INTRO. TO INFO RETRIEVAL: CS 734: A5

Due on Friday, December 15, 2017

Dr. Nelson

Udochukwu Nweke

Contents

$$R'_d = (R_d - R_{min})/(R_{max} - R_{min})$$

Figure 1: Resource Normalization

$$S'_d = S_d(\alpha + (1 - \alpha)R'_d)$$

Figure 2: Document Normalization

Problem 1

10.11 Suggest how the maximum and minimum resource ranking scores, R_{max} and R_{min}, could be estimated for a given query

Solution 1:

In a distributed search environment, a meta search engine (main node) broadcasts a query to multiple different search engines (pair nodes). The pair nodes return a result ranking which is merged by the main node. To select a resource, the main node ranks the pair nodes based on their query likelihood scores for a given query. It then picks the top k nodes which exceed a threshold. After selecting the nodes, a local search is carried out on the nodes and the result is sent to the main node. If the different nodes may have different retrieval models, the ranks from the pair nodes must be normalized before merging to produce a single ranking. The resource ranking score of a document S_d has to be normalized to get the local ranking of score of the document S_d^I which is the score used to get the final ranking. Figure 1 and Figure 2 shows the resource normalization and document normalization respectively.

We need to compute R_{max} and R_{min} in order to get the resource normalization scores.

Estimating R_{max} and R_{min}

Method 1

R_{max} and R_{min} are the maximum and minimum possible scores for a given query. We can estimate these by finding the highest and lowest scores that the resource ranking algorithm could potentially assign to a node database. If T in the equation in Figure 3 is set to 1, we achieve the maximum score the ranking algorithm can assign to a query - R_{max}. If we set T to 0 in Figure 3, we get the minimum ranking score a ranking algorithm can assign to a query.

Method 2

The R_{max} and R_{min} values in Method 1 might be too strict, but R_{max} is often an overestimate in practice. We can also estimate R_{max} and R_{min} by taking finding the maximum (R_{max}) and minimum (R_{min}) ranking scores from a corpus statistics: by finding the maximum and minimum ranking scores

$$T = \frac{df}{df + 50 + 150 \cdot cw/avg\_cw} \tag{5.1}$$

$$I = \frac{\log\left(\frac{C+0.5}{cf}\right)}{\log(C+1.0)} \tag{5.2}$$

$$p(r_k|c_i) = b + (1-b)\cdot T \cdot I \tag{5.3}$$

Figure 3: T can be used to estimate R_{max} and R_{min}

Problem 2

10.3. Compute five iterations of HITS (see Algorithm 3) and PageRank (see Figure 4.11) on the graph in Figure 10.3. Discuss how the PageRank scores compare to the hub and authority scores produced by HITS.

Solution 2:

Listing 1: Five Iterations for HITS and PageRank Code Snippet

```
import matplotlib.pyplot as plt
import networkx as nx

def runPageRank(G):
     pr = nx.pagerank(G)
     #iter: 5
     '''{1: 0.12066620697948402, 2: 0.22304902340308913, 3: 0.09418933961948064,
     4: 0.2795274111395043, 5: 0.09418933961948064, 6: 0.09418933961948064,
     7: 0.09418933961948064}'''
4, 2, 1, (3, 5, 6, 7 - tied)

def runHITS(G):
     h, a = nx.hits(G)
     print 'h:', h
     print 'a:', a

     #iter: 5
     '''pre h: {1: 0.310838445807771, 2: 0.0, 3: 0.9999999999999999,
     4: 0.0, 5: 0.45194274028629855, 6: 0.45194274028629855, 7: 0.0}'''
     '''pre a: {1: 0.5248868778280543, 2: 0.6877828054298643, 3: 0.0,
     4: 0.9999999999999999, 5: 0.0, 6: 0.0, 7: 0.0}'''
4, 2, 1, (3,5,6,7 - tied)



G = nx.DiGraph()
G.add_node(7)
G.add_edge(1, 2)
G.add_edge(3, 1)
G.add_edge(3, 2)
G.add_edge(3, 4)
G.add_edge(5, 4)
G.add_edge(6, 4)

runHITS(G)
```

Problem 2 continued on next page. . .

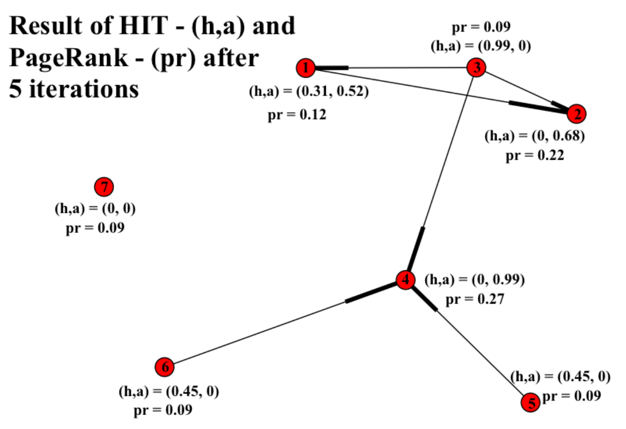


Figure 4: HIT and PageRank After 5 iterations

```
#runPageRank(G)



#nx.draw(G)
#plt.savefig('graph.png')
```

I used the python library networkx to apply the HITS and PageRank algorithms to the graphs in Figure 5.

For both algorithms, the computation did not converge after 5 iterations, so I printed the scores at 5 iterations by modifying the networkx HITS (*hits_alg.py*) and PageRank (*pagerank_al.py*) in order to see the scores at 5 iterations. The HITS algotihm converged after 17 iterations while the PageRank algorithm converged after 14 iterations. The result is in Figure 4

The ranking results of the HITS authority score is the same as the PageRank score. For example, the HITS algorithm ranks the nodes accordingly from highest authority score to lowest authority score:
4, 2, 1, (3,5,6,7 - tied)
Compared to the PageRank order from highest page rank to lowest:
4, 2, 1, (3, 5, 6, 7 - tied)

The PageRank algorithm gives an initial rank to nodes without incoming links. This is why the isolated node 7 has the same page rank as nodes 3, 5, and 6 because they all do not have any incoming links. However, the HITS algorithm distinguishes between the isolated node 7 and non isolated nodes 3, 5, and 6.

Problem 3

10.5. Find a community-based question answering site on the Web and ask two questions, one that is low-quality and one that is high-quality. Describe the answer quality of each question.

Solution 3:

The community-based question and answering system I used is to ask a high quality "yahoo question and answer forum".
The high quality questuion I ask is :" How long does it take to get comfortable coding in any programming

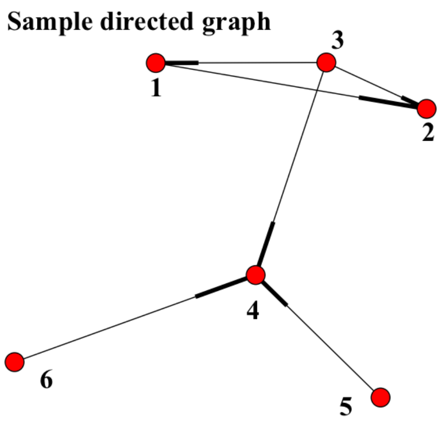
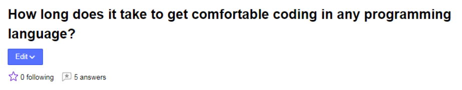


Figure 5: Original Graph



How long does it take to get comfortable coding in any programming language?

Figure 6: High Quality Question

language?"

Observation

I received answers to this question in a couple of hours. But I received about five informative answers. I was even given some tips and books on how to start and get comfortable with programming. The question and the number of answers is in Figure 6

For my low quality question, I asked: "What is your favorite bible traanslation?". I asked this question in `https://christianity.stackexchange.com/` and it did not return any answer but it has viewed. After sometime, it was deleted.

This is a low quality question because it is a subjective question where every answer is valid. There is no wrong answer.

Observation: My observation is that high quality questions receive high quality answers. It might take some time, but the answers are relevant to the questions. Low quality questions on the other hand receive low quality or no answers. They might be viewed sometimes more than the high qulity questions but they are either flagged and possibly removed, depending on how low of a quality they are.

Figure 7: Purchased Item

Problem 4

10.6. Find two examples of document filtering systems on the Web. How do they build a profile for your information need? Is the system static or adaptive?

Solution 4:
The document filtering systems:

1. Amazon
2. Netflix

Figure7 is an image of a banner stand I purchased from amazon. After I purchased the banner stand, I have not stopped receiving recommendations from amazon for similair banner stands. Figure8 is an example of the variety of banner stands amazon has been recommending and they are similar to the one I bought. I also have been receiving other items that are not similar to the item I purchase. Some of them are not banner stand. They were purchased from users that also purchased the same banner stand I did.

Amazon builds their profile based on products that users purchase. They use this profile to recommended products to users based on similar purchase. This system is adaptive.

Just like amazon, Neftlix also uses the same system(adaptive system) to recommend movies to users. When a user sees a movie on Netflix, similar movies are recommended to the user. The user also receives other recommendations based on similar movies that were seen by other users.

Problem 5

Extra Credit: Wiki Small collection, 8 points extra credit:
download the wikipedia.org page URIs from the live web.
summarize the HTTP status codes returned– all 200? Redirections? 404s? do we still have 6043 documents? or have their been splits & merges (see also: wikipedia disambiguation pages)? (1 pt)
compare and contrast the in- and out-links from this collection. the number, the domains linked to, the HTTP status of the non-wikipedia links in the article (i.e., are they 200, 404, or something else?). (1 pt)
compare the least and most popular articles (in terms of in-links) in the 2008 snapshot vs. the least and most popular articles in your 2017 snapshot. (1 pt)
compare and contrast the anchor text for the articles in the collection: are we gaining or losing terms? (1

Problem 5 continued on next page. . .

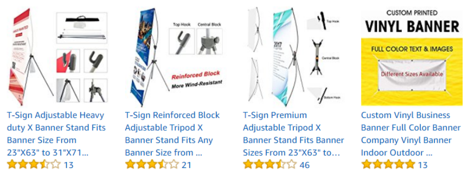

T-Sign Adjustable Heavy duty X Banner Stand Fits Banner Size From 23"X63" to 31"X71...
13
T-Sign Reinforced Block Adjustable Tripod X Banner Stand Fits Any Banner Size from ...
21
T-Sign Premium Adjustable Tripod X Banner Stand Fits Banner Sizes From 23"X63" to...
46
Custom Vinyl Business Banner Full Color Banner Company Vinyl Banner Indoor Outdoor ...
13

Figure 8: Similar items recommended

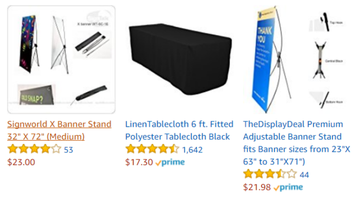

Signworld X Banner Stand 32" X 72" (Medium)
53
$23.00
LinenTablecloth 6 ft. Fitted Polyester Tablecloth Black
1,642
$17.30 prime
TheDisplayDeal Premium Adjustable Banner Stand fits Banner sizes from 23"X 63" to 31"X71")
44
$21.98 prime

Figure 9: Dffierent items recommended

pt)
compare and contrast the sizes of the same pages (i.e., 2008 vs. 2017) in: (2 pts)
- bytes
- total terms
- total unique terms
for all of the pages (i.e., treat each collection as a single unit), compare the 2008 vs. 2017 snapshots in: (2 pts)
- bytes
- total terms
- total unique terms

Solution 5:

To Download the HTML pages and Get disambiguation pages I used:

1. getStatusCodesStatForURLs(): I extracted the wikipedia address from the filenames. For example, the file "en/articles/1/8/1/1810_in_Australia_7eb1.html" points to wikipedia page: "`https://en.wikipedia.org/wiki/1810_in_Australia`"

2. getDisambiguationPages(): I identified disambiguation pages by adding _(disambiguation) to the URLs that returned 200. If a 200 returned a disambiguation page, it means the page has been split. There were X of these. If the page returned a 404 but a 200 for disambiguation page, it means the page has been merged. There were Y of these

The file: *wiki-url-status-codes.txt* has all the status codes

Table 1: HTTP status returned

Status	Url Count
200	5614
404	420
301t	9

Table 2: 2008 and 2017 Versions

	2008 Version	2017 Version
Bytes	2008-141069412	2017-75641915
Terms	2008-539321	2017-6342295
Unique terms	2008-75515	2017-808140

Bytes:

For the 2008 version I checked the size the text data utf8len(), the size on disk was 162,556,332, but the corpus folder also had other content. countBytesLive(): For the 2017 version I made http head request and accumulated the content-length:
I used sklearn CountVectorizer to create a dictionary with key as the term in the collection and value as the frequency of occurrence in the corpus. See Table 2 for the number of unique terms and bytes. I believe the 2008 version is larger because the collection was modified: some of the fields were changed. For example URLs where rewritten, this may have inflated the collection size

Problem 6

11.5. How many papers dealing with term dependency can you find in the SIGIR proceedings since 2000? List their citations

Solution 6:

Paper	**Citations**
Two-stage query segmentation for information retrieval	49
Capturing term dependencies using a language model based on sentence trees	94
A comparison of retrieval models using term dependencies	22
Learning concept importance using a weighted dependence model	157

Problem 6 continued on next page. . .

Automatic P hrase Indexing for Document Retrieval: An Examination of Syntactic and Non-Syntactic Methods	169
[PDF] Capturing term dependencies using a sentence tree based language model	20
Looking inside the box: Context-sensitive translation for cross-language information retrieval	16
Exploring evidence aggregation methods and external expansion sources for medical record search	16
An effective approach to verbose queries using a limited dependencies language model	19
Modeling higher-order term dependencies in information retrieval using query hypergraphs	54
A quasi-synchronous dependence model for information retrievall	35
Modelling term dependence with copulas	8
Parameterized concept weighting in verbose queries	100
Modeling term dependencies with quantum language models for IR	21
The role of knowledge in conceptual retrieval: a study in the domain of clinical medicine	73
Learning to reweight terms with distributed representations	35

Problem 6 continued on next page. . .

[HTML] A dimensional retrieval model for integrating semantics and statistical evidence in context for genomics literature search	39
Word embedding based generalized language model for information retrieval	22
[PDF] Weighted Dependence Model," Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010). 31 citations.âĂć âĂę	22

References

[1]

[2] Evaluation. http://www.cs.sfu.ca/CourseCentral/456/jpei/web Accessed: 2017-12-03.

[3] Githubgist. https://gist.github.com/bwhite/3726239. Accessed: 2017-12-03.

[4] Hits. https://networkx.github.io/documentation/networkx-1.9.1/reference/generated/networkx.algorithms.link_analysis.
Accessed: 2017-12-14.

[5] Information Retrieval in Practice. http://ciir.cs.umass.edu/downloads/SEIRiP.pdf. Accessed: 2017-10-10.

[6] Pagerank. hhttps://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.link_an
Accessed: 2017-12-14.

[7] Quora. https://www.quora.com/What-are-industry-applications-of-the-K-nearest -neighbor-algorithm. Accessed: 2017-12-03.