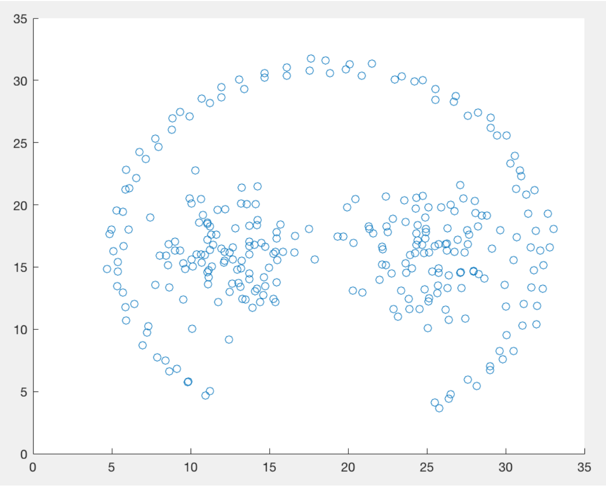
Machine Learning 3rd Assignment

Ida Bagus Dwi Satria Kusuma - 1301140297 - Telkom University 07/12/2017

Problem 1 - In this exercise, we implement partitional clustering method: K-means algorithm.

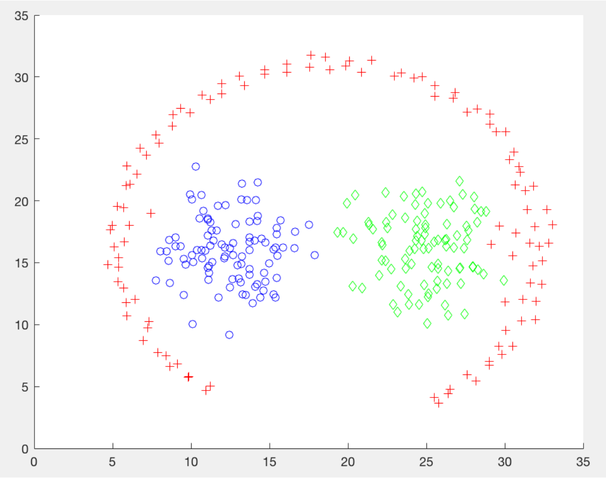
a Load the selected data set. Visualize all data points using scatter plot in one color (no need to give different color for different label). Use only attribute 1 as x -axis and attribute 2 as y -axis. This visualization might give you a brief thought of existed clusters. [5 points]

Figure 1: Visualisasi data tanpa label kelas



b Again, visualize all data points using scatter plot. Use attribute 1 as x -axis, attribute 2 as y -axis. Use different color and/or different symbol for each label. This visualization shows you the real existed clusters. [5 points]

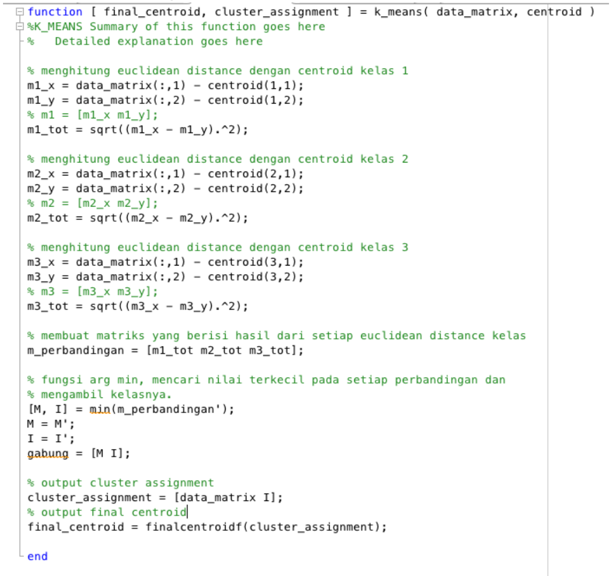
Figure 2: Visualisasi data dengan label kelas



c Apply K-means on the selected data set. Your codes have to clearly contain

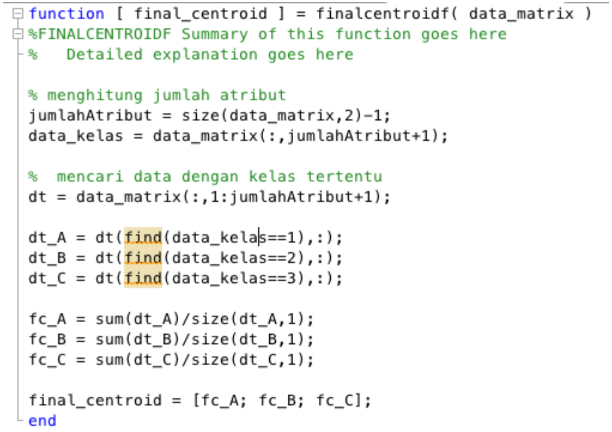
1. Function that takes as inputs: the data matrix and initial centroids; as outputs: the final centroids and the cluster assignments specifying which data vectors are assigned to which centroids after convergence of the algorithm. (To speed up the algorithm sufficiently, use matrix operations wherever possible and avoiding explicit loops). [15 points]

Figure 3: Code: k_means.m



```
function [ final_centroid, cluster_assignment ] = k_means( data_matrix, centroid )
%K_MEANS Summary of this function goes here
%   Detailed explanation goes here

% menghitung euclidean distance dengan centroid kelas 1
m1_x = data_matrix(:,1) - centroid(1,1);
m1_y = data_matrix(:,2) - centroid(1,2);
% m1 = [m1_x m1_y];
m1_tot = sqrt((m1_x - m1_y).^2);

% menghitung euclidean distance dengan centroid kelas 2
m2_x = data_matrix(:,1) - centroid(2,1);
m2_y = data_matrix(:,2) - centroid(2,2);
% m2 = [m2_x m2_y];
m2_tot = sqrt((m2_x - m2_y).^2);

% menghitung euclidean distance dengan centroid kelas 3
m3_x = data_matrix(:,1) - centroid(3,1);
m3_y = data_matrix(:,2) - centroid(3,2);
% m3 = [m3_x m3_y];
m3_tot = sqrt((m3_x - m3_y).^2);

% membuat matriks yang berisi hasil dari setiap euclidean distance kelas
m_perbandingan = [m1_tot m2_tot m3_tot];

% fungsi arg min, mencari nilai terkecil pada setiap perbandingan dan
% mengambil kelasnya.
[M, I] = min(m_perbandingan');
M = M';
I = I';
gabung = [M I];

% output cluster assignment
cluster_assignment = [data_matrix I];
% output final centroid
final_centroid = finalcentroidf(cluster_assignment);

end
```

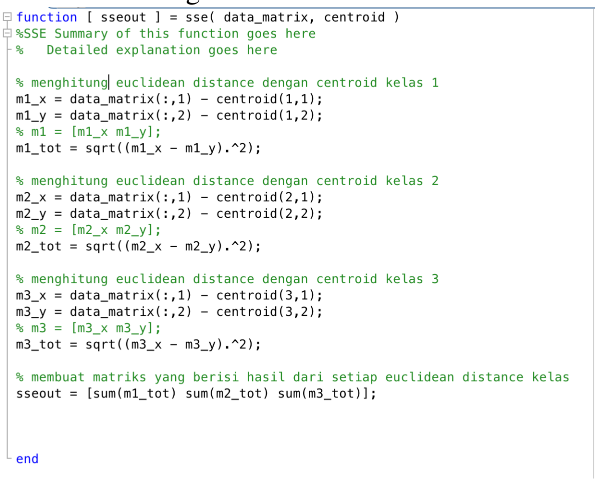
Figure 4: Code: finalcentroidf.m



```
function [ final_centroid ] = finalcentroidf( data_matrix )
%FINALCENTROIDF Summary of this function goes here
%   Detailed explanation goes here

% menghitung jumlah atribut
jumlahAtribut = size(data_matrix,2)-1;
data_kelas = data_matrix(:,jumlahAtribut+1);

%  mencari data dengan kelas tertentu
dt = data_matrix(:,1:jumlahAtribut+1);

dt_A = dt(find(data_kelas==1),:);
dt_B = dt(find(data_kelas==2),:);
dt_C = dt(find(data_kelas==3),:);

fc_A = sum(dt_A)/size(dt_A,1);
fc_B = sum(dt_B)/size(dt_B,1);
fc_C = sum(dt_C)/size(dt_C,1);

final_centroid = [fc_A; fc_B; fc_C];
end
```

2. Function to calculate the objective function of K-means that is Sum of Squared Errors (SSE). It takes as inputs: all data vectors and final centroids resulted from learning. [10 points]

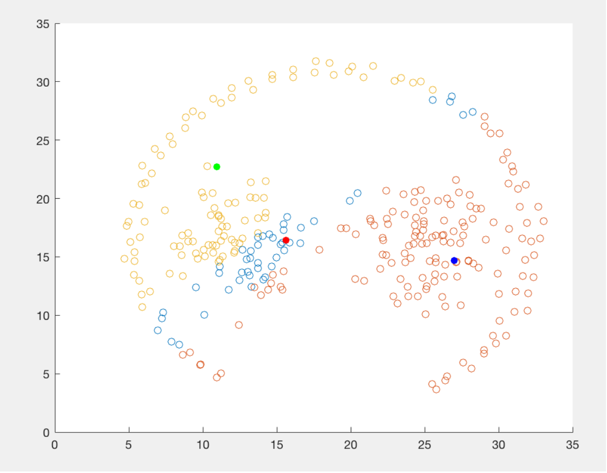
Figure 5: Code: sse.m



```
function [ sseout ] = sse( data_matrix, centroid )
%SSE Summary of this function goes here
%   Detailed explanation goes here

% menghitung euclidean distance dengan centroid kelas 1
m1_x = data_matrix(:,1) - centroid(1,1);
m1_y = data_matrix(:,2) - centroid(1,2);
% m1 = [m1_x m1_y];
m1_tot = sqrt((m1_x - m1_y).^2);

% menghitung euclidean distance dengan centroid kelas 2
m2_x = data_matrix(:,1) - centroid(2,1);
m2_y = data_matrix(:,2) - centroid(2,2);
% m2 = [m2_x m2_y];
m2_tot = sqrt((m2_x - m2_y).^2);

% menghitung euclidean distance dengan centroid kelas 3
m3_x = data_matrix(:,1) - centroid(3,1);
m3_y = data_matrix(:,2) - centroid(3,2);
% m3 = [m3_x m3_y];
m3_tot = sqrt((m3_x - m3_y).^2);

% membuat matriks yang berisi hasil dari setiap euclidean distance kelas
sseout = [sum(m1_tot) sum(m2_tot) sum(m3_tot)];



end
```

d Run your K-means algorithm, using K equals to the number of different label in dataset. The initial centroids taken from randomly selected K data points. After convergence, visualize the centroid of each cluster as well as all data points assigned to that cluster (it should be easily distinguished between the centroids and the data points, also give different color/symbol to different cluster). One may run the algorithm several times in order to obtain the best result (hints: use the SSE as the measure). [10 points]

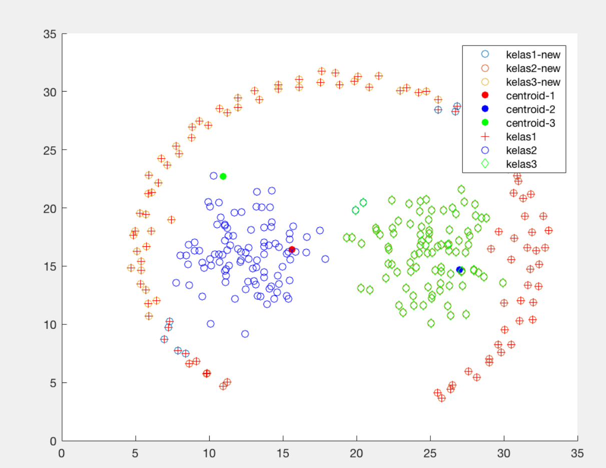
Figure 6: hasil cluster dan final centroid



e Based on visualization resulted from point 1(d), to what extent do the K clusters correspond to the K different labels? (Hints: Use the visualization from point 1(b) to compare and get a view/thought of clustering results shown by point 1(d).) [5 points]

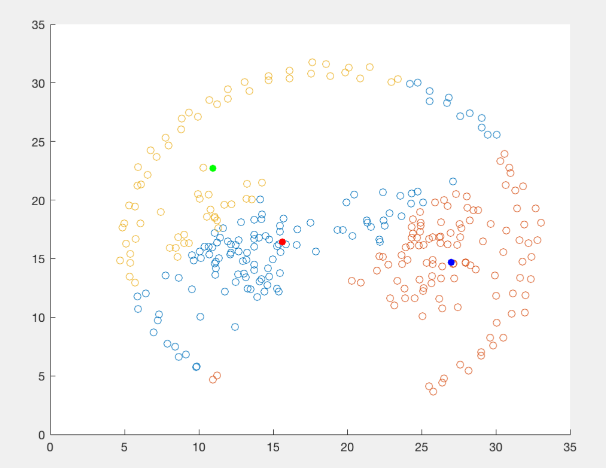
f Re-run K-means but selecting randomly one instance of each label as the initial centroid (so that the initial centroids all represent distinct label). After convergence, visualize the centroid of each cluster as well as all data points assigned to that cluster (it should be easily distinguished between the centroids and

Figure 7: hasil cluster dan final centroid dan kelas awal yang diambil random



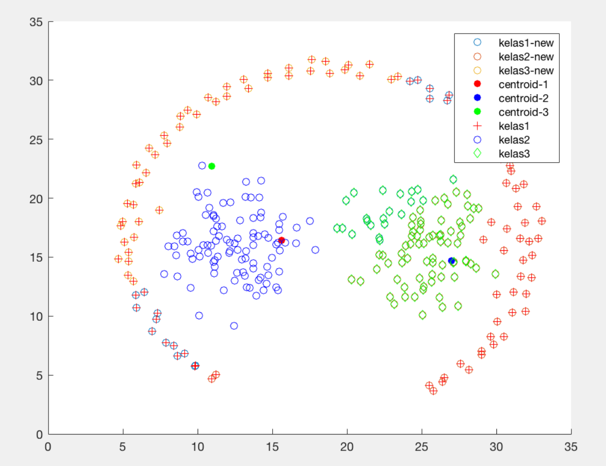
the data points, also give different color/symbol to different cluster). One may run the algorithm several times in order to obtain the best result (hints: use the SSE as the measure). [10 points]

Figure 8: hasil cluster dan final centroid dan kelas awal yang diambil random



g Based on visualization resulted from point 1(f), to what extent do the K clusters correspond to the K different labels? (Hints: Use the visualization from point 1(b) to compare and get a view/thought of clustering results shown by point 1(f).) [5 points]

Figure 9: hasil cluster dan final centroid dan kelas awal yang diambil random



h By visually comparing figures created from point 1(d) and 1(f), what do you think of the clustering results? Give explanations. [5 points]

Berdasarkan hasil pengamatan saya, hasil dari clustering masih kurang bagus, karena ada beberapa titik yang dekat dengan centroid tertentu namun tidak ikut ke dalam kelas centroid tersebut.