Kmean

September 21, 2018

1 Kmeans Clustering

16BCE1259
Shushil Kumar Ravishankar

```
In [2]: from pprint import pprint
```

```
In [7]: def vectorize(doc,terms):
            docvector=[]
            count=0
            for i in range(len(terms)):
                docvector.append(0)
                for syn in terms[i]:
                    docvector[i]+=doc.lower().split(" ",500).count(syn.lower())
            return docvector
        def vectorAvg(a):
            center=[0 for i in range(len(a[0]))]
            for vec in a:
                center=list(map(sum,zip(center,vec)))
            n=len(a)
            center[:]=[round(x/n,3) for x in center]
            return center
        def distInitknn(mat,n,k):
            for i in range(n):
                mat.append([0 for j in range(k)])
            return None
        def kMean(docvectors,centroids,k=2,prev=[],n=1):
            print("\niteration: ",n,"\n")
            cluster=[]
            distMat=[]
            print("\ncentroid vectors:\n")
            pprint(centroids)
            print("\ndoc vectors:\n")
            pprint(docvectors)

            veclen=len(docvectors)
            distInitknn(distMat,veclen,k)
            genDistMat(distMat,docvectors,centroids)
```

```
    for i in range(k):
        cluster.append([i])
    if prev==[]:
        for i in range(k,len(docvectors)):
            cluster[i%k].append(i)
    else:
        for i in range(k,veclen):
            cluster[getMinIndex(i,distMat)].append(i)
    newcentroid=[]
    print("cluster formed:\n",cluster)

    for clus in cluster:
        vec=[docvectors[i] for i in clus]
        center=vectorAvg(vec)
        newcentroid.append(list(center))
    print("\nnew centroids:\n")
    pprint(newcentroid)
    print("*"*50)
    if cluster!=prev:
        kMean(docvectors,newcentroid,k,cluster,n+1)
    return None
def calcManDist(a,b):
    dist=0
    for x,y in zip(a,b):
        dist+=abs(x-y)
    return dist
def genDistMat(distMat,docvectors,centroids):
    for i in range(len(docvectors)):
        for j in range(len(centroids)):
            dist=calcManDist(docvectors[i],centroids[j])
            distMat[i][j]=dist
    return None
def getMinIndex(i,mat):
    mindist=float("inf")
    minindex=0
    for j in range(len(mat[i])):
        if mindist>mat[i][j]:
            mindist=mat[i][j]
            minindex=j
    return minindex
```

In [8]:

```
terms=[['automotive'],['car','cars'],['motorcycles','motorcycle'],['self-drive'],['IoT

doc1='Electric automotive maker Tesla Inc. is likely to introduce its products in Indi
doc2='Automotive major Mahindra likely to introduce driverless cars'
doc3='BMW plans to introduce its own motorcycles in india'
doc4='Just drive, a self-drive car rental firm uses smart vehicle technology based on
doc5='Automotive industry going to hire thousands in 2018'
```

```
doc6='Famous cricket player  Dhoni brought his priced car Hummer which is an SUV'
doc7='Dhoni led india to its second world cup victory'
doc8='IoT in cars will lead to more safety and make driverless vehicle revolution poss
doc9='Sachin recommended Dhoni for the indian skipper post'

docvectors=[]
doclist=[doc1,doc2,doc3,doc4,doc5,doc6,doc7,doc8,doc9]
for doc in doclist:
    docvectors.append(vectorize(doc,terms))
k=4
kMean(docvectors,docvectors[:k],k)
```

```
iteration:  1


centroid vectors:

[[1, 0, 0, 0, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 0],
 [0, 0, 1, 0, 0, 0, 0],
 [0, 1, 0, 1, 1, 0, 0]]

doc vectors:

[[1, 0, 0, 0, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 0],
 [0, 0, 1, 0, 0, 0, 0],
 [0, 1, 0, 1, 1, 0, 0],
 [1, 0, 0, 0, 0, 1, 0],
 [0, 1, 0, 0, 0, 0, 1],
 [0, 0, 0, 0, 0, 0, 1],
 [0, 1, 0, 0, 1, 0, 0],
 [0, 0, 0, 0, 0, 0, 1]]
cluster formed:
 [[0, 4, 8], [1, 5], [2, 6], [3, 7]]

new centroids:

[[0.667, 0.0, 0.0, 0.0, 0.0, 0.333, 0.333],
 [0.5, 1.0, 0.0, 0.0, 0.0, 0.0, 0.5],
 [0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5],
 [0.0, 1.0, 0.0, 0.5, 1.0, 0.0, 0.0]]
**************************************************

iteration:  2
```

```
centroid vectors:

[[0.667, 0.0, 0.0, 0.0, 0.0, 0.333, 0.333],
 [0.5, 1.0, 0.0, 0.0, 0.0, 0.0, 0.5],
 [0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5],
 [0.0, 1.0, 0.0, 0.5, 1.0, 0.0, 0.0]]

doc vectors:

[[1, 0, 0, 0, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 0],
 [0, 0, 1, 0, 0, 0, 0],
 [0, 1, 0, 1, 1, 0, 0],
 [1, 0, 0, 0, 0, 1, 0],
 [0, 1, 0, 0, 0, 0, 1],
 [0, 0, 0, 0, 0, 0, 1],
 [0, 1, 0, 0, 1, 0, 0],
 [0, 0, 0, 0, 0, 0, 1]]
cluster formed:
 [[0, 4], [1, 5], [2, 6, 8], [3, 7]]

new centroids:

[[1.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0],
 [0.5, 1.0, 0.0, 0.0, 0.0, 0.0, 0.5],
 [0.0, 0.0, 0.333, 0.0, 0.0, 0.0, 0.667],
 [0.0, 1.0, 0.0, 0.5, 1.0, 0.0, 0.0]]
**************************************************

iteration:  3


centroid vectors:

[[1.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0],
 [0.5, 1.0, 0.0, 0.0, 0.0, 0.0, 0.5],
 [0.0, 0.0, 0.333, 0.0, 0.0, 0.0, 0.667],
 [0.0, 1.0, 0.0, 0.5, 1.0, 0.0, 0.0]]

doc vectors:

[[1, 0, 0, 0, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 0],
 [0, 0, 1, 0, 0, 0, 0],
 [0, 1, 0, 1, 1, 0, 0],
 [1, 0, 0, 0, 0, 1, 0],
 [0, 1, 0, 0, 0, 0, 1],
 [0, 0, 0, 0, 0, 0, 1],
 [0, 1, 0, 0, 1, 0, 0],
 [0, 0, 0, 0, 0, 0, 1]]
cluster formed:
 [[0, 4], [1, 5], [2, 6, 8], [3, 7]]

new centroids:

[[1.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0],
 [0.5, 1.0, 0.0, 0.0, 0.0, 0.0, 0.5],
 [0.0, 0.0, 0.333, 0.0, 0.0, 0.0, 0.667],
 [0.0, 1.0, 0.0, 0.5, 1.0, 0.0, 0.0]]
**************************************************
```

```
In [9]: files = ['doc1.txt', 'doc2.txt', 'doc3.txt', 'doc4.txt','doc5.txt', 'doc6.txt', 'doc7.t
        terms=[['tesla',"tesla's"], ['electric'], ['car','cars','vehicle','vehicles','automobi
        docvectors=[]
        for fname in files:
            file=open(fname,'r')
            doclines=file.read().split('.')
            doc=''
            for line in doclines:
                doc+=" "+str(line)
            docvectors.append(vectorize(doc,terms))
        kMean(docvectors,docvectors[:k],k)
```

```
iteration:  1


centroid vectors:

[[5, 4, 3, 0, 0, 0, 0],
 [0, 22, 13, 0, 0, 0, 0],
 [0, 1, 7, 0, 0, 0, 0],
 [0, 10, 9, 0, 0, 0, 0]]

doc vectors:

[[5, 4, 3, 0, 0, 0, 0],
 [0, 22, 13, 0, 0, 0, 0],
 [0, 1, 7, 0, 0, 0, 0],
 [0, 10, 9, 0, 0, 0, 0],
 [0, 5, 2, 0, 2, 0, 1],
 [0, 0, 1, 0, 8, 0, 1],
 [0, 0, 4, 0, 0, 14, 0],
 [0, 8, 14, 1, 0, 1, 0],
 [0, 0, 0, 0, 8, 0, 1],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 7, 0],
 [0, 0, 0, 0, 0, 3, 0]]
cluster formed:
 [[0, 4, 8], [1, 5, 9], [2, 6, 10], [3, 7, 11]]

new centroids:

[[1.667, 3.0, 1.667, 0.0, 3.333, 0.0, 0.667],
 [0.0, 7.333, 4.667, 0.0, 2.667, 0.0, 0.333],
 [0.0, 0.333, 3.667, 0.0, 0.0, 7.0, 0.0],
 [0.0, 6.0, 7.667, 0.333, 0.0, 1.333, 0.0]]
*************************************************

iteration:  2


centroid vectors:

[[1.667, 3.0, 1.667, 0.0, 3.333, 0.0, 0.667],
 [0.0, 7.333, 4.667, 0.0, 2.667, 0.0, 0.333],
 [0.0, 0.333, 3.667, 0.0, 0.0, 7.0, 0.0],
 [0.0, 6.0, 7.667, 0.333, 0.0, 1.333, 0.0]]

doc vectors:

[[5, 4, 3, 0, 0, 0, 0],
 [0, 22, 13, 0, 0, 0, 0],
 [0, 1, 7, 0, 0, 0, 0],
 [0, 10, 9, 0, 0, 0, 0],
 [0, 5, 2, 0, 2, 0, 1],
 [0, 0, 1, 0, 8, 0, 1],
 [0, 0, 4, 0, 0, 14, 0],
 [0, 8, 14, 1, 0, 1, 0],
 [0, 0, 0, 0, 8, 0, 1],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 7, 0],
 [0, 0, 0, 0, 0, 3, 0]]
cluster formed:
 [[0, 4, 5, 8, 9], [1], [2, 6, 10, 11], [3, 7]]

new centroids:

[[1.0, 1.8, 1.2, 0.0, 3.6, 0.0, 0.6],
 [0.0, 22.0, 13.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.25, 2.75, 0.0, 0.0, 6.0, 0.0],
 [0.0, 9.0, 11.5, 0.5, 0.0, 0.5, 0.0]]
*************************************************
```

```
iteration:  3


centroid vectors:

[[1.0, 1.8, 1.2, 0.0, 3.6, 0.0, 0.6],
 [0.0, 22.0, 13.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.25, 2.75, 0.0, 0.0, 6.0, 0.0],
 [0.0, 9.0, 11.5, 0.5, 0.0, 0.5, 0.0]]

doc vectors:

[[5, 4, 3, 0, 0, 0, 0],
 [0, 22, 13, 0, 0, 0, 0],
 [0, 1, 7, 0, 0, 0, 0],
 [0, 10, 9, 0, 0, 0, 0],
 [0, 5, 2, 0, 2, 0, 1],
 [0, 0, 1, 0, 8, 0, 1],
 [0, 0, 4, 0, 0, 14, 0],
 [0, 8, 14, 1, 0, 1, 0],
 [0, 0, 0, 0, 8, 0, 1],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 7, 0],
 [0, 0, 0, 0, 0, 3, 0]]
cluster formed:
 [[0, 4, 5, 8, 9], [1], [2, 6, 10, 11], [3, 7]]

new centroids:

[[1.0, 1.8, 1.2, 0.0, 3.6, 0.0, 0.6],
 [0.0, 22.0, 13.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.25, 2.75, 0.0, 0.0, 6.0, 0.0],
 [0.0, 9.0, 11.5, 0.5, 0.0, 0.5, 0.0]]
**************************************************
```