REPORT ON DR. CABRAL’S LECTURE

ARNOLD YANG

Lecture: “Mathematics of Search Engine,” by Dr. Cabral

1. SUMMARY

This lecture covers the topics of web crawling, page rank, HITS, and SALSA.

1.1. Content Index. The basic principle is “a webpage is important if it is pointed to by other important pages.” In the search engine Google, the ranking is done before you even do anything. This means Google’s search engine is query independent. It uses a uniform distribution of importance.

Let a node refer to a webpage. Suppose we have four nodes, $N_1, N_2, N_3,$ and $N_4$. Let $N_1$ point to $N_2$ and $N_3$; $N_2$ point to $N_4$; $N_3$ point to $N_1, N_2,$ and $N_4$; and $N_4$ point to $N_1, N_2,$ and $N_3$. Let $x_n$ be the importance of $N_n$, which is determined by a uniform distribution of importance of the nodes that point to $N_n$.

Then you can see that for our particular example, $x_1 = \frac{1}{3}x_3 + \frac{1}{3}x_4$. (Since $N_3$ points to three nodes, $N_1$ receives $\frac{1}{3}$ of the importance of $N_3$. Likewise, since $N_4$ points to three nodes, $N_1$ receives $\frac{1}{3}$ of the importance of $N_4$.)

Considering the importance of all four nodes, we get

\[
\begin{align*}
    x_1 &= \frac{1}{2}x_1 + \frac{1}{3}x_3 + \frac{1}{3}x_4 \\
    x_2 &= \frac{1}{2}x_1 + \frac{1}{2}x_3 + \frac{1}{3}x_4 \\
    x_3 &= \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_4 \\
    x_4 &= \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3
\end{align*}
\]

Now that we have a system of equations, we can use a matrix to find the solutions.

\[
\begin{pmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4
\end{pmatrix} = \begin{pmatrix}
    0 & 0 & \frac{1}{3} & \frac{1}{3} \\
    \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{3} \\
    \frac{1}{2} & 0 & 0 & \frac{1}{3} \\
    0 & 1 & \frac{1}{3} & 0
\end{pmatrix} \begin{pmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4
\end{pmatrix}
\]

The $4 \times 4$ matrix is a hyperlink matrix. Now, we want to look for the eigenvector. By linear algebra, this problem is solved.

Let the ranking vector $x$ be the eigenvector of eigenvalue 1. Let $H$ be the hyperlink matrix, then $Hx = x$. The idea is to start with any $x_0$.

\textit{Definition 1.1.}

\[
x_k = Hx_{k-1} = H^kx_0.
\]

The question in which we are now interested is “when does $x_k$ converge?”
Definition 1.2. Let $A$ be an $n \times n$ matrix with complex or real elements with eigenvalues $\lambda_1, \ldots, \lambda_n$. Then the spectral radius $\rho(A)$ of $A$ is

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|.$$ 

Theorem 1.1 (Perron-Frobenius). Let $A$ be an $n \times n$ matrix, and suppose $A$ is irreducible and non-negative. Then

(i) $\rho(A) > 0$,
(ii) $\rho(A)$ is an eigenvalue of $A$,
(iii) there exists a positive vector $x$ such that $Ax = \rho(A)x$, and
(iv) $\rho(A)$ is an algebraically simple eigenvalue of $A$.

1.2. Probabilistic Interpretation. Recall our example of a 4 node network. The hyperlink matrix is

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 1 & \frac{1}{3} & 0 \end{pmatrix}.$$ 

To see if this matrix converges, take powers of this matrix. For a matrix to be irreducible, we want to know if the graph is connected and if we can eventually get to every single node of the network. $H^2$ means “can I go from 2 to 1 in 2 steps?”

1.3. Possible Problems.

1.3.1. Dangling Nodes. A node that does not point to any other node is a dangling node. Consider a two node system in which only $N_1$ points to $N_2$. Then

$$H = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$ 

This column of zeros is problematic for two reasons. Firstly, the determinant of $H$ would be 0. But more importantly, $H$ would not be column stochastic in our probabilistic approach to the problem. Once you get to the dangling node, you have nowhere to go.

We will return to this problem in a moment with Google’s solution.

1.3.2. Cycle. Suppose we have a three node system in which $N_1$ points to $N_2$, which points to $N_3$, which points back to $N_1$. Then

$$H = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad H^2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad H^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$ 

We see that $H$ never converges. This has some importance. If there is a block of zeros in $H$, then you will get stuck in a subcycle within the network.
1.4. Brilliant Solution. Returning to the problem of the dangling node, Google addresses this problem by modifying the paths of the nodes - a crucial step in its link analysis algorithm called PageRank.

Definition 1.3. Let $H$ be the hyperlink matrix and $e$ be a matrix of all ones. Then

$$S = H + \frac{ee^T}{n}.$$  

Once a dangling node is reached, Google will artificially create paths from that dangling node to every node, including the dangling node itself, such that there is the same probability to travel every path. Recall our previous example of a two node system in which only $N_1$ points to $N_2$ - $N_2$ is the dangling node. Google artificially creates paths from $N_2$ to itself and to all the other nodes, in this case, to $N_1$. Then the hyperlink matrix

$$H = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

becomes

$$S = \begin{pmatrix} 0 & \frac{1}{2} \\ 1 & \frac{1}{2} \end{pmatrix}.$$  

This is desirable as now the matrix $S$ is column stochastic. There is a total probability of 1 that you will leave a node, instead of getting stuck at a node. Now that we have a column stochastic matrix, we can continue to calculate the PageRank.

Definition 1.4. Let $\alpha \in (0,1)$, $\alpha \in \mathbb{R}$. Then the Google matrix $G$ is given by

$$G = \alpha S + (1 - \alpha) ee^T.$$  

The Google matrix basically does the follow: flip a coin; if heads, then $\alpha$ will be more significant, and you follow the natural structure and flow of the website; if tails, then $(1 - \alpha)$ will be more significant, and you can go anywhere in the world wide web randomly.

The order of convergence depends on $\alpha$. The smaller the $\alpha$, the quicker the convergence. Once a month, Google gets this matrix by doing this calculation. $G$ is primitive and irreducible. It is reported that Google uses $\alpha = 0.85$.

Google raises $G$ to a huge power, say 2,000, then multiplies it by 10,000 to see the results even better. All columns will be the same so you only need to consider one column. The highest number is the most important webpage, and the lowest number is the least. Google’s PageRank algorithm is very effective at weighing the relevant scores of pages shown in search results.

1.5. Other Algorithms. Hypertext Induced Topic Search (HITS) is a search algorithm used by Teoma.com and Ask.com. It is query-dependent and uses the idea of authorities and hubs in a local neighborhood. Its basic principle is “good authorities are pointed to by good hubs, and good hubs point to good authorities.”

Stochastic Approach for Link-Structure Analysis (SALSA) is a probabilistic approach using Markov chains. SALSA is also query-dependent and is said to be like a “popularity contest.”