OLD DOMINION UNIVERSITY

Assignment 9

David Bayard

April 30, 2019

QUESTION 1.

Use knnestimate() to compute the nearest neighbors for both: http://f-measure.blogspot.com/ http://ws-dl.blogspot.com/

Solution:

In order to compute the nearest neighbors for both URIs, the cosine distance metric was implemented. This was done by using the scipy library, which offers a method to calculate cosine distance as shown below.

```
def cosSimilarity(v1,v2):

  return (spatial.distance.cosine(v1, v2))
```

Listing 1: cosine distance metric

This method takes two 1 dimensional arrays and calculates the distance between them as shown below.

$$1 - \frac{u \cdot v}{||u||_2 \, ||v||_2}$$

where $||*||_2$ *is the* 2 *norm of its argument* $*$ *and* $u \cdot v$ *is the cross product of* u *and* v

In order for this method to work, two 1 dimensional arrays are required. This is done by extracting the data from the blogdata.txt file introduced in assignment 7. This file contains the frequency of 1000 words in each row, as well as the blog title.

To create the 1 dimensional arrays, each row must be read from the blogdata.txt file, and the blog titles need to be removed. The readfile function from assignment 7 is used to do this, returning a list of 1 dimensional lists, representing each blog.

```

data=[]
  for line in lines[1:]:
    p=line.strip( ).split('\t')
  # First column in each row is the rowname
    rownames.append(p[0])
  # The data for this row is the remainder of the row
    data.append([float(x) for x in p[1:]])
```

Listing 2: Extract numbers from blogdata.txt file

Using the 1 dimensional list of lists, the knn estimate function may be called to find the nearest k neighbors of a specific blog, represented as a matrix. By providing this function the indexes 0 and 1 of the returned list, representing F-Measuree and Web Science and Digital Libraries Research Group respectively, the distance of k elements from those blogs is estimated.

```
def knnestimate(data,vec1,k=5):
  # Get sorted distances
  newList = []
  dlist=getdistances(data,vec1)

  for i in range(0,k):
    newList.append(dlist[i])

```

```
  return newList
```

Listing 3: knn measure function

The function above returns the nearest k neighbors of the provided matrix, against all other matrices in the blogdata.txt file. This is done by calling the getdistances function, which uses the cosSimalirity function to find the cosine distance between a pair of matrices.

```
for i in range(len(data)):
    vec2=data[i]

    # Add the distance and the index
    distancelist.append((cosSimilarity(vec1,vec2),i))

  # Sort by distance
  distancelist.sort()
  return distancelist
```

Listing 4: get distances function

This function returns a sorted list of distances between every matrix in the data file and the matrix to compare. The code below depicts how to generate the nearest k neighbors of a specific matrix.

```
# Distance of k neighbors from f-measure blog
f_measureDist = helper.knnestimate(data, data[0],5)

# Distance of k neighbors from web science blog
web_ScienceDist = helper.knnestimate(data,data[1],5)
```

Listing 5: get distances of F-Measure and Web Science blogs

The tables below depict the nearest neighbors for both the f-measure and web science blogs, for k = 5

F-Measure Blog		
Cosine Distance	Index	Blog Title
0.0	0	F-Measure
0.6247023086160843	35	Indie Obsessive
0.7324998119138642	7	2 or 3 lines
0.8257794856910945	24	Dave's Music Database
0.8263925840186517	8	This Is Country Music

Web Science Blog		
Cosine Distance	Index	Blog Title
0.0	1	Web Science and Digital Libraries Research Group
0.40134546076893296	73	BishopBlog
0.4524798903145403	42	Data Science Notes
0.454005920449409	51	Big Data Society
0.48310622025301797	48	The Tree of Life

Looking at the tables above, it is evident that the nearest neighbors are related to the blog being compares. This is due to the fact that each of the blogs closest to the one being compared maintain the same theme, either being music or data science. The data for k values of 1,2,5,10, and 20 are stored in the distanceFile.txt file.