5/29/2020

# 

**Machine Learning**

K-Means Clustering Algorithm

sk-means clustering algorithm with matlab

# Table of Contents

[Table of Contents 1](#_Toc41254136)

[1. Aim of the Project 2](#_Toc41254137)

[2. Dataset Description 2](#_Toc41254138)

[3. Explanation of the Algorithm 3](#_Toc41254139)

[4. Implementation of Code 3](#_Toc41254140)

[5. Execution Outcome 5](#_Toc41254141)

[6. Evaluating Results 6](#_Toc41254142)

# Aim of the Project

The aim of this project is to produce a more unique map for the pattern of odor detection with k-means clustering algorithm. With this unique mapping, the identification of the odor will be automated. As a result of this automatic identification, the application of neural neurons will give more reliable answers about the milestone.

More than three perfumes are very difficult for the human nose to be identified. Recognizing the patterns of electronic-nosed perfume scent with multiple sensors is also a problem. Therefore, a new hybrid classifier has been introduced to identify the perfume type from a data set similar to 20 different perfume scents.

# Dataset Description

This dataset created by Prof. Dr. Bekir Karlık in Bahrein University between 2002 and 2003. The dataset obtained from 20 different perfumes by using a handheld odor meter (OMX-GR sensor). Names of these perfumes are:

1. Ajayeb
2. Ajmal
3. Amreaj
4. Aood
5. asgar\_ali
6. Bukhoor
7. Burberry
8. Dehenalaod
9. Junaid
10. Kausar
11. Rose
12. Solidmusk
13. TeaTreeOil
14. Raspberry
15. RoseMusk
16. Strawberry
17. Constrected2
18. Carolina\_herrer
19. Oudh\_ma'alattar
20. Constrected1

Each column represents a measurement and there were 28 takes (one each second).

Dataset Size: 20 x 29

Dataset characteristics: Univariate, Domain-Theory

Number of Instances: 560

Number of Attributes: 2

# Explanation of the Algorithm

K-means algorithm is an iterative algorithm that tries to partition the dataset into Kpre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

Once the value K is determined, the algorithm randomly selects K Center points. By calculating the distance between each data and its randomly determined center points, it assigns the data to a set relative to the nearest center point. A center point is then re-selected for each cluster and clustering is performed according to the new center points. This condition continues until the system becomes stable.

1. The number of K sets is specified.
2. Start the centroids as by first shuffling the dataset and then randomly selecting the k data points for centroids without being changed.
3. It continues to iterate until there are no changes left for Centroids. This means that the assignment of data points to sets does not change.

-The sum of The Square distance between the data points and all centroids is calculated.

-Each data point is assigned to the nearest cluster (centroid).

-Centroids for clusters are calculated by averaging all data points belonging to each cluster.

# Implementation of Code

%Firstly, we open a new lovely white script in our MATLAB.

clc;

clear all;

close all;

for i=1:10

% Set algorithm parameters

TOL = 0.0004;

% The max iteration

ITER = 30;

kappa = 4;

% The dataset that we found on UCI

X = xlsread("perfume\_data.xlsx");

% Called the K-means function

tic;

[C, I, iter] = myKmeans(X, kappa, ITER, TOL);

toc

% Show number of iteration taken by k-means

disp(['k-means instance took ' int2str(iter) ' iterations to complete']);

% We choose colors for the points in the resulting clustering plot

colors = {'red', 'green', 'blue', 'black'};

% Show plot of clustering

figure;

for i=1:kappa

hold on, plot(X((I == i), 1), X(I == i, 2), 'p', 'color', colors{i});

end

title 'Perfume Data';

xlabel 'Odor intensity';

ylabel 'Odor quality';

legend('Cluster 1','Cluster 2','Cluster 3','Data','Location','Best');

% Wait key

pause;

end

% Pause and close all windows in the end.

pause;

close all;

function [C, I, iter] = myKmeans(X, K, maxIter, TOL)

% Number of vectors in X

[vectors\_num, dim] = size(X);

% Compute a random permutation of all input vectors

R = randperm(vectors\_num);

% Construct indicator matrix (each entry corresponds to the cluster

% of each point in X)

I = zeros(vectors\_num, 1);

% Construct centers matrix

C = zeros(K, dim);

% Take the first K points in the random permutation as the center sead

for k=1:K

C(k,:) = X(R(k),:);

end

% iteration count

iter = 0;

% compute new clustering while the cumulative intracluster error in kept

% below the maximum allowed error, or the iterative process has not

% exceeded the maximum number of iterations permitted

while 1

% find closest point

for n=1:vectors\_num

% find closest center to current input point

minIdx = 1;

minVal = norm(X(n,:) - C(minIdx,:), 1);

for j=1:K

dist = norm(C(j,:) - X(n,:), 1);

if dist < minVal

minIdx = j;

minVal = dist;

end

end

% assign point to the cluster center

I(n) = minIdx;

end

% compute centers

for k=1:K

C(k, :) = sum(X(I == k, :));

C(k, :) = C(k, :) / length(find(I == k));

end

% compute RSS error

RSS\_error = 0;

for idx=1:vectors\_num

RSS\_error = RSS\_error + norm(X(idx, :) - C(I(idx),:), 2);

end

RSS\_error = RSS\_error / vectors\_num;

% increment iteration

iter = iter + 1;

% check stopping criteria

if 1/RSS\_error < TOL

break;

end

if iter > maxIter

iter = iter - 1;

break;

end

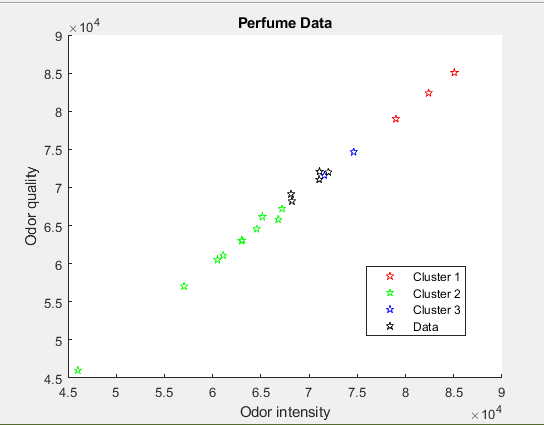
end

end

# Execution Outcome

Elapsed time is 0.006049 seconds.

k-means instance took 1 iterations to complete



# 

# Evaluating Results

The expectations for this project is the results belong to different clusters, and the properties of the data in the cluster which are similar. In this way, it was understood which scent type stimulates which points in the brain according to the odormeter. Since k-means requires k as an input and doesn’t learn it from data, there is no right answer in terms of the number of clusters that we should have in any problem.

When the Dataset was passed through the k-means algorithm, 3 different clusters occurred. These clusters were named as 1,2 and 3. Each cluster has a different pattern. The thing is real life data is almost always complex, disorganized and noisy. Situations in the real world rarely reflect clear conditions in which to apply these type of algorithms right out of the shelf. In the case of K-means algorithm it will be expected that at least one of its assumptions gets violated, so we need not only to identify this, but to know what to do in such case.

The initial result of running this algorithm may not be the best possible outcome and rerunning it with different randomized starting centroids might provide a better performance (different initial objects may produce different clustering results). For this reason, it’s a common practice to run the algorithm multiple times with different starting points and evaluate different initiation methods