**Lab 05 – K Means**

X = load('s1.dat');

K = 7;

epoch = 20;

centroids = initCentroids(X, K);

for i=1:epoch

indices = reassignPoints(X, centroids);

centroids = updateCentroids(X, indices, K);

end

scatter(X(:,1), X(:,2), 32, indices, 'filled')

%%%%%Functions%%%%%

function centroids = initCentroids(X, K)

centroids = X(randperm(length(X), K), :);

end

function indices = reassignPoints(X, centroids)

indices = zeros(size(X, 1), 1);

for i = 1:size(X, 1)

nearestCluster = 1;

currDataPoint = X(i, :);

centroid1 = centroids(1, :);

min\_dist = sum((currDataPoint - centroid1) .^ 2);

for j = 2:size(centroids, 1)

centroidj = centroids(j, :);

dist = sum((currDataPoint - centroidj) .^ 2);

if(dist < min\_dist) % check dist between current centroid the datapoint is in with another to see if it's closer to the other centroid

min\_dist = dist;

nearestCluster = j;

end

end

indices(i) = nearestCluster;

end

end

function centroids = updateCentroids(X, indices, K)

[m n] = size(X);

centroids = zeros(K, n);

for i=1:K

currDataPoints = X(indices == i, :); %All datapoints within current cluster i

ci = size(currDataPoints, i); %Amount of data points in i^th cluster

centroids(i, :) = (1/ci) \* [sum(currDataPoints(:, 1)) sum(currDataPoints(:, 2))];

end

end

A screenshot of a map

Description automatically generated