Taylor Pedretti
005488635

Lab06 - DBScan

```
%main_dbscan.m
% Load data.
X = load('s2.dat');
X = X(:,[1 2]);
% DBSCAN
% initialize parameters ( you can change them to see the difference)
eps = 1.0
minPts = 4
% initialize indices that contains the label/cluster info of every datapoint.
labels = zeros(size(X,1), 1);
% 0 - the datapoint hasn't been touched
% -1 - the datapoint is considered as noise
% any positive integer (e.g 1,2,3,4,...) cluster label
C = 1 % current cluster label starts with 1

% walk through all data points:
[m, n] = size(X);
for i = 1:m

  % check every points, find neighbors, and extend clusters
  % YOUR CODE HERE
  if labels(i) == 0 || labels(i) == -1
     neighbors = findNeighbors(X, i, eps);

     if length(neighbors) < minPts
         labels(i) = -1;
     end

     if length(neighbors) >= minPts
         labels = extendCluster(X, labels, i, neighbors, C, eps, minPts);
         C = C + 1;
     end
  end
  % END OF CODE

end


% Plot
scatter(X(:,1), X(:,2), 32, labels, 'filled')


% findNeighbors.m
function neighbors = findNeighbors(X, p, eps)
    % find all data points in X within eps of the current cendroid point p
    % attention p is a number/ the index of the pth data point
    [m, n] = size(X); % m is the number of data points, n is the number of
features
    neighbors = []; % initialize empty matrix
    for i = 1:m
      if norm(X(p,:)-X(i,:)) < eps
        neighbors = [neighbors; X(i,:)]; % append datapoint to the matrix
      end
```

```
    end
  end


% extendCluster.m
function labels = extendCluster(X, labels, p, neighbors, C, eps, minPts)
    % extend a cluster with label C from the seed p
    % you can define this function recursively. or use a while loop with a
queue.
    labels(p) = C;
    k = 1;

    % YOUR CODE HERE
    while k ~= (length(neighbors) + 1)
        %get next point from the queque
        temp = [1:2];
        temp = neighbors(k,:);
        for j = 1:length(labels)
            if temp == X(j,:)
                Pnext = j;
            end
        end

        if labels(Pnext) == -1
            labels(Pnext) = C;
        end
        if labels(Pnext) == 0
            labels(Pnext) = C;
            neighbors_of_Pnext = findNeighbors(X, Pnext, eps);
            neighbors = vertcat(neighbors, neighbors_of_Pnext);
        end

        k = k + 1;
    end

    % END OF CODE
  end
```