K-Mean Clustering

Xi Chen
17301 Old Vic Blvd, Olney, MD, US 20832

ABSTRACT

In this paper, I apply K-mean clustering method to test the validity of Bank notes by separating notes in the real or fake group automatically with Python 3 in the Jupyter Lab.

General Terms

K-Mean Algorithms: the sim

Distance Function: a plest tool to separate the data by its certain characteristics.function to model the linear distance between two vectors.

Keywords

K-Mean clustering, center, group, distance, iteration

1. INTRODUCTION

Nowadays, the technology has developed a lot. People can learn new information by simply searching it on website such as Google via their electronical devices. However, since there are several resources online, I am not able to easily distinguish the validity of data or information about whether they are true or not. At this moment, applying the method of separating the data into the "real" and "fake" group is necessary for us. There are various methods for allocating the data, but one of the most important and common methods is to cluster with K-mean Algorithm.

K-mean Algorithm is the simplest tool to separate the data by its certain characteristics. Since the method contains a set of algorithms, I have to define a variable "K", where K is the number of centroids or the centers of each number group. By calculating the distance between individual number and each center, I add the number to a group where its distance to the center of the group is the shortest. The way of finding the centroids is to calculate the mean or average number in each group. For instance, there is a set of numbers 1,2,3,4,5. The mean of these five numbers is 3, which adding all the numbers and then dividing by how many numbers the set contains. About the group, the standard deviation is to show the range of group, but it is not necessary in the k-mean clustering. After calculating the new center, it is important to calculate it several times to get the most accurate value for center. Then, based on the updated center, I will return it to the corresponding group.

The advantages of using K-mean clustering are: 1) if a set of data contains thousands of numbers, it is helpful to separate the data computationally; 2) this method is generalized enough that it can be applied to different shapes and groups of data; 3) it is accurate enough since the computer calculates it several times.

In this paper, I apply K-mean clustering method to test the validity of Bank notes by separating notes in the real or fake group automatically with Python 3 in the Jupyter Lab. Later, I compare them to the actual bank notes for the evaluation of K-mean clustering in this situation.

2. IMPORT RAW DATA

The raw data are the set of data from the image of processing bank notes, and first 762 lists are real bank notes. The code below is to open the cvs format file in Jupyter. To be much simpler, I rename the file to from Bank authentication dataset.cvs to two.csv and then, attach the open file to the V row. Since the first row of the entire data is the name of column, not the number, I then use 'pop' function in python to remove the first row and just keep the number in the dataset.

```
import csv
with open ('two.csv') as f:
    data=csv.reader(f)
    v=[ ]
    for row in data:
        w=[ ]
        for x in row:
            w.append(x)
        v.append(w)
v.pop(0)
for i in range (len(v)):
    v[i]= [float(v[i][0]),float(v[i][0])]
```

3. IMPLEMENTATION

3.1 Function Design &Application

3.1.1 Distance Function between Two Vectors

After importing the data, I now have two columns of data to build up the x and y coordinates in the graph. To cluster and separate the group, the distance from one number to its center is very essential. The standard of dividing into groups is based on the smallest distance to its center, or mean value. The distance is the line segment between each vector, and to find the length of the segment, I have to construct the right triangle with this line segment as the hypotenuse. Therefore, using the Pythagorean Theorem as the graph on the right shows, the length of the segment between vector 1 (x_1, y_1) and vector 2 (x_2, y_2) equals to $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, where a=$x_2 - x_1$, b=$y_2 - y_1$. (See Figure 1 for the deduction of distance function)

The code for this is to first define the function as distance (x, y), and then set it equal to the distance function of vectors $\mathrm{d} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. The code is:

```
import math
def distance(x,y):
    return math.sqrt ((x[0]-y[0])**2+(x[1]-
y[1])**2)
```

3.1.2 Group Function

Since the final goal is to separate those banknotes into real and fake group, it is important to define the group function. Each group g(i) has their own center a(i). The number of group depends on the number of their center. Let x be the list of the vectors (list of two numbers, where the first component is population and the second one is the profit) which I have already created from the banknote dataset. Let a be the list of centers, where the first component is each center of population, the second one is the profit. And thus, let g be the list of output where all the x’s whose closet center is a(i). As the code below shows, k is the total number in the dataset. And for each number in the range of dataset, each x belongs to one group, and its distance to that group center a is the shortest. After running the code, the result is two different groups of data separated based on their centers.

```
def groups(x,a):
    k=len(a)
    g=[ ]
    for s in range(k):
        g.append([ ])
    for n in x:
        imin=0
        dmin=distance(n,a[0])
        i=1
        while i<k:
            d=distance(n,a[i])
            if d<dmin:
                dmin=d
                imin=i
            i=i+1
        g[imin].append(n)
    return g
print (groups)
```

3.1.3 Centers Function

Since I estimate the value of a in the previous section, I have to update the a (i) value by calculating the mean value in one specific group. I define the center value by running the code below. The input for the center function is g, while the output is updated value of center a. Starting from 0, I will add up each number and divide by the total number in that group if the total number of the group is greater than 0. Then, I return to new value of a.

```
def centers(g,a):
    for i in range (len(g)):
        if len(g[i]) >0:
            a[i]=[0,0]
            for q in g[i]:
                a[i]=[a[i][0]
            +q[0], a[i][1]+q[1]]
            a[i]=[a[i][0]/len(g[i]),
                    a[i][1]/len(g[i])]
return a
```

3.1.4 Iteration Function

Now since I have both function for group and center, then I have to perform the previous steps several times again to update the value of a to find the most accurate center and then return to the corresponding group. Therefore, I define the iteration function, where the input is all lists of numbers and the output is the final value of center in each group. The code below shows the iteration function.

```
def iteration (x,a):
    g=groups(x,a)
    a=centers(g,a)
    return a,g
```

3.1.5 K-Mean Function

The final step is to define the k-mean function in order to update the center several times. The code below explains that if old a equals to new one, then return the value to a, g from iteration function. Otherwise, performing previous tasks again until old value of a equals to actual value of a. The input is every list of numbers and the output is the final center.

```
def kmean(x,a):
    aold=a[:]
    a,g =iteration(x,a)
    while aold!=a:
        aold=a[:]
        a,g =iteration(x,a)
    return a,g
```

3.2 Model Calculation

3.2.1 Determine a-value for Each Group

From the illustration, I can first estimate the two center points: (-2, -5) and (2, 5). Then, I use a set of functions above to calculate the accurate value of each center: a,g=kmean(v,a), where a is the center of each group, g is each group number lists, and v is all lists of numbers in the dataset. Finally, the result for each center is approximately (-0.1971, -3.638) and (0.890, 5.982). From the two center points, I can separate numbers into each group where their distance to the center is the shortest.

```
a=[[-2,-5], [2,5]]
a,g =kmean(v,a)
print (a)
```

3.2.2 Real and Fake Group

By plugging all the lists into the functions, I separate those data into two distinct groups. The first group is called g[0] with the first center (-0.1971, -3.638), and the second group is called g[1] with its own center (0.890, 5.982). The illustration below shows the lists of numbers in each group. After getting two centers and two distinct groups based on their centers, I then plug all the number lists into the coordinates. (See Figure 2 for the graph of two distinct groups)

4. FIGURES

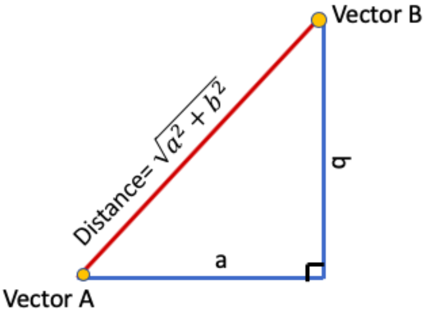
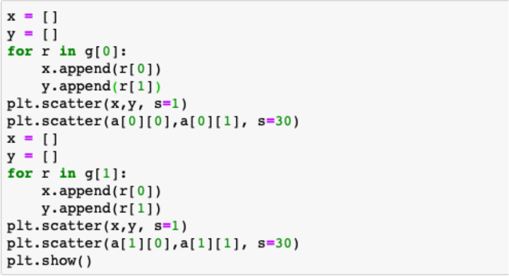


Figure 1: Deduction of Distance Function



```
x = []
y = []
for r in g[0]:
    x.append(r[0])
    y.append(r[1])
plt.scatter(x,y, s=1)
plt.scatter(a[0][0],a[0][1], s=30)
x = []
y = []
for r in g[1]:
    x.append(r[0])
    y.append(r[1])
plt.scatter(x,y, s=1)
plt.scatter(a[1][0],a[1][1], s=30)
plt.show()
```

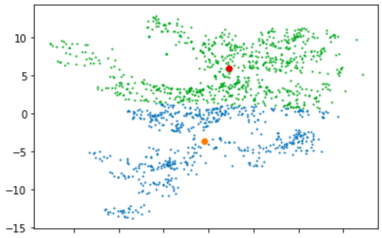


Figure 2: Graph

5. CONCLUSION

5.1 Evaluation

5.1.1 Accuracy & Percent Error

Based on the algorithms, the first group is the fake group, which is centered by (-0.1971, -3.638) and the second group is the real group, where its center is (0.890, 5.982). Since our goal is to use k-mean algorithms to distinguish real banknotes from the fake ones automatically by python, I will then compare the actual real and fake groups to our calculated groups. For the actual real and fake groups, the first 762 banknotes are real, and the rest of them are fake. Therefore, I will compare and evaluate both graphically and logically. For every list i in the first 762 lists, they are real banknotes. Thus, I put the first number of each list in the x-coordinate, and the second number of the list in the y-coordinate. As a result, demonstrated in the graph, the blue group is the actual real group. Similarly, the orange group is fake group. To compare our groups to the actual real and fake groups of banknotes, I compare every list of numbers separately. If the number list belongs to real group g[1] but not the real banknote, then this is counted as one error. Therefore, based on this principle, I create the code on the right. The result is 475 errors total, and the error percentage is approximately 34.62%. The percent error is lower than 50%, meaning that the method is acceptable but needs some improvements. Based on the graph, there is some overlapping pace in the middle of the graph, and k-mean algorithms uses arithmetic mean to calculate the distance between each point and its centroid. Hence, it causes approximately 34 percent of error.

```
error=0
for x in g[1]:
        if x not in real:
                error+=1
for x in g[0]:
        if x not in fake:
                error +=1
print(error)
len(v)
print(error/len(v))
```

5.1.2 Strengths

1) Efficient: our method is productive because by running the code in the computer, it only takes some minutes to finish calculating thousands of data. Even though writing the code takes some time, after finishing design the algorithms, it is easy to write the similar code later in the future by only changing the dataset. The process is always the same.

2) Easy to understand: K-mean clustering method is easy to understand because it only requires some algebraic skills and then everyone can perform it. The basis of k-mean clustering is to calculate the algebraic distance between every number and its center. And if the number has the shortest distance to one center, then that center determines which group this number belongs to. Therefore, k-mean clustering does not require any advanced math skill.

3) Applicable: K-mean clustering not only can be used in distinguish fake group from real one for banknote, but also it can be applied to a large range of area such as public transportation data analysis and insurance fraud detection. The method and the code are always the same and the only difference is the dataset and its background information.

5.1.3 Weaknesses

1) Sensitive to initial condition: The result will be local optimum instead of global optimum due to the estimation of initial value of center. If the initial value is too large or too small, it will affect the accurate result of final value of center produced by the algorithms. Different centroids lead to different cluster groups.

2) Circular cluster: Due to the linear distance to centroids, the shape of group will usually be circular and it is difficult to calculate the overlapping area of two groups. And it will cause the centroid to be away from the actual center of the group.

3) Only for a large range of data: If there are only few numbers, the initial estimated centroid will influence the result significantly, and thus leading to the different cluster of groups. Therefore, k-mean clustering only works for a large range of data.

5.1.4 Improvement

The way to minimize the weakness is to test different values of centroid for initial condition. Based on the results from different values of centroids, I can choose the most trustworthy one to do the k-mean clustering analysis. Another way to help determine the initial condition of center point is to make a graph for the better view of center. Usually, from the graph, it is easier to determine the number of the groups and predict the centroid for each group.

5.2 Result Interpretation

Even though there are a few weaknesses, k-mean clustering is still generalized and can be used in several areas. It uses distance function to determine the center of each group in order to separate the data automatically. To distinguish the fake banknote from real one, I found two centers: (-0.1971, -3.638) is the center of the fake banknote group and (0.890, 5.982) is the center of the real group. By comparing it to the actual two groups, the percent error is still acceptable due to the overlapping area for both groups. K-mean clustering method can also be applied to distinguish any fraud from the real one in our life and it is efficient and simplified enough. For the better application in the future, I suggest test different values for the initial centroids to minimize the weaknesses created by the clustering method.

6. ACKNOWLEDGMENTS

My special thanks to:

Professor Guillermo Goldsztein, a professor in Georgia Institute of Technology, for academic instruction in the area of data science and mentorship in K-mean clustering model and functions.

Mrs. Laura Elizabeth Squirlock, a math teacher in Our Lady of Good Counsel High School, for long-term academic instruction in the area of mathematics and providing the publication methods and resources.

7. REFERENCES

[1] Khan, Muhammad Rizwan. “K Means Clustering Algorithm & Its Application.” Medium, Data Driven Investor, 12 Oct. 2018, https://medium.com/datadriveninvestor/k-means-clustering-algorithm-its-application-ff9e97297e6e.

[2] “k-Means Advantages and Disadvantages | Clustering in Machine Learning.” Google, Google, https://developers.google.com/machinelearning/clustering/algorithm/advantages-disadvantages.