CS-432/532 Introduction to Web Science:
Assignment #10:
kNN and SVM

Due on Saturday, April 30, 2016

Dr. Michael L. Nelson

Plinio Vargas
pvargas@cs.odu.edu

Contents

List of Figures

Listings

List of Tables

Problem 1

Using the data from A8:

- Consider each row in the blog-term matrix as a 500 dimension vector, corresponding to a blog.
- From chapter 8, replace numpredict.euclidean() with cosine as the distance metric. In other words, you'll be computing the cosine between vectors of 500 dimensions.
- 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}.

1.1 Approach

In order to complete this problem, three modules from [1] were taken and modified: **knnestimate**, **getdistances** and *euclidean*, the latest changed to **cosine_distance**. The heart of this approach resides in the **cosine_distance** module (see Listing 1), which takes as a parameter two 500-dimensional vectors.

Listing 1: Cosine Calculator: CosineDistance.py

```
65 def cosine_distance(v1, v2):
66     d = 0.0
67     for i in range(len(v1)):
68         d += (v1[i] - v2[i])
69
70     return 1 - math.cos(d)
```

Since $\cos\theta$ for any point is equal to 1 when x and y are equal in our plotting coordinate, if the cosine distance calculation between vector v_1 and v_2 is equal to zero, then $v_1 = v_2$. The further from 0 the calculation between v_1 and v_2 is, the greater the difference is between them.

Using the same approach as in [1], the values are loaded into a dictionary object (**data**). Term values are extracted from the first line of the input file (see Listing 2 line 21).

Listing 2: Main Module: CosineDistance.py

```
9  title = []
10 def main():
11     m = 500
12     n = 100
13     k = 0
14     terms = []
15     data = []
16
17     with open('blogdata.txt') as file:
```

Problem 1 continued on next page...

```
18          for record in file:
19              record = record.split()
20              if k < 1:
21                  terms = record[-500:]
22              else:
23                  print(k, len(record) - m, record[0:len(record) - m])
24                  title.append(record[0:len(record) - m])
25                  data.append({'input': tuple([int(x) for x in record[-500:]])})
26              k += 1
27      print(terms)
28
29      for k in range(100):
30          print('(%s, %d)' % (' '.join(title[k]), k), end=',')
31      print()
32
33      k_values = [1, 2, 5, 10, 20]
34      print('Closest Blog to %s' % ' '.join(title[76]))
35      for k in k_values:
36          print('\nk=%d' % k)
37          knnestimate(data, data[76]['input'], k)
38
39      print()
40      print('Closest Blog to %s' % ' '.join(title[68]))
41      for k in k_values:
42          print('\nk=%d' % k)
43          knnestimate(data, data[68]['input'], k)
44
45
46      return
```

The blog's title (line 24) is obtained by removing the 500-dimensional vector values from the file entry row. To obtain the row index from our two studied blogs, we visually inspected the entry file (*blogdata.tx*). Index 76 corresponded to `http://f-measure.blogspot.com/`; index 68 corresponded to `http://ws-dl.blogspot.com/`

In order to get the cosine-distance between two blogs, we passed the vector values to function *getdistances*. As the textbook explains, it "calls the distance function on the vector given against every other vector in the dataset and puts them in a big list. The list is sorted so that the closest item is on the top".

Listing 3: Getdistances Module: CosineDistance.py

```
1  def getdistances(data, vec1):
2      distancelist = []
3
4      # Loop over every item in the dataset
5      for i in range(len(data)):
6          vec2 = data[i]['input']
7
8          # Add the distance and the index
9          distancelist.append((cosine_distance(vec1, vec2),i))
10
11     # Sort by distance
12     distancelist.sort()
```

```
13
14      return distancelist
```

To display k-blogs closest to blog-x, an iteration is performed to call function *knnestimate*. This function, a modification from [1], simply returns the k-elements from the sorted list provided by function *getdistances* between blog-x and the other 99 blogs.

Listing 4: Knnestimate Module: CosineDistance.py

```
1  def knnestimate(data, vec1, k=5):
2      # Get sorted distances
3      dlist = getdistances(data, vec1)
4      avg = 0.0
5
6      print(dlist)
7      # Take the average of the top k results
8      for i in range(1, k + 1):
9          idx = dlist[i][1]
10         print(' '.join(title[idx]))
11
12     return avg
```

1.2 Solution

Since we clustered our 100 blogs in assignment 8, we can use the previous work as a comparison. Tables 1 through 10 contain our two blog similarities among the selected 100 blogs, using cosine-distance calculator.

According to Table 1, the most similar blog to **F-Measure** is `http://didnotchart.blogspot.com/`. This blog passed the eye test since it is also a blog related to music. Looking at the dendrogram from assignment 8, "Did Not Chart" is clustered close to **F-Measure**, but is not the closest. **F-Measure** was clustered closer to `http://ihatethe90s.blogspot.com/`. The last is also a blog related to music, so it also passed the eye test.

Then, which blog is the most similar to **F-Measure**? `http://ihatethe90s.blogspot.com/` or `http://didnotchart.blogspot.com/`? To answer this question we need to dive into the raw data, the DNA of our research:

DidNotChart
20110580000300201201000000001320220160020001107101402304011100002133231000000028221102104201470307210000271830001010130110000007300240105000000011301301112000042010006110520002002001100600005006001200010100211021032031400100000400011010000220601100900101103000031310034201000004006209010010000082010311201101000141030301055380420301011000000110301120411100000023000020003000102000001360000110300410041200317001101051200200180000105115011031031000100100007400712500102130020040301001010105011011033000000011000

F-Measure
20010003801111113621000400000100010130000001090102002100000100040000120320002002510000010041090502001010081260312211022400100302040005030110005101020020200212124040211121448000020000122080001100100111000010102021018302202010300110123402002182103101200510160041000200302010000010103000112400120001101610110022002001146000002122220221002031422000150202025030330110000505100001110020000100100015000602000014140101013060011800710001000160011000108011020000001214200000500102500301601010114003010120500130200212330060

IHateThe90s
54101436110063000111532610014503410141541441031102041502321122507404029109021191311274001966276235740638572282415100014910212011827014312221667521601610020114504224350371340501647740111211880021413171302533340303022052716121731045862133032741126123391122646221113601421213132251283126734412111311101461111200930361216332302601355224414116022175331910320211641030125491745601965317211133914122066001526116214130311623243034631584234123730131076015605020182263108033016156556757612281113210134304341761113241101393001220764517106106015120529121358036282112700000011215118212152165311111147034

To understand the raw data above, we need to inspect the 500 matrix file: *blogdata.txt*. The DNA thickness difference between I-Hate-The-90s and F-Measure is not due to extra terms, but rather many term frequency in I-Hate-The-90s. The differences are in the double digits, while in F-Measure and Did-Not-Chart, the differences are in the single digits. Then, although the blogs could be talking about the same thing, their frequencies are so far apart that they make them less similar.

As we expand the number of blog k-neighbors for K=20, an examination to the most distant neighbors "funky little demons" revealed this blog is not exactly about music, but it has lyrics which are related to

Problem 1 continued on next page...

music. Looking at dendrogram from assignment 8, “funky little demons” was not clustered near **F-Measure**, and the deepest element in the cluster is `http://kidchair.blogspot.com/`, which is about songs (music).

F-Measure
2001000380111111362100040000010001013000000109010200210000010004000012032000200251
0000010041090502001010081260312211022400100302040005030110005101020020200212124040
2111214480000200001220800011001001110000101020210183022020103001101234020021821031
0120051016004100020030201000001010300011240012000110161011002200200114600000212222
0221002031422000150202025030330110000505100001110020000100100015000602000014140101
0130600118007100010001600110001080110200000012142000005001025003016010101140030101
20500130200212330060

funkylittledemons
0000000200000000000000000000002000000000000000000000000000000000100000000000000000
0001010000000000000000000000000010000000000210000000000000000000000000000000000000
0000010000000000010000000000000000000000000010000100000001000000000010000000000000
0000000000000000001010000010000001000000000000000000000000000000000000000000000000
0000001000010000100000001100000100000000001001000000000000000000000000001000100000
0000000100030000200000100000000000000000000000000000000000010000000000000000000000
00000000

KiDCHAIR
1600102010030100220010000012200000000101100000001110003200200220200201003300000001101
30000002204030801000000152300001720110001010000005000010000033001221000100151120 10
0021024010000210600210020000120030040011000021204000000110201310000000000102010113
0172030000010000100010000010401002013030002110010020012000113100500000011101000001
0000101510021210010310300011600100000000300120001100107110000004100021120503100010
0004010200100001060010101000002100010002600001510201101100200000010122165101001111
0000101041000

The answer to the question which one is more similar to **F-Measure** between the last two blogs is more difficult to answer from a 40K-feet-view looking at their DNAs. The frequency of 500 terms in “funky little demons” is very low; many elements in the vector have zero values. We could be very quick to say that `http://kidchair.blogspot.com/` is more similar to **F-Measure** than “funky little demons”, but if we were going element by element in the vector, we will notice the combined difference of the last blog is smaller.

A surprising result was the comparison with `http://ws-dl.blogspot.com/`. The closest blog in similarity was `http://lostintheshuffle899.blogspot.com/` - Lost in the Shuffle (see Table 6); however in the dendrogram from assignment 8, it was clustered closer to **SpinitronBlog**.

Using similar comparison to their DNAs, we can notice **SpinitronBlog** does not have too many terms while **LostintheShuffle** does, making it closer by default.

WebScienceandDigitalLibrariesResearchGroup
3800605137033020145201800443102040118128020060021020010112007011207710014600000031
7131702030031167060300019306001230101301900040113101800001208002000010100211060413
3075030204511041112157802890560271238200024611100723112092600200015630000007370010
6843114500301131012124200206105100050201310880020190100050002000860133022000000031
2615000101340100010200231360210011150140004014172012242045011200011501183800022121

0254012010003512030003000221310206097212151313220710130201101111001792541113150270
002200100000060702012430100110000100191801016400041051001031302133901000502 12

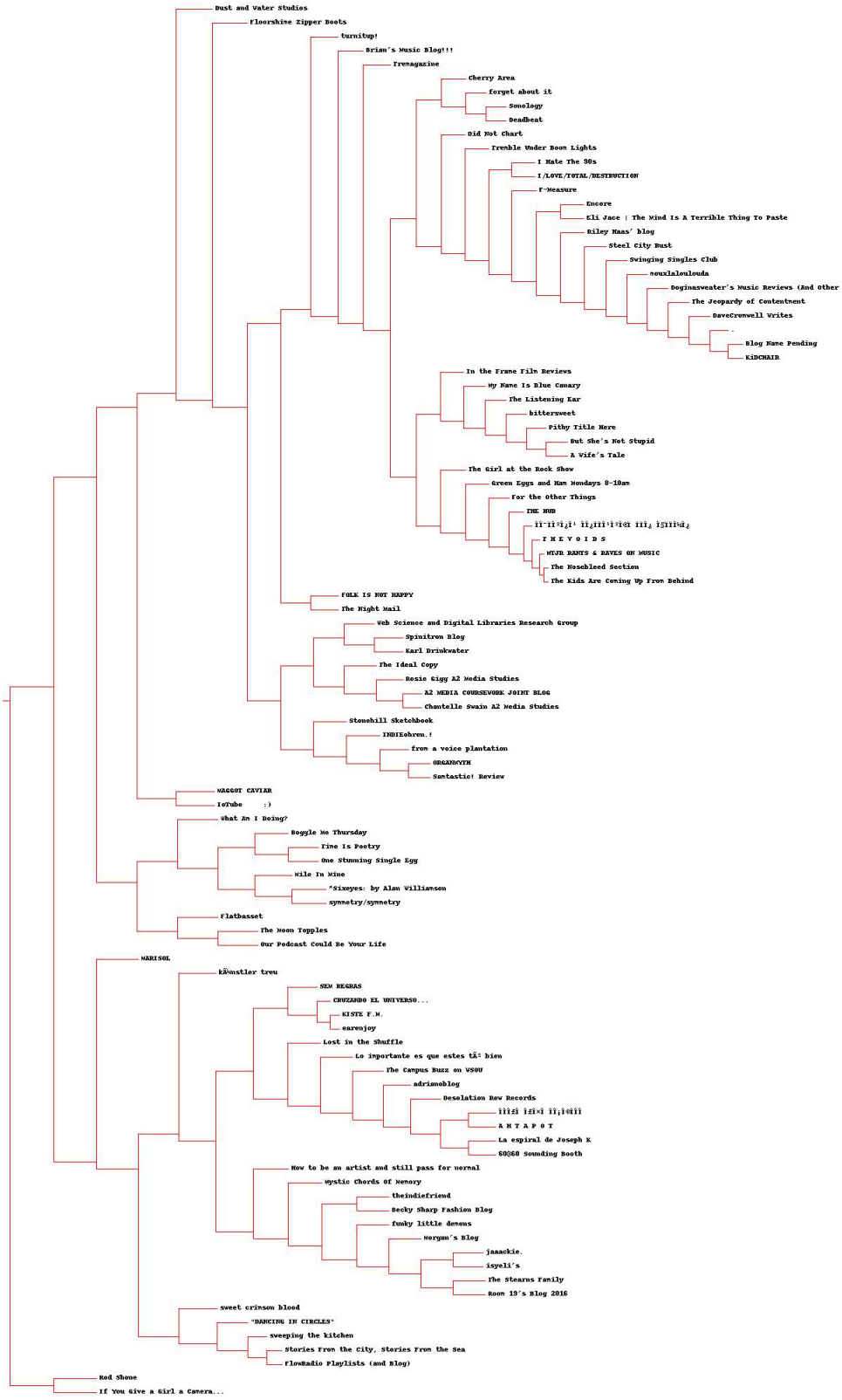
LostintheShuffle
0002030202200010014200121000001051001000000001002100000200101000000005000100121610 00
2102000014010100002000010011000301010000130000201000000000100000012000000010122001 10
2800210100000102000000001601000110203200001001300003100010001020100410220010102000
0000101000100002110000102000000300000000100000106033140001000000000000200200041000 0
0000000400001000011011017101010000000009000101119020011101212100000020001000000102 1
0001010031110100200000010400203100002030000000000000091017000220020001000000025000 30
111000000000000010

SpinitronBlog
0000000000000000000010000000000000010000000000000000000030000000000010000000010005070
0000000000000061001000000000007000000000003000000000000000000000100000000100000000
0010000000000010000000000000000000000000000010000000200001000001000000000000000000
0000201000000100001000000000000000000000000001000000000000000000000000000000000000
0220100000020000001000000011001900000000100000000001000000000009000000000101001000 0
0000000050000010001100000000000000000000000000000000000000000000050000000000300
00000000000

Blog-Terms:
groups-ashtray-stretching-sum-indians-adventure-rightly-cell-reviewing-tstyle-reverse-statements-chilling-existent-sleepy-fortunately-typically-settle-exit-lone-behave-compositions-bastards-canned-pagination-jacob-reflective-legit-eleven-exclusively-spending-rival-wings-license-norm-encourage-haircut-emerge-actors-beg-honored-closet-outfits-pr-collector-whore-teenagers-leisure-mob-ali-marry-hopeful-jobs-knocking-westerns-pleased-females-elliot-someday-cutting-ken-historic-harris-meditation-insane-chant-myth-challenge-passive-longing-curse-lenny-jewelry-ducks-supreme-joan-graduation-nigel-recordings-pedal-security-tale-butter-redhead-spice-oklahoma-html-ringing-combine-army-sentiments-funk-cuff-violence-orbit-producer-ariel-switched-snakes-careless-equation-arriving-larger-syd-trivia-sharp-quit-critic-benefits-norton-colbert-glen-industry-clue-appeared-rivers-shaking-recipe-expanded-undoubtedly-endings-dancer-sailing-viewed-jurado-drove-kil-sheen-cassettes-steam-twigs-rockabilly-pummeling-reminder-sensibility-practically-lame-grande-resulting-chatter-afghan-remembering-breathtaking-carl-alps-happiness-buying-carrie-careers-dual-bucks-origins-quintessential-infinite-andrews-sunny-tower-chapters-ipod-immortal-ma-historically-cinema-escaped-abuse-powered-eric-argue-lip-approaches-schools-cramps-ash-blogged-gentleman-caring-downs-cos-appropriate-attempting-accepting-mo-eliminate-polar-cornell-flash-christian-leaning-electricity-strongly-grief-polite-responsible-rebellion-fake-rides-implies-ha-melvins-remove-plethora-gained-bundle-mc-crashed-astronaut-celebrate-fists-patient-knowledge-pulled-wives-symphony-laundry-instinct-spiritualized-mister-allow-landed-minority-futuristic-catches-surprises-claimed-index-railroad-fictional-luke-importance-aquarium-daddy-neighbors-fourteen-dime-kim-sports-delay-loser-slower-kramer-gutter-grinding-pro-gods-vocabulary-movies-goal-mondays-birth-mould-partnership-skinny-upcoming-grandfather-delivering-groundbreaking-jacuzzi-skills-announcement-insanely-sprawling-lucas-discography-served-humming-ashley-idiots-crows-truckers-subsequently-amused-effortlessly-haze-additions-flourishes-mccarthy-heath-execution-formed-tons-analysis-winding-percentage-evokes-donuts-impress-outlet-zone-surviving-julian-complain-wheels-palace-edward-shaped-ratio-tones-ry-cred-trapped-puppets-simpson-insert-consume-zeppelin-virtue-eps-nostalgic-alaska-bend-swallowed-accidentally-hallelujah-russia-

Problem 1 continued on next page. . .

directed-misery-greg-bizarre-dean-exhausting-eccentric-formerly-clicks-accurate-crashes-mod-rarity-goodies-kenny-shannon-pros-categories-majority-audible-cosmic-euro-proven-product-remedy-elliott-problematic-amongst-slate-blew-expert-tribe-circus-vox-iris-ny-ye-crushed-cartoon-parties-university-hyper-sidewalk-jan-trials-promised-wisconsin-grape-victims-matched-graphic-booker-umbrella-eternal-junior-swamp-oil-drake-originality-apartment-potter-marianne-explosive-dum-lily-chromatics-magnet-disgusting-pockets-stanley-assured-oldham-illustrated-legend-ads-logo-continent-noir-respond-tribal-sites-barnes-delivery-impression-health-info-chairs-los-straw-directors-desk-simone-thin-cheaper-thrive-allen-maiden-elite-ab-combined-venue-jonny-nothings-emergency-axis-studied-necessity-bush-confirm-newest-walker-shades-windows-realistic-handy-mystic-correct-palmer-eden-sans-observed-coachella-gimmick-fashioned-traded-titus-ellis-couch-bean-bursts-labor-quietly-featured-tease-species-louise-guessed-reviewer-experimental-ms-overhead-breakdown-drifting-amy-finest-northwest-babes-lonesome-owls-korea-bow-swords-confidence-suzanne-observations-trumpet-pile-hating-reflects-websites-horribly-cheating-sufjan-intensity-advantage-engage-zombie-laced-moore-wears-tremendous-winners-songwriters-concise-temptation-entering-jefferson-casey-veteran-coincidentally-distorted-riding-worlds-germans-lang-manning



Problem 1 continued on next page. . .

Table 1: F-Measure with K=1

Blog Title
Did Not Chart

Closest K-Blogs to F-Measure for K=1.
Values obtained from CosineDistance.py

Table 2: F-Measure with K=2

Blog Title
Did Not Chart
THIS CHARMING YAN

Closest K-Blogs to F-Measure for K=2.
Values obtained from CosineDistance.py

Table 3: F-Measure with K=5

Blog Title
Did Not Chart
THIS CHARMING YAN
Encore
isyeli's
If You Give a Girl a Camera...

Closest K-Blogs to F-Measure for K=5.
Values obtained from CosineDistance.py

Problem 1 continued on next page...

Table 4: F-Measure with K=10

Blog Title
Did Not Chart
THIS CHARMING YAN
Encore
isyeli's
If You Give a Girl a Camera...
mattgarman
Rants from the Pants
60@60 Sounding Booth
turnitup!
Doginasweater's Music Reviews (And Other ..)

Closest K-Blogs to F-Measure for K=10.
Values obtained from CosineDistance.py

Table 5: F-Measure with K=20

Blog Title
Did Not Chart
THIS CHARMING YAN
Encore
isyeli's
If You Give a Girl a Camera...
mattgarman
Rants from the Pants
60@60 Sounding Booth
turnitup!
Doginasweater's Music Reviews (And Other ..)
"DANCING IN CIRCLES"
I/LOVE/TOTAL/DESTRUCTION
Lost in the Shuffle
this time tomorrow
Web Science and Digital Libraries Research Group
FlowRadio Playlists (and Blog)
.
Stories From the City, Stories From the Sea
The Jeopardy of Contentment
funky little demons

Closest K-Blogs to F-Measure for K=20.
Values obtained from CosineDistance.py

Table 6: dl.blogspot.com with K=1

Blog Title
Lost in the Shuffle

Closest K-Blogs to dl.blogspot.com for K=1.
Values obtained from CosineDistance.py

Table 7: dl.blogspot.com with K=2

Blog Title
Lost in the Shuffle
“DANCING IN CIRCLES”

Closest K-Blogs to dl.blogspot.com for K=1.
Values obtained from CosineDistance.py

Table 8: dl.blogspot.com with K=5

Blog Title
Lost in the Shuffle
“DANCING IN CIRCLES”
Stories From the City, Stories From the Sea
Doginasweater’s Music Reviews (And Other Horse ...)
turnitup!

Closest K-Blogs to dl.blogspot.com for K=5.
Values obtained from CosineDistance.py

Table 9: dl.blogspot.com with K=10

Blog Title
Lost in the Shuffle
“DANCING IN CIRCLES”
Stories From the City, Stories From the Sea
Doginasweater’s Music Reviews (And Other Horse ...)
turnitup!
The Jeopardy of Contentment
If You Give a Girl a Camera...
mattgarman
Pop Tones
isyeli’s

Closest K-Blogs to dl.blogspot.com for K=10.
Values obtained from CosineDistance.py

Table 10: dl.blogspot.com with K=20

Blog Title
Lost in the Shuffle
“DANCING IN CIRCLES”
Stories From the City, Stories From the Sea
Doginasweater’s Music Reviews (And Other Horse ...)
turnitup!
The Jeopardy of Contentment
If You Give a Girl a Camera...
mattgarman
Pop Tones
isyeli’s
Encore
THIS CHARMING YAN
Samtastic! Review
Did Not Chart
F-Measure
Room 19’s Blog 2016
Rants from the Pants
60@60 Sounding Booth
Cherry Area
I/LOVE/TOTAL/DESTRUCTION

Closest K-Blogs to dl.blogspot.com for K=20.
Values obtained from CosineDistance.py

References

[1] Segarn, Toby. Programming Collective Intelligence. *Building Smart Web 2.0 Application*. (pp 29-53). Sebastopol, CA: O'Reilly Media.

[2] Text Mining, Analytics & More. (n.d.) Retrieved April 21, 2016, from `http://www.text-analytics101.com/2014/10/computing-precision-and-recall-for.html`