```

```
    xlabel("x feature");
    ylabel("y feature");
    title(link + " linkage for " + K + " clusters");

    %dendrogram
    if dendro
        figure(fig_num + 1);
        dendrogram(Z);
        title(link + " linkage dendrogram");
        ylabel("distance")
        for i = 1:(K-1)
            yline(median([Z(height(Z)+1-i,3), Z(height(Z)-i,3)]), '--'
   );
        end
    end

    %silhouette score
    if buildin_sil
        s_score = sum(silhouette(data, T)) / length(T);
    else
        s_score = 0;
        C_length = zeros(1, K);
        for i = 1:K
            C_length(i) = nnz(T == i);
        end
        for i = 1:length(T)
            if C_length(T(i)) > 1
                a = 0;
                b = zeros(1, K);
                for j = 1:length(T)
                    if i ~= j
                        dist = (data(i,1) - data(j,1))^2 + (data(i,2) -
    data(j,2))^2;
                        if T(i) == T(j)
                            a = a + dist;
                        else
                            b(T(j)) = b(T(j)) + dist;
                        end
                    end
                end
                a = a / (C_length(T(i)) - 1);
                for j = 1:K
                    b(j) = b(j) / (C_length(j));
                end
                B = min(b(b > 0));
                s_score = s_score + ((B-a) / max(a,B));
            end
        end
        s_score = s_score / length(T);
    end
end
```