BONUS (Grade Classifier) - EE5327

Srujana B - MA17BTECH11001

22/03/2019

Question

Implement a classifier for Grading using Relative Grading

Algorithm used : K - Means Clustering

- Unsupervised Learning Algorithm which is used when you have unlabeled data (i.e., data without defined categories or groups)
- Goal :
  find groups in the data, with the number of groups represented by the variable K.
- works iteratively to assign each data point to one of K groups based on the features that are provided

K - Means Clustering

Pseudo Code :

Initialize k means with random values

For a given number of iterations or Repeat until means donot update:

- Iterate through items:
  - Find the mean closest to the item
  - Assign item to mean
  - Update mean

Algorithm

Step 1 : Choose K centroids/means randomnly
Step 2 : Data Assignment Step
Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance.

$$\underset{c_i \in C}{argmin}\ dist(c_i, d_j)^2$$

$c_i \rightarrow$ centroid

$d_j \rightarrow$ data point

Algorithm

Step 3 : Centroid update step

$$c_i = \frac{1}{N} \sum x_j$$

$x_j \rightarrow$ datapoints assigned to centroid c_i

In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

Step 4 : Termination Step

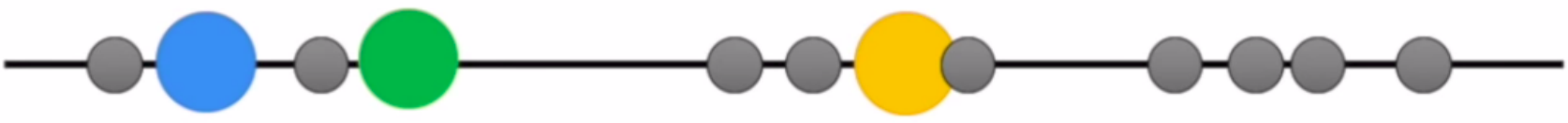
The algorithm iterates between steps one and two until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).

Example

Let K = 3

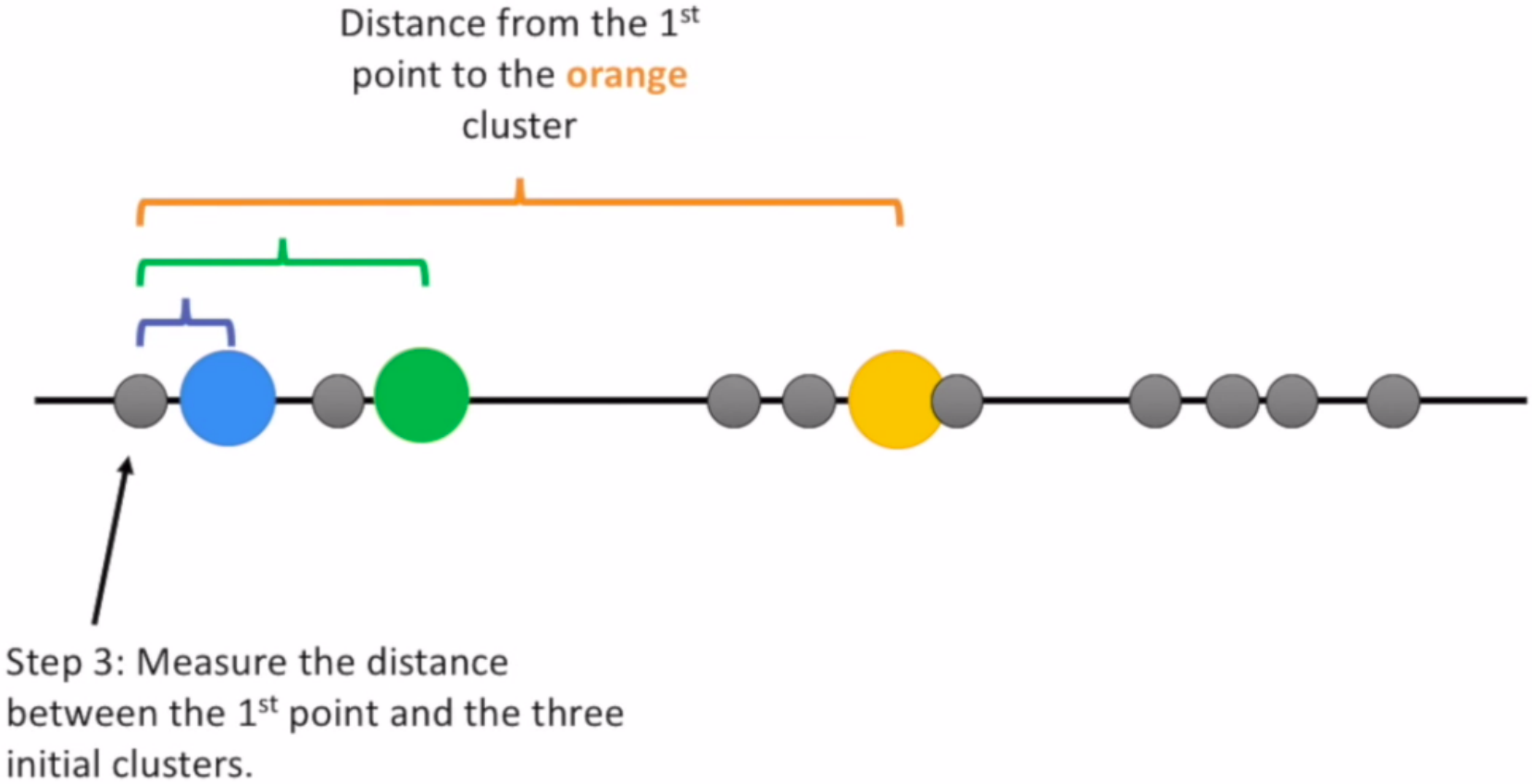
PLot the given data

Randomnly choose K centroids, Each of them represent one cluster



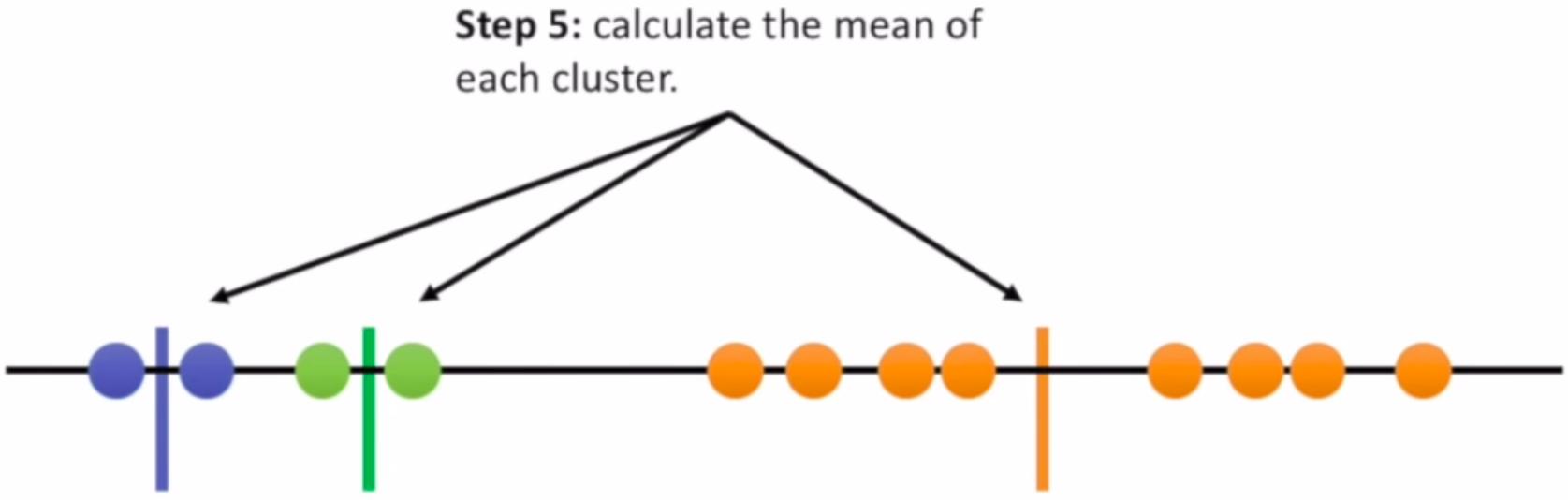
Example

Assign each data point to a cluster using the distance



Example

Calculate the mean, update it and repeat the process



Grade Classifier

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
dataframe x- total marks
df = pd.DataFrame(
'x' :
[17.2,0,0,76.7,66.7,74.2,10,10,95,26.1,15,99.2,99.2,0,0,0,0,96.4,10,
27.7,88.4,86.7,84.2,26.9,91.7,95.7,94.2,96.7,17.2,0,93.2,74.7,0,8,90.1,
92.5,59.2,102,90,92.5,97.5,50.4,25.2,96,86.5,72.6,60.4,63.4],
'y' : [0]*48
)
k = 8 no of grades - FR,D,C-,C,B-,B,A-,A
colmap = 1:'r',2:'g',3:'b',4:'y',5:'k',6:'c',7:'m',8:'b22222'
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = k)
kmeans.fit(df)
```

Cont'd

```
labels = kmeans.predict(df)
centroids = kmeans.cluster_centers
fig = plt.figure(figsize=(5,5))
colors =map(lambda x: colmap[x+1],labels)
colors1 = list(colors)
plt.scatter(df['x'],df['y'],color = colors1,edgecolor='k')
for idx, centroid in enumerate(centroids):
plt.scatter(*centroid, color = colmap[idx+1])
plt.xlim(0,110)
plt.show()
```

Grade Classifier

- Create a data frame with x values as total marks obtained, stored as an array and y values as zeroes
- Value of K = 8
- Output of the code is a graph, which has 8 clusters indicated by a different color
- Each cluster can be assigned highest grade coming from the right most cluster