Linear Algebra – MT1004
Linear Algebra in Information Retrieval and PageRank Algorithm

Group Members

Ali Nadir 20K0325

Imtiaz Ali 20K0313

Ahmed Abdullah 20K0470

Teacher Name: Amber Sheikh

Submitted: Friday, December 17, 2021

Abstract

Our motivation was to fully attempt to explore the various intricacies and functions of web search engines, primarily to observe the purpose of linear algebra in computer science. What followed was thorough research on two fundamental components of web searching, namely Information Retrieval and Ranking of Web Pages. This report covers various aspects of linear algebra in web search as detailedbelow:

Contents

Representation of web and its documents:

Internet as a graph:

- The internet can be considered a large collection of documents, called web pages
- In terms of graph theory, such web pages from individual nodes
- Nodes must contain at least one link to each other
- These links are represented by directional arrows, which shows how a web page links to one or more different pages
- Ultimately, the internet can be represented as a directed graph
- Each node represents a page
- Links represent ordered pair [u,v] where u and v are pages
- **Hyperlinks connect pages which forms basis for principles such as ordering, querying, and ranking for information retrieval**

Modelling information as vectors:

- Web search is entirely dependent on modelling textual data in the form of vectors and matrices consisting of numerical values
- Column vectors can be used to show what sort of content is contained in a particular document/page
- A column vector represents a web page, and the N rows of the Nx1 matrix represent various aspects of information that may be relevant to a query
- Also indicate the likelihood or probability of matching information to a search
- Let a vector represent a page P containing 3 key terms
- Each term constitutes a row of the column P
- The value of the row represents the composition of P in terms of a key word
- Value is always in the range 0<v<1
- Value is also known as the weighted percentage composition of the page P
- $\mathbf{P} = \begin{matrix} 1 \\ (0) \\ 0 \end{matrix}$, where P contains 100% weight/info on Key1, 0% about Key2, and 0% on Key3
- All N webpages will entirely consist of those M terms and will lie in the vector space R^n
- Suppose there are only 7 pages labeled $\boldsymbol{d_1}$, $\boldsymbol{d_2}$, $\boldsymbol{d_3}$, $\boldsymbol{d_4}$, $\boldsymbol{d_5}$, $\boldsymbol{d_6}$, $\boldsymbol{d_7}$, in the *web* database and only the three keywords (desserts, breads, vegetables) to distinguish article topics. Then, the matrix **A** capturing the data for *web* documents is

- d_1 d_2 d_3 d_4 d_5 d_6 d_7

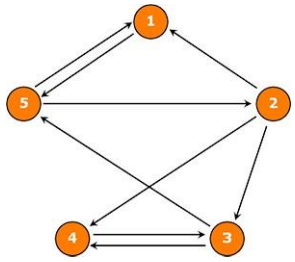
- $\begin{pmatrix} 1 & 0 & 0 & 0.2 & 0.65 & 0.5 & 0.9 \\ 0 & 1 & 0.7 & 0.4 & 0.35 & 0.5 & 0.1 \\ 0 & 0 & 0.3 & 0.4 & 0 & 0 & 0 \end{pmatrix}$

 return multiple pages relating to the query's content

 Such results would need to be displayed in some order, particularly with respect to how relevant a webpage is to the search

 As the web is understood to be a connected digraph of such pages, an edge, or link, of the graph would show that page A is giving importance to page B. This would result in page A recommending/citing page B.

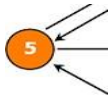
 Considering all such webpages would be recommended by at least one other page, some pages will have more relevance to the search than others, thus paving way for hyperlink analysis

- 

Page 5 authorizes 1 and 2

- Consider Page 5, which has 2 outgoing links to Pages 1 and 2
- Thus Page 5 is giving authority to those pages
- The authority received by Page 1 and 2 will be weighted with respect to the total out degrees of Page 5
- Thus, page 1 and 2 only receive **½** of the importance (1) of page 5
- These pages can be combined to form a link matrix

- The square link matrix for the graph can be shown as:
- The **MxN** matrix represents the **weighted links** of pages
- The **column space** spans **the vector space representing outgoing links**(citations)
- 

- The **row space** spans **the vector space representing incoming links**(citations)
- 

 Searching on the internet is a completely random process, where users can click any page unpredictably.

```
1/3
 0
1/3
1/3
 0
```

 In that respect, the reason for **normalization of the vectors** is to estimate the **probability** of a user clicking on a particular webpage, relative to other pages within its **network**.

 Probability of a visit is dependent on the last/current page accessed. Thus

 For page 5, the user will have an equal probability of **½** of visiting other pages from the source page since page 5 only links to 2 of the 4 available pages in its network

 Thus, each **vector representing a page in the (inter)network forms a <u>probability vector</u>**, whose entries will always sum up to 1 (shows probabilities of visit)

 For page 2, there is an equal probability that pages 1, 3, and 4 will be accessed at any given time (importance is shared)

<u>PageRank Algorithm:</u>

 We are interested in the continuous transformation of the link matrix based on these probability vectors

 This matrix transformation will result in some stable vector denoting the PageRanks of all pages in the graph

 As the link matrix entirely consists of such probability vectors, it can be said that the link matrix of the network is a **column stochastic matrix, or the transition probability matrix of the network**

- Note that:
- For unbalanced adjacency matrix A, multiplying A by itself results in A^2, which indicates number of paths from page I to page j (Pij) having distance of 2 (2 distinct paths from i to j)
- Repeated multiplication of the adjacency matrix A tells how the user can ***transition*** from one page i to another page j, and in how many ways
- **Entries of A grow largest for the pages with the most hyperlinks**
- Google's PageRank algorithm views this network as a **dynamical system** to conclude its long term behavior in terms of page probabilities which is its rank
- Instead of paths, transformation now updates the probabilities
- The increase or decrease of a pages probability is done by repeatedly multiplying the transition matrix by itself
- After certain number of transitions, equilibrium is achieved for a steady state vector $\boldsymbol{\pi}$

- Let P be transition probability matrix
- Formally speaking the entries of **P**, or P_{ij}, tell the probability of going to next state/page
- For all i,

$$\sum_{j=1}^{n} P_{ij} = \mathbf{1}$$

- **<u>If a node has K>0 out-links:</u>**
- The user teleports to random state with probability $\mathbf{0} < a < \mathbf{1}$, $\mathbf{P} = a/\mathrm{N}$
- A normal traversal will have probability $(\mathbf{1} - a)/\mathbf{K}$

- Consider $a = \mathbf{0.5}$ and let transition matrix be P
- The matrix will now have to be multiplied by scalar $(\mathbf{1} - a)$ for normalization since it is the probability of a normal walk
- Normalized row vector:

$$P[1,*] = (1-\alpha)\ (0\ 1\ 0) + \alpha\ (1/N\ 1/N\ 1/N)$$

- To account for teleportation, add α/N to every entry of the weighted adjacency matrix to obtain transition probability matrix P.
- Resultant transition matrix for $\alpha = 0.5$:
- Now we can proceed to calculation of PageRank

$$P_{\alpha=0} = \begin{pmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix}$$

- Initial weighted transition matrix for markov chain

$$P_{\alpha=0.5} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

- Resultant normalized transition probability matrix

Calculation:

- Power Method:
- Repeat this iteration until vector **converges** i.e. becomes the steady state vector π
- Evaluation of linear system
- Order is always **query independent**
- **Can be incorporated with information retrieval algorithms like those based on previously discussed vector space model (VSM) algorithms**

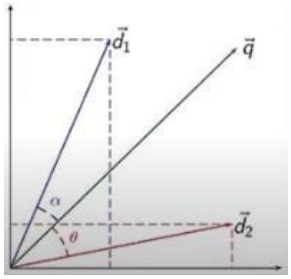
Introduction to vector space model:

- **Vector space models** are to consider the relationship between data that are represented by vectors. It is popular in information retrieval systems but also useful for other purposes. Generally, this allows us to compare the similarity of two vectors from a geometric perspective.
- A vector space is a mathematical term that defines some vector operations. In layman's term, we can imagine it is a n-dimensional metric space where each point is represented by a n-dimensional vector. In this space, we can do any vector addition or scalar-vector multiplications.

Definition and uses

- The Vector-Space Model (VSM) for Information Retrieval represents documents and queries as vectors of weights. Each weight is a measure of the importance of an index term in a document or a query, respectively.

- **USES**
- information filtering
- information retrieval
- indexing
- relevancy rankings



More on VSM:

- As it is used in information retrieval to determine which document **d1** or **d2** is more similar to given query **q**.
- Note that documents and queries are represented in the same space.
- The angle between the two vectors (cosine similarity) is used as a proxy for the similarities of underlying documents.

Cosine Similarity

- The cosine measured is computed as the normalized dot product of the two vectors:

$$\sigma(D,Q) = \frac{|D \cap Q|}{\sqrt{|D||Q|}} = \frac{\sum(d_i q_i)}{\sqrt{\sum(d_i)^2}\sqrt{\sum(q_i)^2}}$$

- A variant of cosine is the Jaccard coefficient:

$$\sigma(D,Q) = \frac{|D \cap Q|}{|D \cup Q|}$$

Latent Semantic Indexing:

- **Latent semantic analysis** (**LSA**) is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.
- **Main Idea**

- Map each documents into some concepts.
- Map each term into some concepts.

- **Concept :-** A set of terms, with weight
- LSI can decompose our initial matrix into two smaller matrix.

- Term-concept matrix
- Document-concept matrix
- The ‘1’s represent the weights of a concept in both the tables
- **Question: How to search for “system”?**
- **Answer: Find the corresponding concepts and its documents (works like an automatic constructed thesaurus).**

Benefits of LSI

- To derive concepts from documents.
- Works as an automatic thesaurus.
- Reduce dimensionality down to fewer concepts. (Matrix decomposition)

More benefits of LSI: -

- LSI is not restricted to working only with words. It can also process arbitrary character strings. Any object that can be expressed as text can be represented in an LSI vector space. For example, tests with MEDLINE abstracts have shown that LSI is able to effectively classify genes based on conceptual modeling of the biological information contained in the titles and abstracts of the MEDLINE citations.
- Text does not need to be in sentence form for LSI to be effective. It can work with lists, free-form notes, email, Web-based content, etc. As long as a collection of text contains multiple terms, LSI can be used to identify patterns in the relationships between the important terms and concepts contained in the text.

How does LSI work (SVD):

- LSI uses SVD (single value decomposition) which is mathematic tool. It finds concepts in matrices. Moreover, SVD can solve through finding these concepts we can ran these concepts and throw away the less important data, so that data is stored more efficiently.
- It is a generalized technique and can manipulate any kind of data, All it takes is a matrix in rows and columns regardless of the content for eg:-Netflix/movies and grocery/items etc.

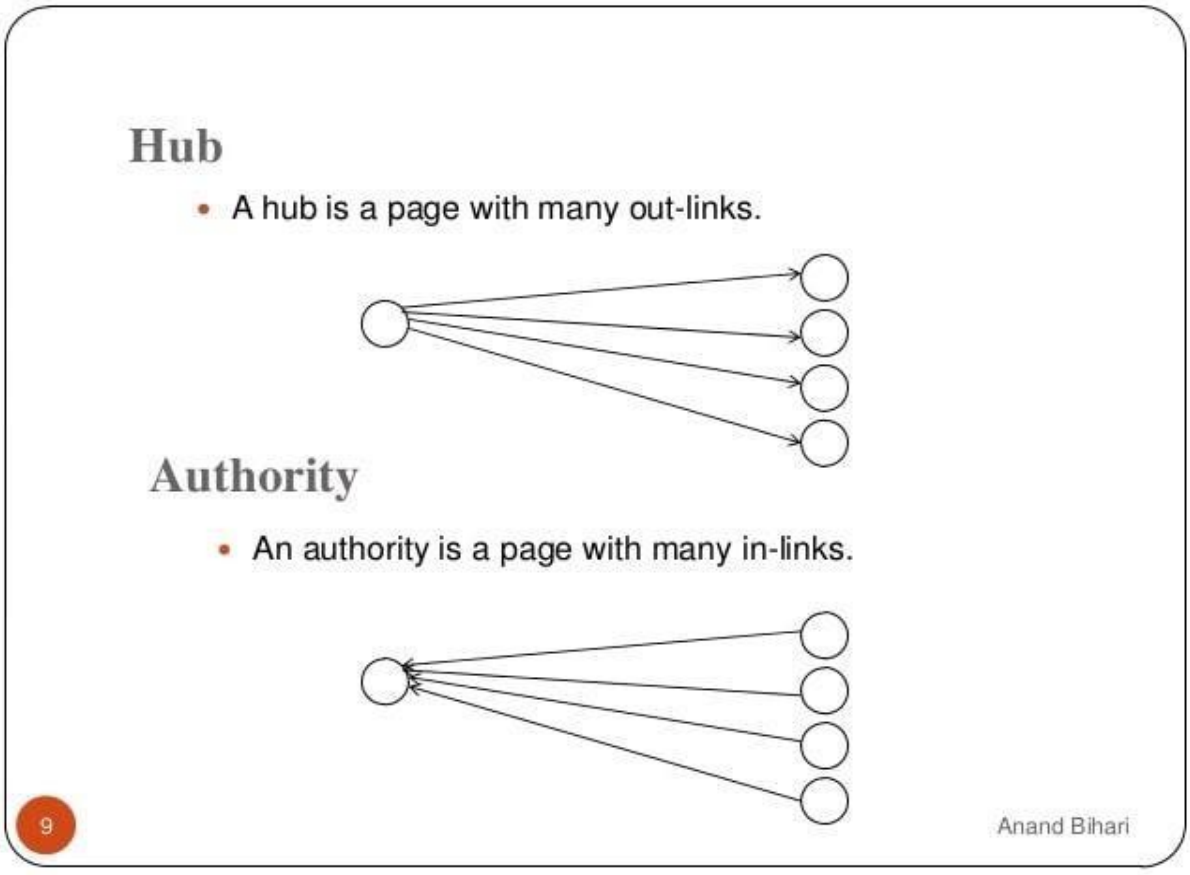
Link Analysis:

What Is Link Analysis?

- Link analysis is a data analysis technique used to evaluate relationships (connections) between nodes. Relationships may be identified among various types of nodes (objects), including organizations, people and transactions
- Normally, the nodes represent specific data points, and the links represent the connections between them
- Main use of link analysis is for extracting or analyzing huge pieces of data/information . It is used in all professional organizations .
- Some of the basic uses of link analysis are:
- To find matches in data for known patterns of interest
- Find anomalies where known patterns are violated
- Discover new patterns of interest.(used heavily for data mining)
- Link analysis has been used for investigation of criminal activity (fraud detection, counterterrorism, and intelligence) to help police know the previous history of criminals etc
- Its used in computer security analysis, search engine optimization (also used in google to provide user relevant information from huge chunks of data) market research(data mining), medical research, art etc.

Hyperlink Induced Topic Search (HITS) Algorithm:

- **Hyperlink Induced Topic Search** (HITS) Algorithm is a Link Analysis Algorithm that rates webpages, developed by Jon Kleinberg. This algorithm is used to the web link-structures to discover and rank the webpages relevant for a particular search.
- HITS uses hubs and authorities to define a recursive relationship between webpages.
- Given a query to a Search Engine, the set of highly relevant web pages are called **Roots**. They are potential **Authorities**.
- Pages that are not very relevant but point to pages in the Root are called **Hubs**. Thus, an Authority is a page that many hubs link to whereas a Hub is a page that links to many authorities.



Algorithm:

 The algorithm performs a series of iterations, each consisting of two basic steps:

 Authority update: Update each node's *authority score* to be equal to the sum of the *hub scores* of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.

 Hub update: Update each node's *hub score* to be equal to the sum of the *authority scores* of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

 Start with each node having a hub score and authority score of 1.

 Run the authority update rule

 Run the hub update rule

 Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores

References

- linear algebra - Why is PageRank an eigenvector problem? - Mathematics Stack Exchange
- The Linear Algebra Behind Search Engines - Introduction | Mathematical Association of America (maa.org)
- Search engines and linear algebra - Nibcode Solutions
- [PDF] The Use of the Linear Algebra by Web Search Engines | Semantic Scholar
- Web Search Algorithms and PageRank (wlu.ca)
- PageRank Algorithm - The Mathematics of Google Search (cornell.edu)