Assignment 9

Dr. Nwala

Nathaniel Everett

April 21, 2018

Contents:

Problem 1 3

1. 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

---

Two files I need are the blogdata.txt from Assignment 7 and the Python program given here, known as numpredict.py. I created a copy of numpredict.py known as numpredictA9.py, importing both the math and re libraries as necessary. If you notice, numpredict.py does not have a cosine() function, but it does have a euclidean() function. This one is not necessary, so I created my own cosine() function to use:

```
def cosine(vector1, vector2):
    sumxx = 0
    sumxy = 0
    sumyy = 0
    for i in range(0, 958):
        x = vector1[i]
        y = vector2[i]
        sumxx += x*x
        sumyy += y*y
        sumxy += x*y
    return sumxy/math.sqrt(sumxx*sumyy)
```

I also removed the majority of functions from numpredict.py, as they are not needed. I kept both the getdistances() and knnestimate() functions, making very minor changes to them to ensure correctness. Then I get started on my main program, which is titled nearestneighbor.py. First I open blogdata.txt, strip the lines and append the integers from the blog matrix into a list known as data. Two vectors, both for http://f-measure.blogspot.com/ and http://ws-dl.blogspot.com/, were created. The program goes accordingly:

```
result = numpredictA9.knnestimate(data, firstVector, 1)
print('kNN estimate: k = 1: ' + str(result))
```

The above code snippet gets the kNN estimate from http://f-measure.blogspot.com where k = 1. I repeat this snippet for k = 2, k = 5, k = 10, and finally k = 20, getting the kNN estimates for the first vector. Then I move to http://ws-dl.blogspot.com, which is the second vector, repeating the procedure. The results are shown:

```
http://f-measure.blogspot.com/ results
kNN estimate: k = 1: 0.012567107614194687
kNN estimate: k = 2: 0.013146496801386381
kNN estimate: k = 5: 0.02663877275942832
kNN estimate: k = 10: 0.04268398176685228
kNN estimate: k = 20: 0.07424903543185482
```

```
http://ws-dl.blogspot.com/ results
kNN estimate: k = 1: 0.0
kNN estimate: k = 2: 0.003625616577196871
kNN estimate: k = 5: 0.010625310279761366
kNN estimate: k = 10: 0.018242374280891586
kNN estimate: k = 20: 0.03260407942906764
```

I did have trouble with my blogdata.txt file, which I managed to fix by removing all non-int values (in other words, the names that are not numerical characters). This is the only thing I modified in the blogdata.txt file, which will only be used for this assignment.

Files included:
Assignment9Report.pdf – this file
numpredict.py – used as reference for this problem
blogdata.txt – modified blogdata file from Assignment 7
numpredictA9.py – modified numpredict.py with cosine function and removed other unneeded functions.
Nearestneighbor.py – main program.