CSIT 6000I Search Engines and Applications

Mid-Term Examination II, Fall 2019

23 Nov, 2019, 1:30pm

Time Allowed: 24 hours

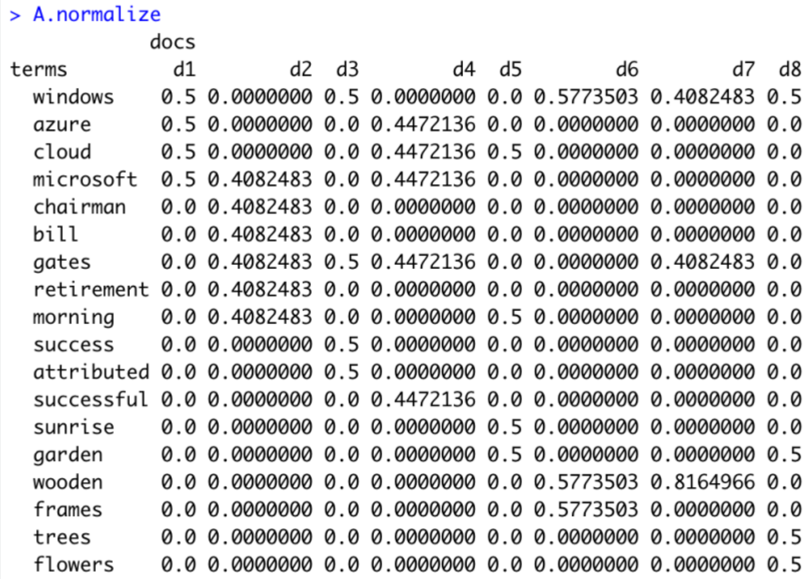
Rule: You can use any tool or information source to answer the questions. You are allowed to be online and use any tools or information sources to help you with answering the questions. The only restriction is that YOU MUST DO IT BY YOURSELF AND NOT CONSULT OR DISCUSS WITH ANY PEOPLE.

1. [40] You have the following documents, and important keywords to be indexed are underlined.

D1	I use Windows and Azure Cloud from Microsoft.
D2	Microsoft Chairman Bill Gates announced retirement this morning.
D3	Success of Windows is attributed to Gates.
D4	Azure Cloud becomes successful after Gates left Microsoft.
D5	I watched sunrise in my garden. There was a little cloud in the morning.
D6	I like windows with wooden frames.
D7	I have wooden windows and wooden gates.
D8	Through the windows, I can see trees and flowers in my garden.

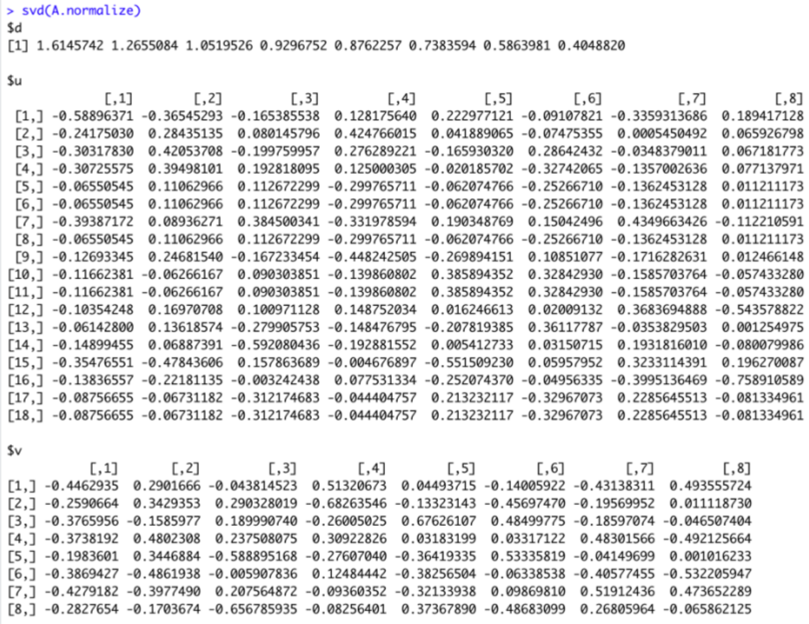
Using R, Python or any programming language or online websites you are familiar with, answer the following questions. You can include the screenshot, or copy-and-paste the output into your answer.

a) [5] Create the term-document matrix for the document set; use tf as the term weight, and normalize the document vector lengths to 1.



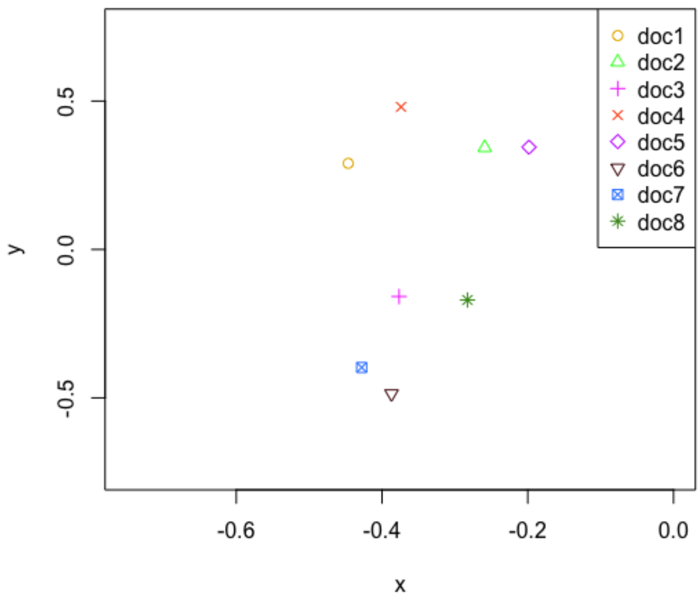
```
> A.normalize
            docs
terms         d1        d2  d3        d4  d5        d6        d7  d8
  windows    0.5 0.0000000 0.5 0.0000000 0.0 0.5773503 0.4082483 0.5
  azure      0.5 0.0000000 0.0 0.4472136 0.0 0.0000000 0.0000000 0.0
  cloud      0.5 0.0000000 0.0 0.4472136 0.5 0.0000000 0.0000000 0.0
  microsoft  0.5 0.4082483 0.0 0.4472136 0.0 0.0000000 0.0000000 0.0
  chairman   0.0 0.4082483 0.0 0.0000000 0.0 0.0000000 0.0000000 0.0
  bill       0.0 0.4082483 0.0 0.0000000 0.0 0.0000000 0.0000000 0.0
  gates      0.0 0.4082483 0.5 0.4472136 0.0 0.0000000 0.4082483 0.0
  retirement 0.0 0.4082483 0.0 0.0000000 0.0 0.0000000 0.0000000 0.0
  morning    0.0 0.4082483 0.0 0.0000000 0.5 0.0000000 0.0000000 0.0
  success    0.0 0.0000000 0.5 0.0000000 0.0 0.0000000 0.0000000 0.0
  attributed 0.0 0.0000000 0.5 0.0000000 0.0 0.0000000 0.0000000 0.0
  successful 0.0 0.0000000 0.0 0.4472136 0.0 0.0000000 0.0000000 0.0
  sunrise    0.0 0.0000000 0.0 0.0000000 0.5 0.0000000 0.0000000 0.0
  garden     0.0 0.0000000 0.0 0.0000000 0.5 0.0000000 0.0000000 0.5
  wooden     0.0 0.0000000 0.0 0.0000000 0.0 0.5773503 0.8164966 0.0
  frames     0.0 0.0000000 0.0 0.0000000 0.0 0.5773503 0.0000000 0.0
  trees      0.0 0.0000000 0.0 0.0000000 0.0 0.0000000 0.0000000 0.5
  flowers    0.0 0.0000000 0.0 0.0000000 0.0 0.0000000 0.0000000 0.5
```

b) [5] Perform SVD on the matrix and show the USV matrices.

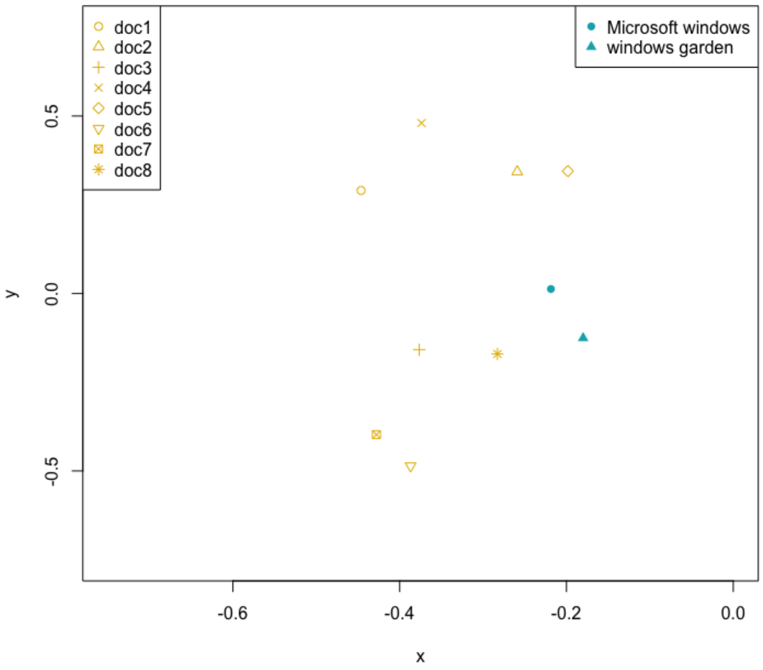


```
> svd(A.normalize)
$d
[1] 1.6145742 1.2655084 1.0519526 0.9296752 0.8762257 0.7383594 0.5863981 0.4048820

$u
             [,1]        [,2]         [,3]         [,4]         [,5]        [,6]          [,7]         [,8]
 [1,] -0.58896371 -0.36545293 -0.165385538  0.128175640  0.222977121 -0.09107821 -0.3359313686  0.189417128
 [2,] -0.24175030  0.28435135  0.080145796  0.424766015  0.041889065 -0.07475355  0.0005450492  0.065926798
 [3,] -0.30317830  0.42053708 -0.199759957  0.276289221 -0.165930320  0.28642432 -0.0348379011  0.067181773
 [4,] -0.30725575  0.39498101  0.192818095  0.125000305 -0.020185702 -0.32742065 -0.1357002636  0.077137971
 [5,] -0.06550545  0.11062966  0.112672299 -0.299765711 -0.062074766 -0.25266710 -0.1362453128  0.011211173
 [6,] -0.06550545  0.11062966  0.112672299 -0.299765711 -0.062074766 -0.25266710 -0.1362453128  0.011211173
 [7,] -0.39387172  0.08936271  0.384500341 -0.331978594  0.190348769  0.15042496  0.4349663426 -0.112210591
 [8,] -0.06550545  0.11062966  0.112672299 -0.299765711 -0.062074766 -0.25266710 -0.1362453128  0.011211173
 [9,] -0.12693345  0.24681540 -0.167233454 -0.448242505 -0.269894151  0.10851077 -0.1716282631  0.012466148
[10,] -0.11662381 -0.06266167  0.090303851 -0.139860802  0.385894352  0.32842930 -0.1585703764 -0.057433280
[11,] -0.11662381 -0.06266167  0.090303851 -0.139860802  0.385894352  0.32842930 -0.1585703764 -0.057433280
[12,] -0.10354248  0.16970708  0.100971128  0.148752034  0.016246613  0.02009132  0.3683694888 -0.543578822
[13,] -0.06142800  0.13618574 -0.279905753 -0.148476795 -0.207819385  0.36117787 -0.0353829503  0.001254975
[14,] -0.14899455  0.06887391 -0.592080436 -0.192881552  0.005412733  0.03150715  0.1931816010 -0.080079986
[15,] -0.35476551 -0.47843606  0.157863689 -0.004676897 -0.551509230  0.05957952  0.3233114391  0.196270087
[16,] -0.13836557 -0.22181135 -0.003242438  0.077531334 -0.252074370 -0.04956335 -0.3995136469 -0.758910589
[17,] -0.08756655 -0.06731182 -0.312174683 -0.044404757  0.213232117 -0.32967073  0.2285645513 -0.081334961
[18,] -0.08756655 -0.06731182 -0.312174683 -0.044404757  0.213232117 -0.32967073  0.2285645513 -0.081334961

$v
           [,1]       [,2]         [,3]        [,4]        [,5]        [,6]        [,7]         [,8]
[1,] -0.4462935  0.2901666 -0.043814523  0.51320673  0.04493715 -0.14005922 -0.43138311  0.493555724
[2,] -0.2590664  0.3429353  0.290328019 -0.68263546 -0.13323143 -0.45697470 -0.19569952  0.011118730
[3,] -0.3765956 -0.1585977  0.189990740 -0.26005025  0.67626107  0.48499775 -0.18597074 -0.046507404
[4,] -0.3738192  0.4802308  0.237508075  0.30922826  0.03183199  0.03317122  0.48301566 -0.492125664
[5,] -0.1983601  0.3446884 -0.588895168 -0.27607040 -0.36419335  0.53335819 -0.04149699  0.001016233
[6,] -0.3869427 -0.4861938 -0.005907836  0.12484442 -0.38256504 -0.06338538 -0.40577455 -0.532205947
[7,] -0.4279182 -0.3977490  0.207564872 -0.09360352 -0.32133938  0.09869810  0.51912436  0.473652289
[8,] -0.2827654 -0.1703674 -0.656785935 -0.08256401  0.37367890 -0.48683099  0.26805964 -0.065862125
```

c) [5] Perform Rank-2 approximation and plot the documents on a two-dimensional space.



d) [10] Map the two queries (i) Microsoft windows, and (ii) windows garden, into the reduced vector space and plot them on the same two-dimensional graph as in (c).



e) [5] Using inner product, compute the top 3 documents for each query in the reduced vector space; compare them with the top 3 documents in the original vector space, i.e., directly from the term-document matrix in (a).

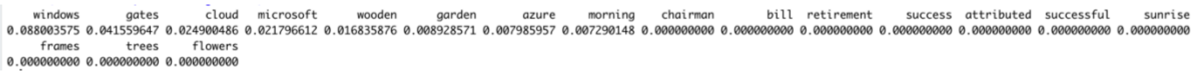
```
> #inner product: Original matrix A with query 1
> sort(x = combo.noDecompose[1,],decreasing = TRUE)
       d1        d6        d3        d8        d4        d2        d7        d5 
1.0000000 0.5773503 0.5000000 0.5000000 0.4472136 0.4082483 0.4082483 0.0000000 
> #inner product: rank-2 approximation with query 1
> sort(x = combo.decompose[1,],decreasing = TRUE)
        d1         d7         d4         d3         d6         d2         d8         d5 
0.10121032 0.08859115 0.08774027 0.08035969 0.07852597 0.06093286 0.05969652 0.04768133 
> #inner product: Original matrix A with query 2
> sort(x = combo.noDecompose[2,],decreasing = TRUE)
       d8        d6        d1        d3        d5        d7        d2        d4 
1.0000000 0.5773503 0.5000000 0.5000000 0.5000000 0.4082483 0.0000000 0.0000000 
> #inner product: rank-2 approximation with query 2
> sort(x = combo.decompose[2,],decreasing = TRUE)
          d6           d7           d3           d8           d1           d4           d2           d5 
 0.130723801  0.126993624  0.087719600  0.072304562  0.043909688  0.006992149  0.003574445 -0.007575229 
```

f) [10] Discuss how synonymy and polysemy are resolved, or are not resolved, with LSI in this example.

Unfortunately in this example we are submitted to the problems of synonymy and polysemy. Here even when using the original matrix we still struggle with polysemy, on query 1 we can see that the second best document is suffering of the polysemy effects.

2. [20] Using the document set in Q. 1, identify the two words that you would choose as index terms using term discrimination value.

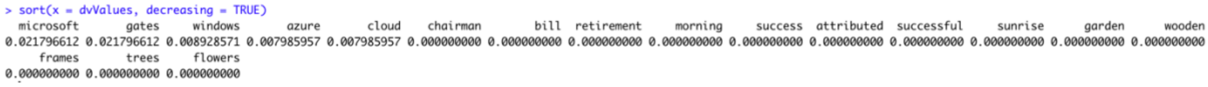
In first the try we use all the documents and compute the similarity and then sort we get the following terms as the best. So the best to relate all the documents are: “windows” and “gates”.



```
  windows     gates     cloud microsoft    wooden    garden     azure   morning  chairman      bill retirement   success attributed successful   sunrise 
0.088003575 0.041559647 0.024900486 0.021796612 0.016835876 0.008928571 0.007985957 0.007290148 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 
   frames     trees   flowers 
0.000000000 0.000000000 0.000000000 
```

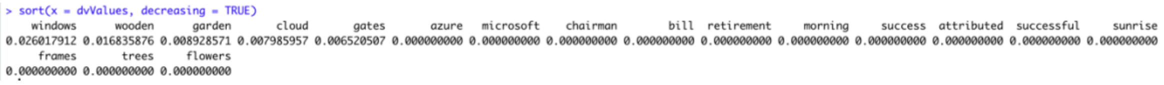
On the second try we analyzed the documents and group them, and them get the best terms using the term discrimination value. We grouped the docs in 2 groups: the first 4 which are related to the company Microsoft (Windows), and them the other 4 which are related to the to windows and gates.

To the first group this are the terms



```
> sort(x = dvValues, decreasing = TRUE)
 microsoft     gates   windows     azure     cloud  chairman      bill retirement   morning   success attributed successful   sunrise    garden    wooden 
0.021796612 0.021796612 0.008928571 0.007985957 0.007985957 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 
   frames     trees   flowers 
0.000000000 0.000000000 0.000000000 
```

Second group



```
> sort(x = dvValues, decreasing = TRUE)
  windows    wooden    garden     cloud     gates     azure microsoft  chairman      bill retirement   morning   success attributed successful   sunrise 
0.026017912 0.016835876 0.008928571 0.007985957 0.006520507 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 
   frames     trees   flowers 
0.000000000 0.000000000 0.000000000 
```

So the two words chosen are: “microsoft” to the first group, and “windows” for the second.

3. [20] A search for <u>trade conflict</u> returns the following results. The check marks mean the user clicked the corresponding pages; unchecked pages are not clicked. The user clicked Page 2 before Page 8. Show the page preference pairs derived from each of the five preference mining strategies discussed in the lecture.

1.	China–United States trade war - Wikipedia	
2.	Trade war - Wikipedia	√
3.	The Impact of Trade Conflict on Developing Asia | Asian ...	
4.	US-China trade dispute | Financial Times	
5.	US-China trade war | South China Morning Post	
6.	The US-China Trade War: A Timeline - China Briefing News	
7.	A quick guide to the US-China trade war - BBC News	
8.	Trading Away from Conflict: Using Trade to Increase ...	√
9.	China and the United States: Trade Conflict and Systemic ...	

Strategy 1: Click > Skip Above → L2 > L1 || L8 > L1 || L8 > L3 || L8 > L4 || L8 > L5 || L8 > L6 || L8 > L7
Strategy 2: Last Click > Skip Above → L8 > L1 || L8 > L3 || L8 > L4 || L8 > L5 || L8 > L6 || L8 > L7
Strategy 3: Last Click > Click Earlier → L8 > L2
Strategy 4: Last Click > Skip Previous → L2 > L1 || L8 > L7
Strategy 5: Click > No Click Next→ L2 > L3 || L8 > L9

4. [10] The web consists of a set of linked web pages; search engine is a software which directs users to a small subset of the webpages (typically 10 pages for each query). Discuss how the existence of Google, which implements PageRank and indexes all pages on the web, impacts the random surfer model, in which users first pick a random page to visit and after landing on the page, either randomly pick another page to visit or following a random link in that page. Write no more than 5 sentences.

Google also has analytics, which can gather a lot of information included the links clicks. Therefore, if I’m in a website and I click in a link there is a chance which the linked website is related to the original one. This also happens in the random surfer, e.g. if I’m reading one new, I can go directly to another website to confirm the new, in this way the new page is also in some sense related. With this information google can use the linked websites and the random surfer model websites to related them in the search results. Once we search for one of the websites we could also see the other ones.