Name: Xinrun Zhang

ML Homework 8 Report

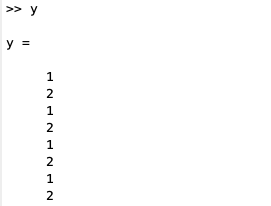
Instructor: Prof. Kaliappan Gopalan

Time: 04/15/2019

1. Using K = 5, apply K-means algorithm and develop MATLAB code to arrive at the five cluster centroids with initial centroids

After few attempts, the D is determined as 2603.5.

Also, because of the fact that the initial centroids are close to the final centroids, the loop converged after only 3 iterations.

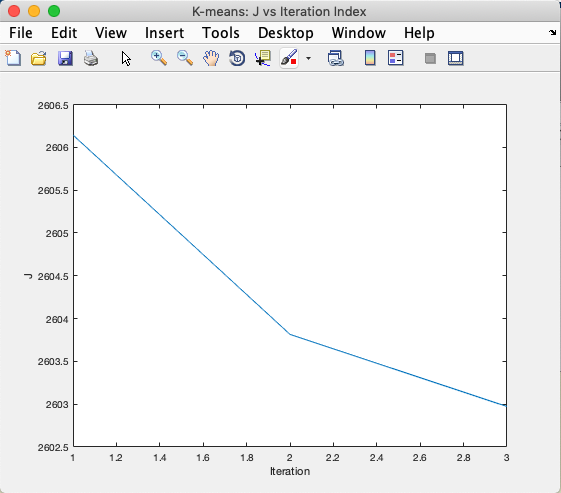


The label of each input data is stored at y and the label indicates which cluster is the input data belong to.

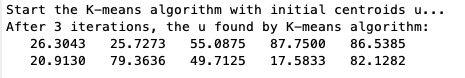
1. Plot the cost function D after each epoch as a function of epoch. Note that cluster assignment and centroid modification for all K clusters together constitute one epoch. If the error does not converge after 500 epochs, try raising D to 0.5 or so. What are the final centroids?

The final cost is:

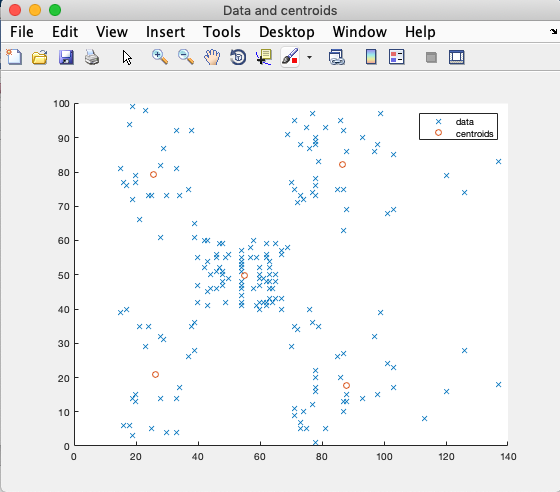




The final centroids are:



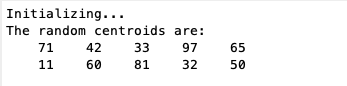
1. Plot the final centroid locations along with the data points. How many iterations does it take to converge?



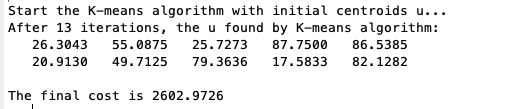
The number of iterations is 3.

1. Instead of initializing centroids with fixed values, randomize the input data (note that the income column is in an increasing order) using the function randperm and pick the first five observations as initial centroids. Print the initial and final centroids and show the plot with centroids for each run.
2. Run the code first time

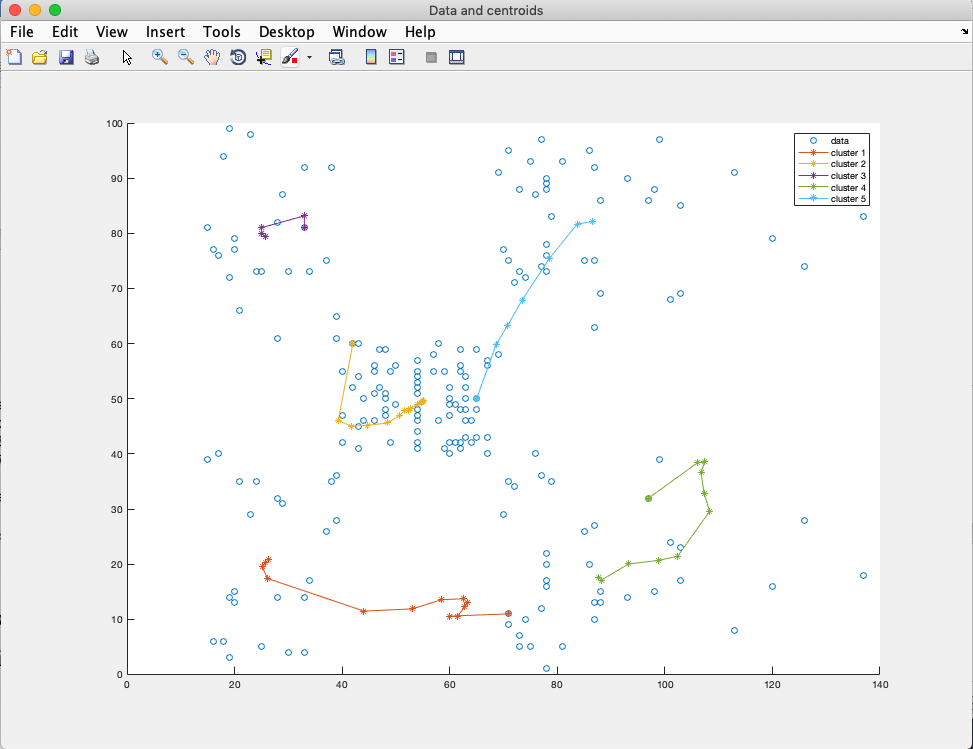
Initial centroids:



Final centroids:

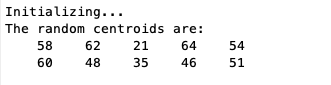


The plot with centroids with each iteration:

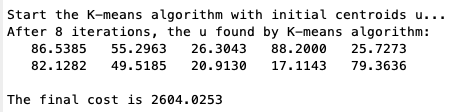


1. Run the code second time

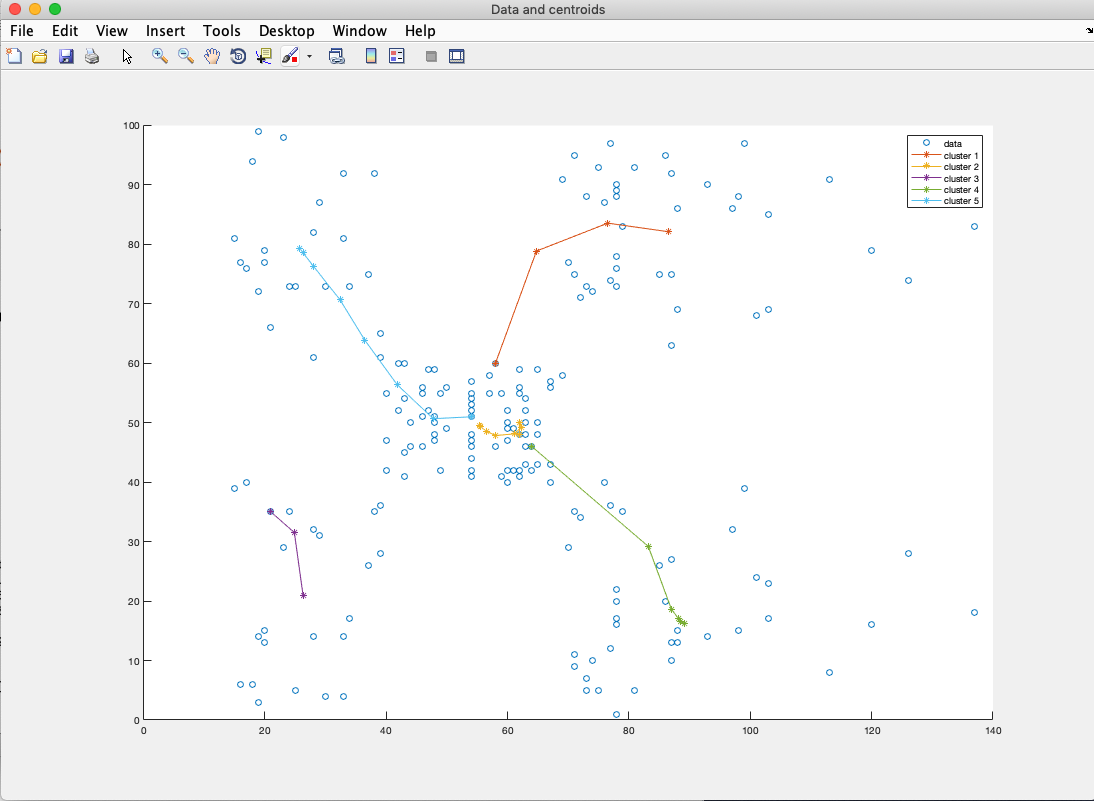
Initial centroids:



Final centroids:

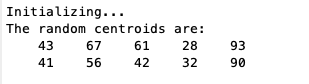


The plot with centroids with each iteration:

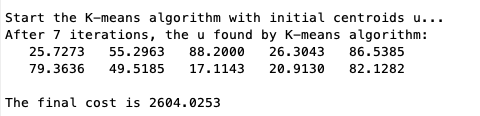


1. Run the code third time

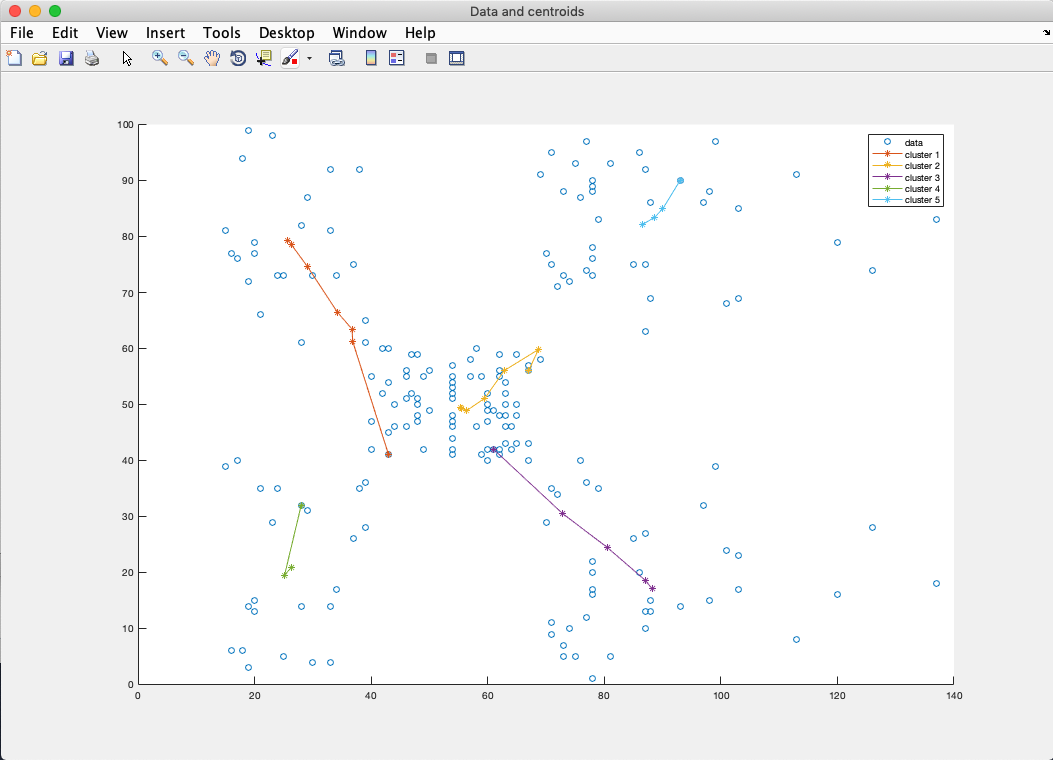
Initial centroids:



Final centroids:

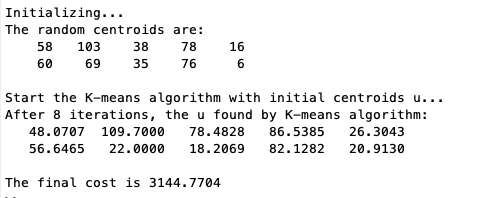


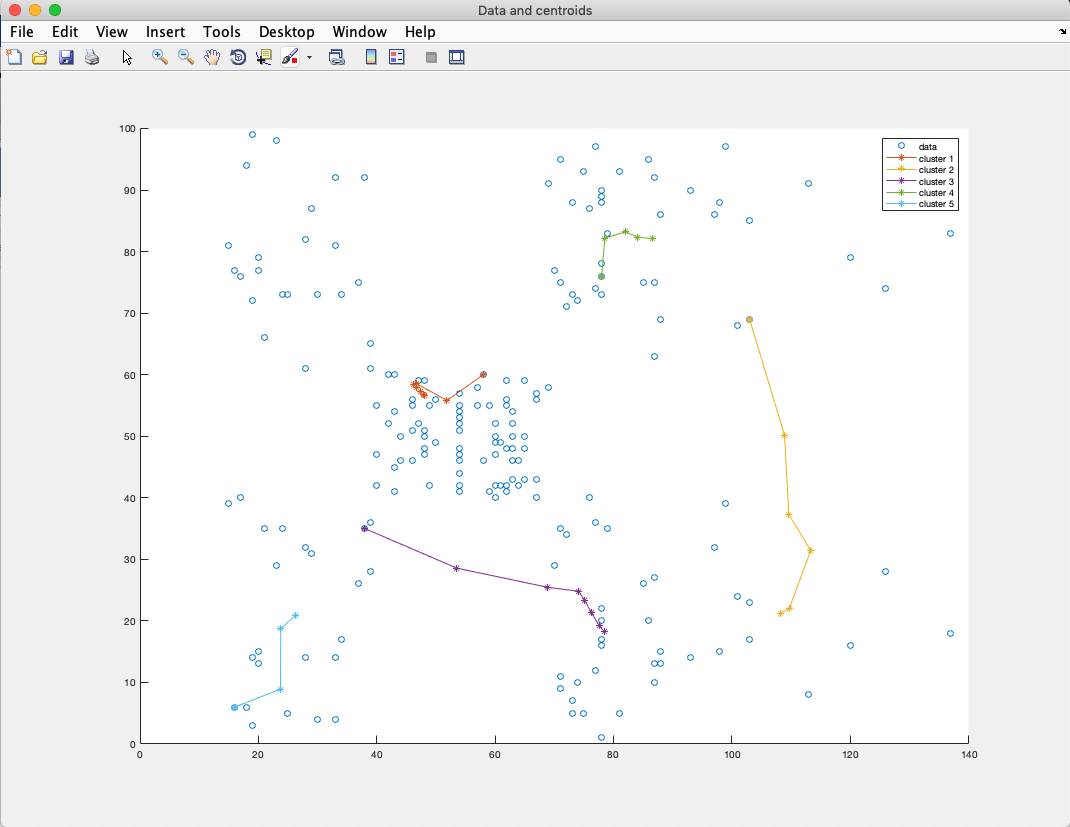
The plot with centroids with each iteration:



1. Some special cases

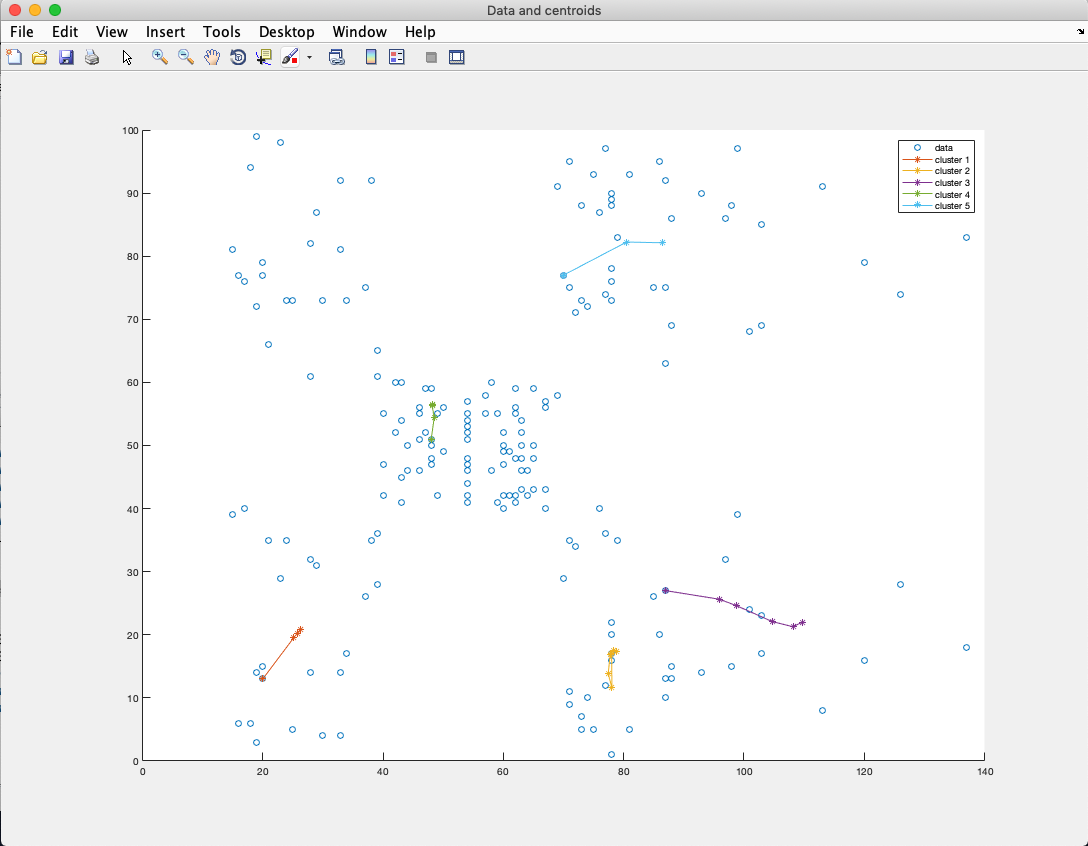
Sometimes I get a case that cost is around 3136.





From the plot, the reason is that the centroids may converge into a very close place.

Also, sometimes when I run the code, I get:



That means if the initial centroids are too close to each other, they may converge into a close cluster. However, sometimes even with close centroids, they eventually converge into different clusters.

1. Code
2. main\_part\_1.m

%% Machine Learning Homework 8 part 1

% K-means algorithm with initial centroids

% Author: Xinrun Zhang

% Time: 04/14/2019 20:22

% =====================================================================

%% Initializing

clear ; close all; clc

fprintf('Initializing...\n');

% Import the data

x = importdata('HW8.mat');

[m, ~] = size(x);

% Initial k and centroid u matrix

k = 5;

u = [20, 20, 60, 85, 80; 15, 80, 50, 20, 90];

% Initial D

D = 2603.5;

% Initial matrix y to store labels

% Initial cost J

y = zeros(200,1);

J = 1;

% =====================================================================

%% K-means algorithm with initial centroids u

% start the algorithm

fprintf('Start the K-means algorithm with initial centroids u...\n');

itr = 0;

J\_history = zeros(3,1);

while(1)

    [y, count] = calculateLabels(u, x, y, k, m);

    u = updateU(u, x, y, count, k, m);

    % compute the cost J

    itr = itr + 1;

    J\_history(itr) = computeCost(u, x, y, k, m);

    if(J\_history(itr) < D)

        break;

    end

end

fprintf('After %d iterations, the u found by K-means algorithm:\n', itr);

disp(u);

fprintf('The final cost is %.4f\n', J\_history(itr));

% =====================================================================

%% Plot

figure('Name','Data and centroids','NumberTitle','off');

scatter(x(:,1), x(:,2),'x');

hold on;

scatter(u(1,:), u(2,:), 'o');

legend('data', 'centroids')

figure('Name','K-means: J vs Iteration Index','NumberTitle','off');

plot(J\_history); % plot J vs iteration index

ylabel('J');

xlabel('Iteration');

1. main\_part\_2.m

%% Machine Learning Homework 8 part 2

% K-means algorithm with random centroids

% Author: Xinrun Zhang

% Time: 04/15/2019 16:00

% =====================================================================

%% Initializing

clear ; close all; clc

fprintf('Initializing...\n');

% Import the data

x = importdata('HW8.mat');

x = x(randperm(200),:);

[m, ~] = size(x);

% Initial k

% Initial random centroids

k = 5;

u = [x(1:5,1)';x(1:5,2)'];

fprintf('The random centroids are:\n');

disp(u);

% Initial matrix y to store labels

% Initial cost J

y = zeros(200,1);

J = 1;

% =====================================================================

%% K-means algorithm with random centroids u

% start the algorithm

fprintf('Start the K-means algorithm with initial centroids u...\n');

itr = 0;

% initial a cell to store centroids

u\_store = cell(20,1);

while(1)

    itr = itr + 1;

    % store the centroids

    u\_store{itr} = u';

    u\_old = u;

    [y, count] = calculateLabels(u, x, y, k, m);

    u = updateU(u, x, y, count, k, m);

    % compute the cost J

    J = computeCost(u, x, y, k, m);

    if(norm(u\_old - u) <= 0.0001)

        break;

    end

end

fprintf('After %d iterations, the u found by K-means algorithm:\n', itr);

disp(u);

fprintf('The final cost is %.4f\n', J);

% =====================================================================

%% plot

c1\_history = zeros(itr, 2);

c2\_history = zeros(itr, 2);

c3\_history = zeros(itr, 2);

c4\_history = zeros(itr, 2);

c5\_history = zeros(itr, 2);

figure('Name','Data and centroids','NumberTitle','off');

scatter(x(:,1), x(:,2),'o');

hold on;

for i = 1:20

    if(~isempty(u\_store{i}))

        c1\_history(i,:) = u\_store{i}(1,:);

        c2\_history(i,:) = u\_store{i}(2,:);

        c3\_history(i,:) = u\_store{i}(3,:);

        c4\_history(i,:) = u\_store{i}(4,:);

        c5\_history(i,:) = u\_store{i}(5,:);

    end

end

plot(c1\_history(:,1),c1\_history(:,2),'-\*');

hold on;

plot(c2\_history(:,1),c2\_history(:,2),'-\*');

hold on;

plot(c3\_history(:,1),c3\_history(:,2),'-\*');

hold on;

plot(c4\_history(:,1),c4\_history(:,2),'-\*');

hold on;

plot(c5\_history(:,1),c5\_history(:,2),'-\*');

hold on;

legend('data', 'cluster 1', 'cluster 2', 'cluster 3', 'cluster 4', 'cluster 5')

hold off;

1. calculateLabels.m

function [y, count] = calculateLabels(u, x, y, k, m)

d = zeros(1,5);

count = zeros(5,1); % Initial matrix count to store the number of each labels

for i = 1:m % for each data point

    v = u - x(i,:)'; % calculate the distance from data point to each centroid

    for j = 1:k

        d(j) = sqrt(v(:,j)'\*v(:,j));

    end

    index = find(d == min(d)); % find the min distance

    y(i) = index; % add the label

    count(index)  = count(index) + 1; % count the number of this label

end

end

1. updateU.m

function u = updateU(u, x, y, count, k, m)

for j = 1:k % for each category/label

    sum\_c = zeros(1,2); % calculate the new centroids

    for i = 1:m

        if(y(i) == j)

            sum\_c = sum\_c + x(i,:);

        end

    end

    u(:,j) = (sum\_c') / count(j);

end

end

1. computeCost.m

function J = computeCost(u, x, y, k, m)

J = 0;

for j = 1:k

    for i = 1:m

        if(y(i) == j)

            J = J + norm(u(:,j) - x(i,:)');

        end

    end

end

end