Name: Xinrun Zhang

ML Homework 8 Report

Instructor: Prof. Kaliappan Gopalan

Time: 04/15/2019

a) 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.

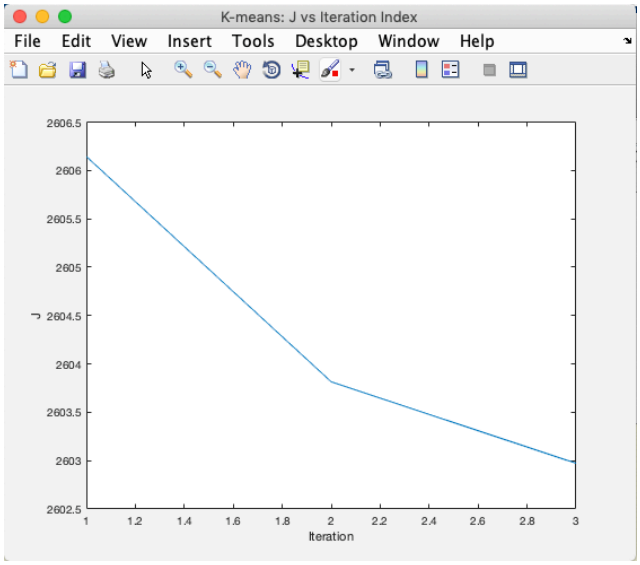
```
>> y

y =

     1
     2
     1
     2
     1
     2
     1
     2
```

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

b) 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:

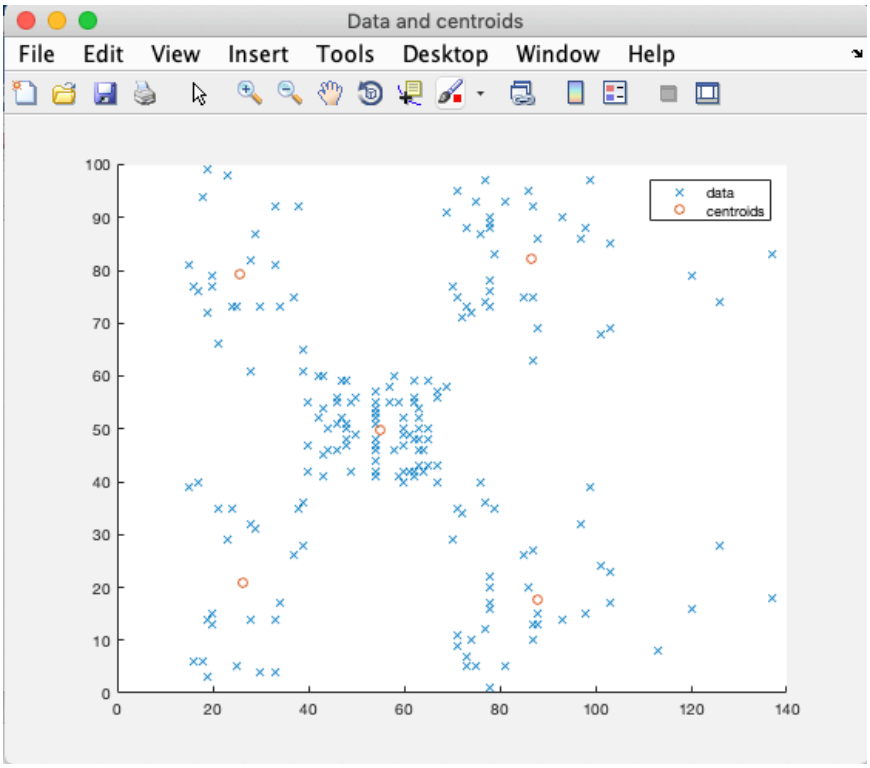
```
The final cost is 2602.9726
```



The final centroids are:

```
Start the K-means algorithm with initial centroids u...
After 3 iterations, the u found by K-means algorithm:
   26.3043   25.7273   55.0875   87.7500   86.5385
   20.9130   79.3636   49.7125   17.5833   82.1282
```

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



The number of iterations is 3.

d) 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.

i. Run the code first time

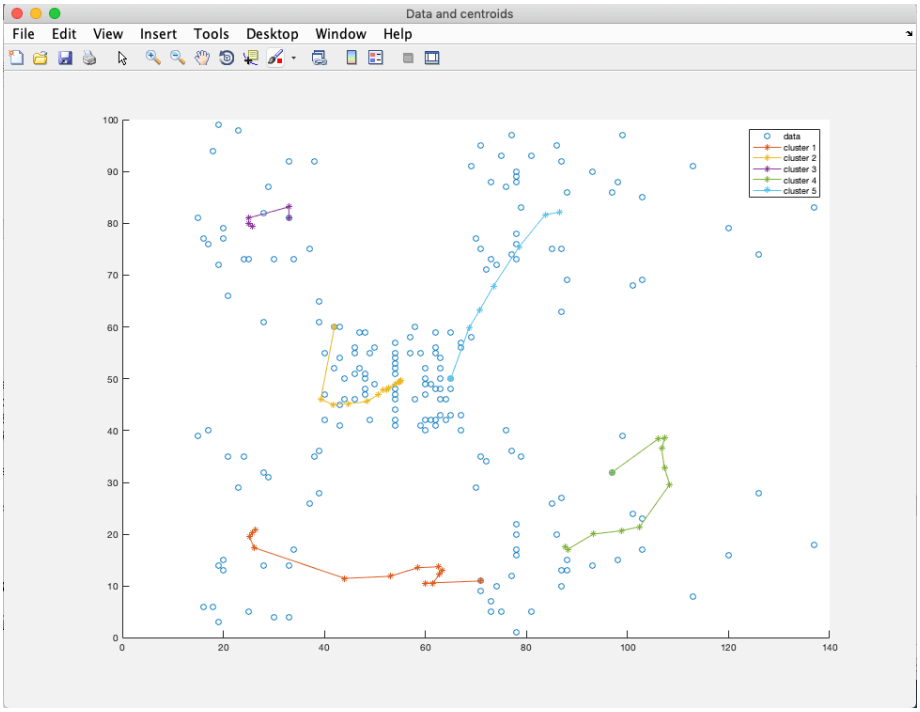
Initial centroids:

```
Initializing...
The random centroids are:
    71    42    33    97    65
    11    60    81    32    50
```

Final centroids:

```
Start the K-means algorithm with initial centroids u...
After 13 iterations, the u found by K-means algorithm:
   26.3043   55.0875   25.7273   87.7500   86.5385
   20.9130   49.7125   79.3636   17.5833   82.1282

The final cost is 2602.9726
```

The plot with centroids with each iteration:



ii. Run the code second time

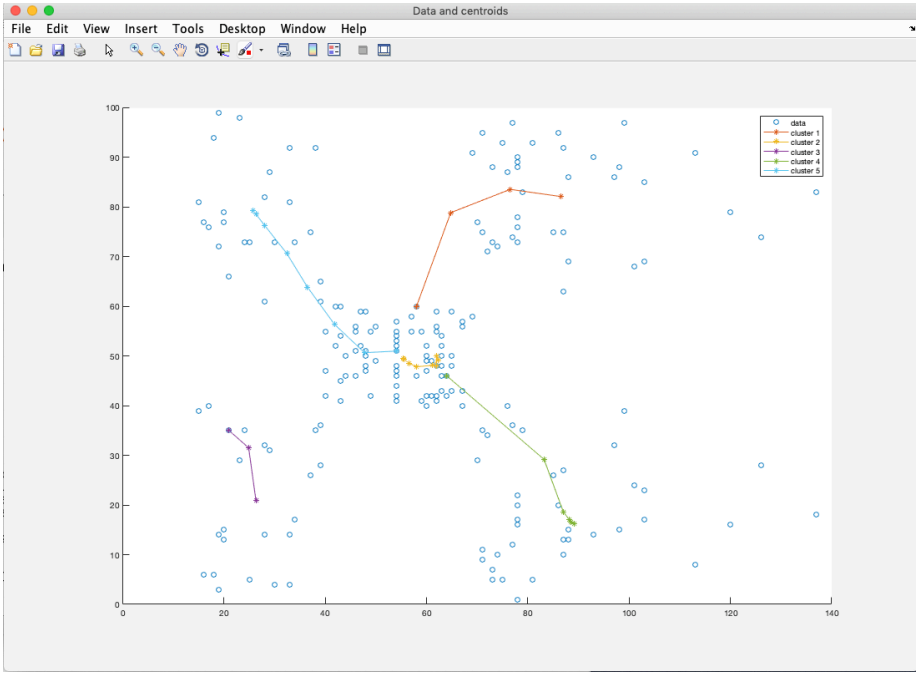
Initial centroids:

```
Initializing...
The random centroids are:
    58    62    21    64    54
    60    48    35    46    51
```

Final centroids:

```
Start the K-means algorithm with initial centroids u...
After 8 iterations, the u found by K-means algorithm:
   86.5385   55.2963   26.3043   88.2000   25.7273
   82.1282   49.5185   20.9130   17.1143   79.3636

The final cost is 2604.0253
```

The plot with centroids with each iteration:



iii. Run the code third time

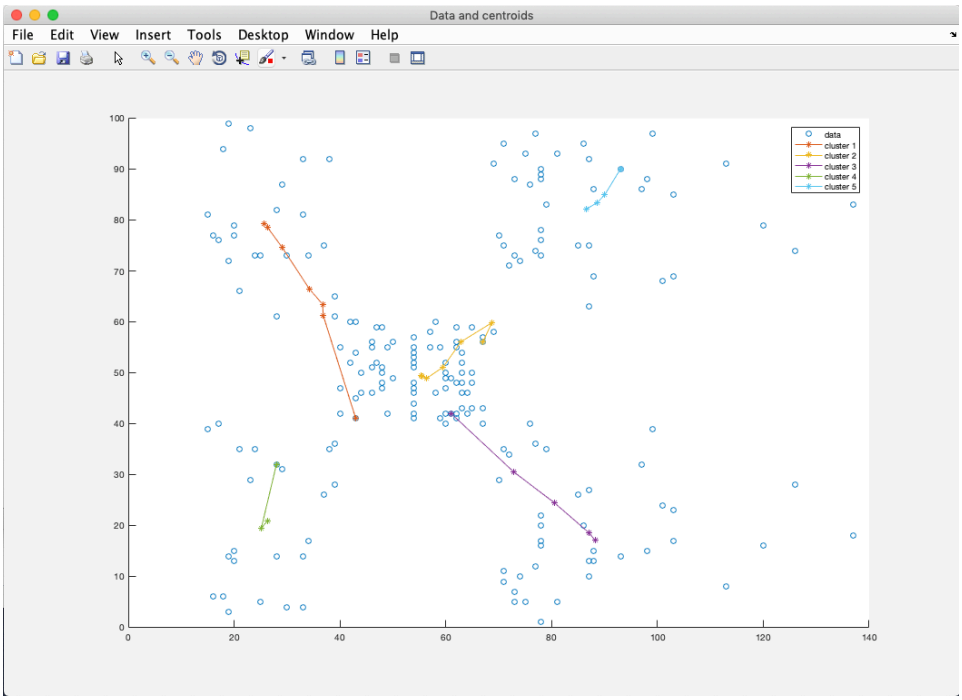
Initial centroids:

```
Initializing...
The random centroids are:
    43    67    61    28    93
    41    56    42    32    90
```

Final centroids:

```
Start the K-means algorithm with initial centroids u...
After 7 iterations, the u found by K-means algorithm:
   25.7273   55.2963   88.2000   26.3043   86.5385
   79.3636   49.5185   17.1143   20.9130   82.1282

The final cost is 2604.0253
```

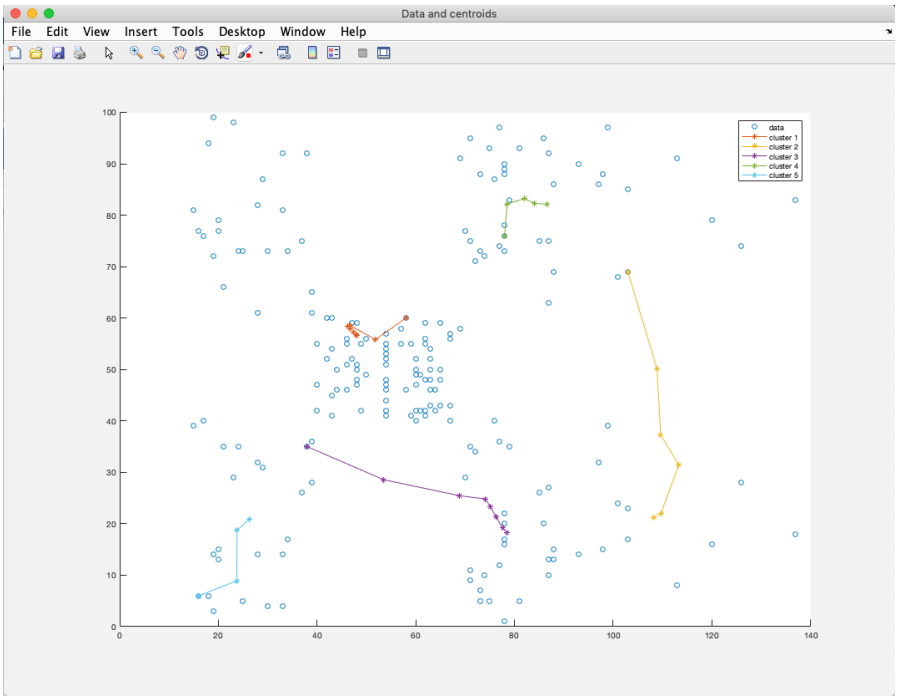
The plot with centroids with each iteration:



iv. Some special cases

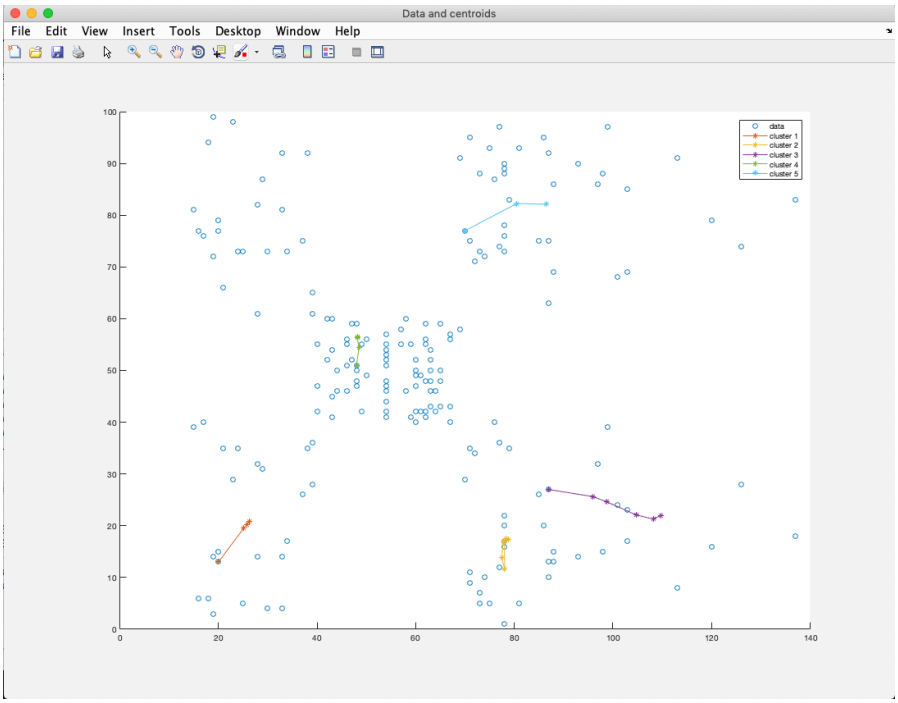
Sometimes I get a case that cost is around 3136.

```
Initializing...
The random centroids are:
    58   103    38    78    16
    60    69    35    76     6

Start the K-means algorithm with initial centroids u...
After 8 iterations, the u found by K-means algorithm:
   48.0707  109.7000   78.4828   86.5385   26.3043
   56.6465   22.0000   18.2069   82.1282   20.9130

The final cost is 3144.7704
```



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.

e) Code

1) 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');
```

2) 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;
```

3) 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
```

4) 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
```

5) 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
```