Linear Algebraic Models in Information Retrieval

Nathan Pruitt and Rami Awwad

December 12th, 2016
Information Retrieval—Defined as finding relevant information to a search in a database containing documents, images, articles, etc.

Practical real life example—Finding an article or book in a library through catalog system or through library’s database via search engine

Most common type are internet search engines *a la* Google, Yahoo, but also used on many other sites wherever there’s a search feature
S.M.A.R.T. (System for the Mechanical Analysis and Retrieval of Text) developed at Cornell University in the 1960s

Obtains legacy for the development of I.R. models including the vector space model
The Vector Space Model

- A text based ranking model common to internet search engines in the early 1990s
- Works by making a $t \times d$ matrix, where $t$ can represent all terms in an English dictionary
- $d$ representing the number of documents in a search engine database
The Vector Space Model

Each \( m \) given a weight depending on number of times each term \( t \) occurs in document \( d \), then weighed with an arithmetic weighing scheme.

Weight allows comparison between document to document and document to query by the angles between their column vectors.
VSM: A Simpler Example

\[ M_{\text{example}} = \begin{bmatrix} \text{internet} & \text{graph} & \text{directed} \\ doc_1 & 38 & 10 & 0 \\ doc_2 & 14 & 20 & 2 \\ doc_3 & 20 & 5 & 10 \end{bmatrix} \]

\[ \text{Query} = \begin{bmatrix} \text{internet} & \text{graph} & \text{directed} \\ \text{term} & 1 & 1 & 1 \end{bmatrix} \]

- Entries called **term frequencies**
- Term frequencies processed through arithmetic weighing scheme because higher \( tf \) doesn’t necessarily mean a more relevant website
- Engine considers query as a **bag of words**— order of terms eschewed
Length Normalized $t \times d$ Matrix and Query Vector

\[
\text{Query}^* = \begin{bmatrix}
\text{internet} \\
\text{graph} \\
\text{directed}
\end{bmatrix}
\begin{bmatrix}
\frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{3}}
\end{bmatrix}
= \begin{bmatrix}
doc_1 \\
doc_2 \\
doc_3
\end{bmatrix}
\begin{bmatrix}
\text{internet} \\
\text{graph} \\
\text{directed}
\end{bmatrix}
= \begin{bmatrix}
0.790 & 0.630 & 0.659 \\
0.612 & 0.676 & 0.487 \\
0 & 0.382 & 0.573
\end{bmatrix}
\]

- After arithmetic scheme, matrix and query vector are length normalized
- Serves to simplify calculation of angles between document vectors, and between the document vectors and the query
VSM: The "Cosine Similarity"

\[
\cos(\text{doc}1, \text{doc}2) \approx \frac{\begin{bmatrix} 0.790 \\ 0.612 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0.630 \\ 0.676 \\ 0.382 \end{bmatrix}}{\|\text{doc}1\| \|\text{doc}2\|} \approx \frac{0.912}{1} \approx 0.912
\]

\[\cos(\text{doc}1, \text{doc}3) \approx 0.819\]
\[\cos(\text{doc}2, \text{doc}3) \approx 0.963\]
\[\cos(\text{Query}, \text{doc}1) \approx 0.810\]
\[\cos(\text{Query}, \text{doc}2) \approx 0.975\]
\[\cos(\text{Query}, \text{doc}3) \approx 0.993\]

These calculations imply the following angles separate each vector:

\[(\text{doc}1, \text{doc}2) \approx \arccos 0.912 \left( \frac{180^\circ}{\pi} \right) \approx 24.188^\circ\]
\[(\text{doc}1, \text{doc}3) \approx 34.985^\circ\]
\[(\text{doc}2, \text{doc}3) \approx 15.530^\circ\]
\[(\text{Query}, \text{doc}1) \approx 35.901^\circ\]
\[(\text{Query}, \text{doc}2) \approx 12.918^\circ\]
\[(\text{Query}, \text{doc}3) \approx 7.006^\circ\]
VSM: Visualization of Document Vectors and their Shared Angles

**Figure:** Cosine similarity between doc1 to doc2 and doc2 to doc3

**Figure:** Cosine similarity between doc1 and doc3
VSM: Visualization of Document Vectors and their Shared Angles with Query Vector

**Figure:** Cosine similarity between doc2 to the query and doc3 to query

**Figure:** Cosine similarity between doc1 and the query
PageRank Algorithm

- Google’s matrix has over 8 billion row and columns.

This directed graph represents the overall rankings of the websites.
- This is a Markov Chain.
- The arrows represent links between different websites.
- For example, website 1 only links to website 2.
This matrix $P$ shows the probabilities of movement between these websites. Because website 1 only links to website 2, there is a 100 percent chance of that move.

Matrix $P$ is a **transition matrix** because the entries describe the probability of a transition from state $j$ to state $i$.  

$$
P = \begin{bmatrix}
  i_1 & j_1 & j_2 & j_3 & j_4 & j_5 & j_6 & j_7 \\
  i_2 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
  i_3 & 1 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{4} & 0 \\
  i_4 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\
  i_5 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\
  i_6 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 \\
  i_7 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 1 
\end{bmatrix}$$
Notice that each column vector in transition matrix $P$ obtains entries that when added total 1. Therefore, all column vectors in $P$ are probability vectors.

Thus our transition matrix is also a stochastic matrix, which describes a Markov chain with some interesting properties.

One of these properties state that all stochastic matrices have at least one eigenvalue of 1. The eigenvector corresponding to 1 will tell us the rank of our 7 websites, or in Google terms, the PageRank of each website.
To approach this eigenvector, we calculate the **steady-state vector** \( x_n \) of our 7 website chain:

\[
x_n = \begin{bmatrix}
  a_1 \\
  \vdots \\
  a_j \\
  \vdots \\
  a_7
\end{bmatrix}
\]

All stochastic matrices have a steady-state vector. Our \( x_n \) is a probability vector describing the chance of landing on each website after clicking through \( n \) links within our chain.
We use this equation to compute steady-state vectors:

$$
\lim_{n \to \infty} x_n = P_k^n x_0
$$
Adjustment to Transition Matrix

Google is said to use a $p$ with a value of 0.85. Then, we retrieve our $P_k^n$ as follows:

$$P_k^n = 0.85 \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{7} \\ 1 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{7} \\ 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{7} \\ 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 & \frac{1}{4} & \frac{1}{7} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{7} \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{7} \end{bmatrix} + 0.15 \begin{bmatrix} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{bmatrix}$$

$$= \begin{bmatrix} 0.02142 & 0.02142 & 0.02142 & 0.44642 & 0.02142 & 0.02142 & 0.14285 \\ 0.87142 & 0.02142 & 0.44642 & 0.02142 & 0.44642 & 0.23392 & 0.14285 \\ 0.02142 & 0.30476 & 0.02142 & 0.02142 & 0.02142 & 0.23392 & 0.14285 \\ 0.02142 & 0.30476 & 0.44642 & 0.02142 & 0.02142 & 0.23392 & 0.14285 \\ 0.02142 & 0.02142 & 0.44642 & 0.02142 & 0.23392 & 0.23392 & 0.14285 \\ 0.02142 & 0.30476 & 0.02142 & 0.02142 & 0.44642 & 0.23392 & 0.14285 \\ 0.02142 & 0.02142 & 0.02142 & 0.02142 & 0.23392 & 0.23392 & 0.14285 \\ 0.02142 & 0.02142 & 0.02142 & 0.02142 & 0.23392 & 0.23392 & 0.14285 \end{bmatrix}$$
Final Rank

\[
\lim_{n \to 75} \mathbf{x}_n = \begin{bmatrix}
0.02142 & 0.02142 & 0.02142 & 0.44642 & 0.02142 & 0.02142 & 0.142857 \\
0.87142 & 0.02142 & 0.44642 & 0.02142 & 0.44642 & 0.23392 & 0.142857 \\
0.02142 & 0.30476 & 0.02142 & 0.02142 & 0.02142 & 0.23392 & 0.142857 \\
0.02142 & 0.30476 & 0.44642 & 0.02142 & 0.02142 & 0.23392 & 0.142857 \\
0.02142 & 0.02142 & 0.02142 & 0.44642 & 0.02142 & 0.23392 & 0.142857 \\
0.02142 & 0.30476 & 0.02142 & 0.02142 & 0.02142 & 0.23392 & 0.142857 \\
0.02142 & 0.02142 & 0.02142 & 0.02142 & 0.02142 & 0.23392 & 0.142857 \\
\end{bmatrix}^n \begin{bmatrix}0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]

\[
\mathbf{x}_n = \begin{bmatrix}0.104631 \\ 0.253767 \\ 0.100953 \\ 0.177828 \\ 0.138598 \\ 0.159857 \\ 0.063021 \end{bmatrix}
\]
1 Christopher D. Manning, Prabhankar Reghaven, Hinrich Schutze, Introduction to Information Retrieval
2 Michael W. Berry, Zlatko Drmac, Elizabeth R. Jessup. Matrices, Vector Spaces, and Information Retrieval
3 Raluca Tanase, Remus Redu. The Mathematics of Web Search
5 Howard Anton, Robert C. Busby. Contemporary Linear Algebra