Assignment 9

CS 532: Introduction to Web Science

Spring 2018

Hrishi Gadkari

1

Question

1. Using the data from A7:

- Consider each row in the blog-term matrix as a 1000 dimension vector, corresponding to a blog.

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

for k={1,2,5,10,20}.

Use cosine distance metric (chapter 8) not euclidean distance.
So you have to implement numpredict.cosine() instead of using
numpredict.euclidean() in:
https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter8/numpredict.py

Answer

Since I had not done Assignment 7, I decided to use **blogdata.txt** from one my classmatesv[1] github account. I then modified the **numpredict.py** as memtioned in the question [2] for the functions knnestimate, getdistances and cosine. In knnestimate I removed the average value whereas I implemented cosine similarity using scipy library [3]. For the getdistance, I removed the Euclidean distance function and used cosine function instead. The changes looks as follows:

```
1
2  def euclidean(v1,v2):
3    d=0.0
4    for i in range(len(v1)):
5      d+=(v1[i]-v2[i])**2
6    return math.sqrt(d)
7
8  def cosine(v1,v2):
9
10   result = 1 - spatial.distance.cosine(v1, v2)
11   return result
12
13
14
15
16
17 def getdistances(data,vec1):
18   distancelist=[]
19
20   #import pdb
21   #pdb.set_trace()
22
23   # Loop over every item in the dataset
24   for i, cal in enumerate(data):
25     vec2 = cal
26
27     # Add the distance and the index
28
29     distancelist.append((cosine(vec1,vec2),i))
30
31
32   # Sort by distance
33   distancelist.sort(reverse=True)
34   return distancelist
35
36 def knnestimate(data,vec1,k=5):
37   # Get sorted distances
38   distant=getdistances(data,vec1)
```

```
39
40
41    return distant
```

Listing 1: Python program for modifies numpredict.py

In-order to get the vector data from **blogdata.txt**, I fetched the blogdata in tuples and added to specified blog arrays and created new dictionary with blog as key. I have placed the program in **knncalc.py**. The vectors were then used for the cosine measurement by running through the F-measure and WSDL group blogs.

```
1  from numpredict import *
2
3
4  def calculateData():
5
6      fmeasure = 'F-Measure'
7      wlblog = 'Web Science and Digital Libraries Research Group'
8      bnames = {}
9      mesf = []
10     cesf = []
11     with open("blogdata.txt", 'r', encoding='utf-8') as f:
12         doctext = f.readlines()
13         for i, line in enumerate(doctext):
14             if i == 0:
15                 # skip header
16                 continue
17             tuples = line.strip().split('\t')
18             if tuples[0] == fmeasure:
19                 for i in range(1, len(tuples)):
20                     mesf.append(float(tuples[i]))
21             elif tuples[0] == wlblog:
22                 for i in range(1, len(tuples)):
23                     cesf.append(float(tuples[i]))
24             else:
25                 bnames[tuples[0]] = []
26                 for i in range(1, len(tuples)):
27                     bnames[tuples[0]].append(float(tuples[i]))
28
29     return bnames, mesf, cesf
30
31
32 def knnest(calval, mesvec, gpvec):
33     nn = knnestimate(bnames.values(), mesvec)
34     print("=" * 2)
35     print("F-Measure")
36     print("=" * 2)
37     kvals = [1, 2, 5, 10, 20]
```

```
    for k in kvals:
        print('k =', k)
        for j in range(k):
            print('%s\t%.6f' % (list(bnames.keys())[nn[j][1]],
                nn[j][0]))

        print("-----------------------------------------" * 2)
    print()

    print("=========================================" * 2)
    print("Web Science and Digital Libraries Research Group")
    print("=========================================" * 2)
    nn = knnestimate(bnames.values(), gpvec)
    for k in kvals:
        print('k =', k)
        for j in range(k):
            print('%s\t%.6f' % (list(bnames.keys())[nn[j][1]],
                nn[j][0]))

        print("-----------------------------------------" * 2)



if __name__ == "__main__":
    bnames, dvec, worvec = calculateData()
    knnest(bnames.values(), dvec, worvec)
```

Listing 2: Python program for KNN

The results are stored in **output.txt**

```
(python35) C:\Users\GADKARI\Documents\Python Scripts\web_science
    \a9>python knncalc.py
==========================================================
F-Measure
==========================================================
k = 1
DaveCromwell Writes      0.546405
----------------------------------------------------------
k = 2
DaveCromwell Writes      0.546405
The Jeopardy of Contentment      0.510352
----------------------------------------------------------
k = 5
DaveCromwell Writes      0.546405
The Jeopardy of Contentment      0.510352
Pithy Title Here         0.502027
KiDCHAIR         0.500508
Alex Denney      0.484831
----------------------------------------------------------
k = 10
DaveCromwell Writes      0.546405
The Jeopardy of Contentment      0.510352
Pithy Title Here         0.502027
KiDCHAIR         0.500508
Alex Denney      0.484831
Skiptrack music 0.482491
New Amusements  0.469968
My Name Is Blue Canary  0.464781
The Slow Music Movement 0.462693
.        0.459632
----------------------------------------------------------
k = 20
DaveCromwell Writes      0.546405
The Jeopardy of Contentment      0.510352
Pithy Title Here         0.502027
KiDCHAIR         0.500508
Alex Denney      0.484831
Skiptrack music 0.482491
New Amusements  0.469968
My Name Is Blue Canary  0.464781
The Slow Music Movement 0.462693
.        0.459632
Diagnosis: No Radio      0.453053
The Power of Independent Trucking        0.451962
Did Not Chart    0.451597
Captain Panda's Local & Independent Music Showcase       0.451494
PSI LAB 0.448001
```

```
unter diesem gesichtspunkt       0.421705
Lyrically Speaking       0.412260
macthemost       0.407770
Notes from a Genius      0.403266
She May Be Naked         0.397429
-------------------------------------------------

=================================================
Web Science and Digital Libraries Research Group
=================================================
k = 1
unter diesem gesichtspunkt       0.393208
-------------------------------------------------
k = 2
unter diesem gesichtspunkt       0.393208
Myopiamuse       0.390856
-------------------------------------------------
k = 5
unter diesem gesichtspunkt       0.393208
Myopiamuse       0.390856
She May Be Naked         0.379675
Alex Denney      0.367140
DaveCromwell Writes      0.350813
-------------------------------------------------
k = 10
unter diesem gesichtspunkt       0.393208
Myopiamuse       0.390856
She May Be Naked         0.379675
Alex Denney      0.367140
DaveCromwell Writes      0.350813
Pithy Title Here         0.348897
The Power of Independent Trucking        0.347941
The Professional Daydreamer      0.344014
Diagnosis: No Radio      0.333850
My Name Is Blue Canary   0.325889
-------------------------------------------------
k = 20
unter diesem gesichtspunkt       0.393208
Myopiamuse       0.390856
She May Be Naked         0.379675
Alex Denney      0.367140
DaveCromwell Writes      0.350813
Pithy Title Here         0.348897
The Power of Independent Trucking        0.347941
The Professional Daydreamer      0.344014
Diagnosis: No Radio      0.333850
My Name Is Blue Canary   0.325889
Mile In Mine     0.319843
Morgan's Blog    0.318304
```

```
96  Media Coursework          0.313015
97  .         0.311779
98  macthemost        0.310349
99  New Amusements    0.306653
100 ELLIA TOWNSEND A2         0.301376
101 hello my name is justin.          0.297976
102 PSI LAB 0.295423
103 Nothing But Ordinary Glances At Extraordinary Things      0.295058
104 ----------------------------------------------------------------------
```

Listing 3: Python program for KNN

2

Question

```
==================================
===The questions below is for 3 points extra credit===
==================================

3. Re-download the 1000 TimeMaps from A2, Q2.  Create a graph where
the x-axis represents the 1000 TimeMaps.  If a TimeMap has "shrunk",
it will have a negative value below the x-axis corresponding to the
size difference between the two TimeMaps.  If it has stayed the
same, it will have a "0" value.  If it has grown, the value will be
positive and correspond to the increase in size between the two
TimeMaps.

As always, upload all the TimeMap data.  If the A2 github has the
original TimeMaps, then you can just point to where they are in
the report.
```

Answer

For the above problem I noticed that I didnt have a csv file with number of Mementos count for each URI in Assignment 2. I then wrote a program in python placed in **check.py** to count the number of mementos by parsing those json files of each URIs for each Timemap from Assignment 2 [4]. I then stored the counts individually in **previousMC.csv** file.

```
1  import json
2  from pprint import pprint
3  import csv
4  #with open('timemaps/1.json', 'w+') as outfile:
5  with open('previousMC.csv', 'a+', newline='') as csvfile:
6          fieldnames = ['Memento_Count']
7          writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
8          writer.writeheader()
9          csvfile.close()
10
11 for i in range(1,1001):
12         try:
13                 #loading json data if file is with .json
                       extension
14                 json_file = 'timemaps/' + str(i) +'.json'
15                 json_data=open(json_file)
```

```
16                  data = json.load(json_data)
17                  #accesing the number of mementos by counting
                        number of keys in list
18                  j = data['mementos']['list']
19                  #pprint(len(j))
20                  json_data.close()
21                  mementocount=len(j)
22                  json_data.close()
23                  with open('previousMC.csv', 'a+', newline='') as
                        csvfile:
24                      fieldnames = ['Memento_Count']
25                      writer = csv.DictWriter(csvfile,
                            fieldnames=fieldnames)
26                      writer.writerow({'Memento_Count':
                            mementocount})
27                      csvfile.close()
28          except:
29                  mementocount=0
30                  # memento count is zero if file extension is txt
31                  with open('previousMC.csv', 'a+', newline='') as
                        csvfile:
32                      fieldnames = ['Memento_Count']
33                      writer = csv.DictWriter(csvfile,
                            fieldnames=fieldnames)
34                      writer.writerow({'Memento_Count':
                            mementocount})
35                      csvfile.close()
36                  pass
37
38          #j = json.loads(outfile.read())
39          #print (j[''])
40          #outfile.close
```

Listing 4: Python program to count the number of mementos from previous timemaps downloaded items

I reused **timemaps.py** from Assignment 2 and modified so that I can store the memento count for each of the 1000 URIs in **counting.csv** file.

```
1  import requests
2  import json
3  import os
4  import csv
5
6  #directory for storing timemaps for each uri
7  os.mkdir("timemaps")
8  count = 1
9
10 with open('counting.csv', 'a+', newline='') as csvfile:
```

```
11          fieldnames = ['Memento_Count']
12          writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
13          writer.writeheader()
14          csvfile.close()
15  with open('1000_links.txt') as fp:
16          for line in fp:
17  #get request made to memgatorin as json
18                  url = 'http://memgator.cs.odu.edu/timemap/json/'
                        + line.strip()
19
20
21                  try:
22                          req = requests.get(url, stream=True,
                                headers={'User-Agent':'Mozilla/5.0'})
23                          resp = requests.head(url, stream=True,
                                headers={'User-Agent':'Mozilla/5.0'})
24
25
26
27                          if req.status_code == 200 :
28
29  #dumping json data into respective timemap files for URI
30                                  with open('timemaps/%s.json' %
                                        count, 'w+') as outfile:
31                                          json.dump(req.json(),
                                                outfile, sort_keys =
                                                True, indent = 4,
                                                ensure_ascii = False)
32  #getting mementocount
33                                          mementocount = resp.
                                                headers.get('X-
                                                Memento-Count')
34                                          with open('counting.csv
                                                ', 'a+', newline='')
                                                as csvfile:
35                                                  fieldnames = ['
                                                        Memento_Count
                                                        ']
36                                                  writer = csv.
                                                        DictWriter(
                                                        csvfile,
                                                        fieldnames=
                                                        fieldnames)
37                                                  writer.writerow
                                                        ({'
                                                        Memento_Count
                                                        ':
                                                        mementocount
                                                        })
```

```
38                                                  csvfile.close()
39
40
41
42  #if 404 storing no mementos in for respective URI
43                              else :
44                                      with open('timemaps/%s.txt' %
                                            count, 'w+') as outfile1:
45                                          outfile1.write('No
                                                Momentos')
46                                          mementocount = 0
47                                          with open('counting.csv
                                                ', 'a+', newline='')
                                                as csvfile:
48                                              fieldnames = ['
                                                    Memento_Count
                                                    ']
49                                              writer = csv.
                                                    DictWriter(
                                                    csvfile,
                                                    fieldnames=
                                                    fieldnames)
50                                              writer.writerow
                                                    ({'
                                                    Memento_Count
                                                    ':
                                                    mementocount
                                                    })
51                                              csvfile.close()
52
53                              count = count + 1
54
55
56                      except:
57                              with open('timemaps/%s.txt' % count, 'w
                                    +') as outfile1:
58                                          outfile1.write('No
                                                Momentos')
59                                          mementocount = 0
60                                          with open('counting.csv
                                                ', 'a+', newline='')
                                                as csvfile:
61                                              fieldnames = ['
                                                    Memento_Count
                                                    ']
62                                              writer = csv.
                                                    DictWriter(
                                                    csvfile,
                                                    fieldnames=
```

```
                                                    fieldnames)
63                                                writer.writerow
                                                    ({'
                                                    Memento_Count
                                                    ':
                                                    mementocount
                                                    })
64                                                csvfile.close()
65                              count = count + 1
66                              pass
```

Listing 5: Python program to download timemaps and number of mementos

To plot the graph with respect to changes in size in timemaps, I wrote a program in r to calculate the difference in memento counts by subtracting **previousMC.csv** from **counting.csv**. The program is stored in **size.r**

```
1 setwd(getwd())
2 # csv dataframe for number of mementos
3 prevMementos <- read.table(header = TRUE, sep = ",", 'previousMC.
      csv')
4 numMementos <- read.table(header = TRUE, sep = ",", 'counting.csv
      ')
5
6 #calculating the difference
7 app <- numMementos-prevMementos
8 barplot(app$Memento_Count, main="Changes in Timemaps Overtime",
      xlab = "Timemaps",ylab = "Difference")
```

Listing 6: R script to plot graph for changes in timemeaps overtime

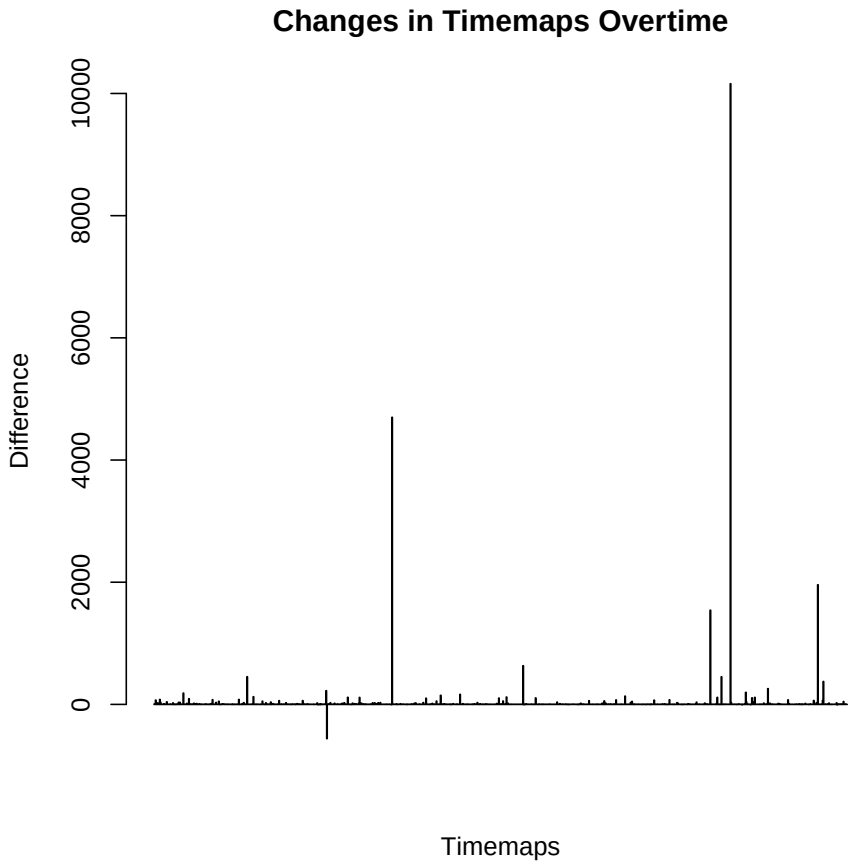

Figure 1: Bar graph for Timemap changes

References

[1] cmuth001/anwala.github.io. "GitHub" N.p., May 01, 2018.https://github.com/cmuth001/anwala.github.io/blob/master/Assignments/A7/blogdata.txt

[2] arthur-e/Programming-Collective-Intelligence. "GitHub". Web. May 01, 2018. https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter8/numpredict.py

[3] Cosine Similarity between 2 Number Lists"python - Cosine Similarity between 2 Number Lists - Stack Overflow". n.p., n.d. Web. May 01, 2018. https://stackoverflow.com/questions/18424228/cosine-similarity-between-2-number-lists

[4] Hrishi29/anwala.github.io. "GitHub" N.p., May 01, 2018.https://github.com/Hrishi29/anwala.github.io/tree/master/Assignments/A2