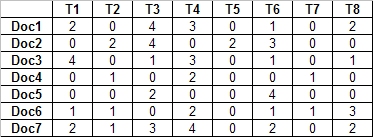
**Intelligent Information Retrieval**   
**CSC 575**

**Assignment 4  
Due: Thursday, March 10**

**Rong Zhuang**

1. Consider the following document-term matrix containing raw term frequencies (See [**Excel Spreadsheet**](http://facweb.cs.depaul.edu/mobasher/classes/csc575/assignments/Assign4.xlsx)).



* 1. Construct a term-term distance matrix using ***Euclidean distance***as the measure.

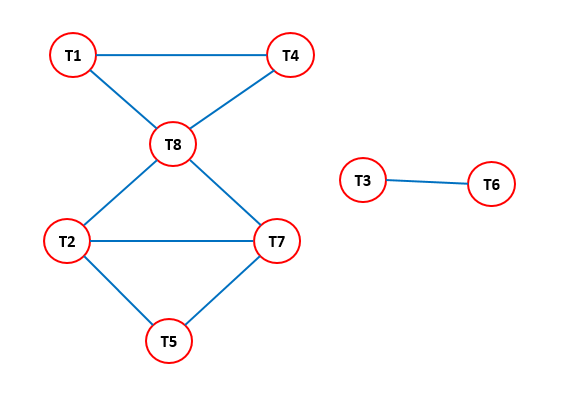
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **T1** | **T2** | **T3** | **T4** | **T5** | **T6** | **T7** | **T8** |
| **T1** | 0.00 | 5.10 | 5.92 | 3.32 | 5.39 | 5.92 | 5.00 | 3.61 |
| **T2** | 5.10 | 0.00 | 5.57 | 5.75 | 1.73 | 4.58 | 2.24 | 3.87 |
| **T3** | 5.92 | 5.57 | 0.00 | 5.83 | 5.83 | 4.00 | 6.93 | 5.83 |
| **T4** | 3.32 | 5.75 | 5.83 | 0.00 | 6.78 | 6.48 | 6.00 | 3.74 |
| **T5** | 5.39 | 1.73 | 5.83 | 6.78 | 0.00 | 4.90 | 2.45 | 4.69 |
| **T6** | 5.92 | 4.58 | 4.00 | 6.48 | 4.90 | 0.00 | 5.66 | 5.48 |
| **T7** | 5.00 | 2.24 | 6.93 | 6.00 | 2.45 | 5.66 | 0.00 | 3.74 |
| **T8** | 3.61 | 3.87 | 5.83 | 3.74 | 4.69 | 5.48 | 3.74 | 0.00 |

\*Refer to python program file hw4-1.py

* 1. Determine the binary term relationship matrix using a distance threshold of 4.00, and then construct the graph corresponding to this matrix. **Note: there is an edge between each pair of nodes whose Euclidean distance is less than or equal to 4.00**.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **T1** | **T2** | **T3** | **T4** | **T5** | **T6** | **T7** | **T8** |
| **T1** | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| **T2** | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| **T3** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **T4** | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| **T5** | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **T6** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **T7** | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| **T8** | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |

\*Refer to python program file hw4-1.py



* 1. Using the above graph, perform clustering based on the **Clique algorithm**.

{T1, T4, T8}, {T2, T7, T8}, {T2, T5, T7}, {T3, T6}

* 1. Using the above graph, perform clustering based on the **Single Link algorithm**.

{T1, T2, T4, T5, T7, T8}, {T3, T6}

1. Consider again the same document-term matrix for the previous problem. In this problem we focus on clustering documents.
   1. Starting with documents 1 and 2 in Cluster 1, and with documents 5, 6 and 7 in Cluster 2, perform K-means clustering of the documents using K = 2 (show the reallocation of the documents to clusters after each iteration). Use the ***cosine similarity*** measure in the clustering algorithm.

|  |
| --- |
| Start kmeans clustering based on cosine similarity:  cluster1: 1, 2  cluster2: 5, 6, 7  cluster1: 1, 2, 5  cluster2: 3, 4, 6, 7  cluster1: 2, 5  cluster2: 1, 3, 4, 6, 7  cluster1: 2, 5  cluster2: 1, 3, 4, 6, 7  Not changed anymore, cluster result finalized |

\*Refer to python program file hw4-2a.py

The two final clusters are: {1,3,4,6,7}, {2,5}.

* 1. Perform clustering of documents using the one-pass (single-pass) technique (show your work). Use **dot product** as your similarity measure, and use a pre-specified similarity threshold of 10. Allow for overlapping clusters if an item satisfies the similarity threshold for more than one cluster. The algorithm for this techniques was given [**lecture notes on clustering**](http://facweb.cs.depaul.edu/mobasher/classes/csc575/Lectures/Clustering.pptx). For an example of how this technique is applied see [**single-pass.html**](http://facweb.cs.depaul.edu/mobasher/classes/csc575/assignments/single-pass.html).

|  |
| --- |
| Start single pass clustering based on dot product:  cluster1: 1, 3, 4, 6, 8  cluster2: 2  cluster3: 5  cluster4: 7 |

\*Refer to python program file hw4-2b.py

1. Read the paper [**Hyperlink Analysis for the Web**](http://facweb.cs.depaul.edu/mobasher/classes/csc575/papers/hyperlink.pdf) by Monika Henzinger, Research Director at Google, and answer the following questions (in each case, briefly discuss your answers or provide short explanations).
   1. What are the basic assumptions behind the hyperlink analysis algorithms?

Assumption 1. A hyperlink from page A to page B is a recommendation of page B by the author of page A.

Assumption 2. If page A and page B are connected by a hyperlink, then they might be on the same topic.

* 1. What is meant by connectivity-based ranking, and how does it relate to co-citation analysis?

We model a collection of Web pages as a link graph when ranking them. Each web page is a node in this graph, and the direct edge between two node represents page A contains a hyperlink to page B. This is so-called Connectivity-based ranking.

In an undirected co-citation graph, nodes A and B are connected by an undirected edge if and only if there exists a third page C hyperlinking to both A and B. We say that A and B are co-cited by C.

The link graph has been used for ranking, finding related pages, and solving various other IR problems. The co-citation graph has been used for categorizing and finding related pages.

* 1. What are some of the primary applications of hyperlink analysis in Web information retrieval?

Search-by-Example, mirrored hosts, web page categorization, geographical scope.

* 1. Briefly describe the PageRank algorithm used by the Google search engine.

The PageRank algorithm evaluates the page ranking by calculating the citation of this page. The more hyperlinks pointing to a page, the better the page. The PageRank of a page A depends on the PageRank of a page B pointing to A. PageRank algorithm is better than the classic word count algorithm which can easily be manipulated.

* 1. Briefly describe the HITS algorithms. What are some of the problems associated with this algorithm, and how might one resolve these problems?

HITS algorithm is to determine good hubs and authorities. Given a user query, the algorithm iteratively computes hub and authority scores for each node in the neighborhood graph, and then ranks the nodes by those scores. Nodes with high authority scores should be good authorities, and nodes with high hub scores should be good hubs.

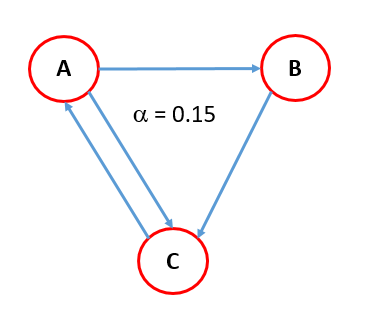
There are two problems with the HITS algorithm:

* It is relatively easy to manipulate scores by adding edges to a few nodes, which can change the resulting hubs and authority scores considerably.
* The topic drift problem: the top-ranked authority and hub pages might be on the different topic. Adding weights to edges based on text in the documents or their anchors alleviates this problem considerably.
  1. How can one spam hyperlink-based search engines? Are they subject to spamming attacks as keyword based approaches? If so, can you think of some approaches to reduce or counter such attacks?

Spam by creating huge number of hyperlinks pointing to a specific page to boost the ranking of this page.

Yes, some so-called Web-positioning companies make money by advising their clients how to manipulate search engine rankings.

Web search engines need to constantly invent and update their ranking algorithms. And keep them secret without any disclosure.

1. **Hyperlink Analysis.**  
   1. Consider the following Web pages and their linkage structure:  
        
      **Page A has links to pages B and C  
           Page B has a link to Page C  
           Page C has a link to Page A**Run the PageRank algorithm on this subgraph of pages (see slides 51 and 52 of [**Lecture 8**](http://facweb.cs.depaul.edu/mobasher/classes/csc575/lectures/lecture8.ppt) for the algorithm and an example). Assume that **** = 0.15**. Simulate the algorithm for three iterations and show PageRank scores twice for each iteration, both before and after normalization.  
      

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A** | **B** | **C** |
| **A** | 0 | 0 | 1 |
| **B** | 1 | 0 | 0 |
| **C** | 1 | 1 | 0 |

|  |
| --- |
| Calculate page rank:  iteration 1:  before norm: 0.38 0.22 0.55  after norm: 0.33 0.19 0.48  iteration 2:  before norm: 0.53 0.22 0.41  after norm: 0.46 0.19 0.35  iteration 3:  before norm: 0.40 0.28 0.47  after norm: 0.35 0.24 0.41 |

\*Refer to python program file hw4-4a.py

* 1. Consider the following Web pages and their linkage structure:  
       
     **Page A has links to pages C, D, and E  
          Page B has a link to Page D and E**Run the HITS (Hubs and Authorities) algorithm on this subgraph of pages (see slides 34 and 35 of[**Lecture 8**](http://facweb.cs.depaul.edu/mobasher/classes/csc575/lectures/lecture8.ppt) for the algorithm and an example). Simulate the algorithm for three iterations and show the hub and authority scores (represented as two vectors over A, B, C, D, E) twice for each iteration, both before and after normalization.

|  |
| --- |
| Executing hits algorithm:  iteration 1:  before norm:  auth: 0.00 0.00 1.00 2.00 2.00  hub: 3.00 2.00 0.00 0.00 0.00  after norm:  auth: 0.00 0.00 0.33 0.67 0.67  hub: 0.83 0.55 0.00 0.00 0.00  iteration 2:  before norm:  auth: 0.00 0.00 0.83 1.39 1.39  hub: 1.67 1.33 0.00 0.00 0.00  after norm:  auth: 0.00 0.00 0.39 0.65 0.65  hub: 0.78 0.62 0.00 0.00 0.00  iteration 3:  before norm:  auth: 0.00 0.00 0.78 1.41 1.41  hub: 1.69 1.30 0.00 0.00 0.00  after norm:  auth: 0.00 0.00 0.37 0.66 0.66  hub: 0.79 0.61 0.00 0.00 0.00 |

\*Refer to python program file hw4-4b.py