Google PageRank Algorithm: Markov Chain Model and Hidden Markov Model

Prerna Rai
CCCT,Polytechnic
South Sikkim

Arvind Lal
CCCT,Polytechnic
South Sikkim

ABSTRACT

In this document, the algorithm behind Google PageRanking and their techniques have been put up. The basic algorithm used by Google, for PageRanking and other applications are Markov model or Markov Chain model and Hidden Markov model. These algorithms are used to search and rank websites in the Google search engine. PageRank is a way of measuring the importance of website pages. Markov chain model and Hidden Markov model is a mathematical system model. It describes transitions from one state to another in a state space. The Markov model is based on the probability the user will select the page and based on the number of incoming and outgoing links, ranks for the pages are determined. HMM also finds its application within Mapper/Reducer. These algorithms are a link analysis algorithm.

Keywords

Markov chain Model, PageRanking, Finite state machine.

1. INTRODUCTION

The largest and widely used search engine on the web today is Google. Every second it is providing services to millions of queries receiving from Internet user. Google organize huge amount of information and make it universally accessible. The search engine aims at returning the page being queried in order of their preference. The first ten pages returned by the search engine are the most visited page and are mostly of importance to the user. The technique used by Google to rank web pages according to their importance is called PageRank, developed by Larry Page and Sergey Brin at Stanford University. PageRank works by counting the number and quality of outgoing links and incoming links to a particular page in order to find out the importance of a page. The Google search engine works with an assumption that websites that receives more links from other websites have higher importance and given higher rank.

Apart from basic PageRank method Google also uses other technique to determine rank of page. They are Markov model which follows Markov chain model and Hidden Markov model. Using these techniques Google computes the score of each web page and based on the score it determines the order of the pages in which it should be listed in search results provided by the search engine. These models Markov model or Markov chain model and Hidden Markov model, are stochastic finite state machine. Stochastic here means random or based on theory of probability. A stochastic process is a process whose behavior is nondeterministic and the system's subsequent state is determined both by the process's predictable actions and by a random elements.

2. RELATED APPROACH

When the user provides query to the Google Search Engine the pages are listed in user interface, in accordance to the importance of individual page. The importance of the page is determined by finding the score of the page. This score can be found using various techniques. And the techniques that have been discussed here are:

1. Google PageRanking.
2. Markov Chain Model
3. Hidden Markov Model.

Google PageRank algorithm as per [1]assumes a probability distribution between 0 and 1.It works by assigning score to number of web pages on world wide web and hence when user request for any page from the search engine, it returns the pages based on the rank of page based on their score. And hence we can find that the user's search is returned with the importance of the pages and most often we find that the first top ten search result provided by Google may be enough for the user request fulfillment.

The other technique used by Google to find the score of a page is by using Markov Model. It uses the concept of Markov chain. It is used to predict the behavior of the system that moves from one state to another. The state space of the model depends on the current state and not on the sequence of events that preceded it.

Hidden Markov Model is a stochastic finite state machine used to solve three problems of encoding, decoding and training. Unlike Markov Model it has hidden and observed states where the observed states are given with the output but the hidden states need to be determined. This model aims at solving the three problem i.e decoding problem is solved using Viterbi algorithm, encoding problem by using the Forward Algorithm and training problem is solved using Baum-Welch algorithm. It also can be used to find the score of the page for returning the page according to their importance.

2.1 Google PageRank Algorithm

The basic algorithm used by Google to perform link analysis is Link analysis algorithm. This algorithm [6] for web search engines determines the relevance of web pages. Among many link analysis algorithm PageRank is one of the mechanism used by Google search engine. PageRank [1] is a query and content independent algorithm. Content independent means the PageRank algorithm does not include the contents of web page for ranking instead it uses the link structure of the web and assigns PageRank score to each web pages on www using mathematical model. When query for a particular page is made by the Internet user, Google search algorithm compiles the PageRank scores using PageRank algorithm and hence return the pages depending upon the score of each page being queried. The basic PageRank algorithm outputs a probability distribution. The transition matrix that outputs the final probability distribution determines the score of each page.

PageRank algorithm considers the web as a directed graph, where nodes represent the pages and edges represent the hyperlinks in the page. PageRank algorithm for Google [14] is developed by page and Brin, and today this algorithm has been the heart and soul of Google search engine. The pages are ranked offline and are content independent. The name PageRank states that the pages in the web are assigned a score which is assigned as a numerical value and for any suitable page P, a PageRank is denoted by PR(P).The rank of page is based on mathematical algorithm based on webgraph, created by world wide web pages as nodes and hyperlinks as edges. This algorithm has its basis on the probability distribution. The score of the page indicates the importance of the page. More the hyperlinks from the other important pages (incoming links) higher will be the score and hence higher the score of the page higher will be the rank of a page.

The PageRank computations [2][14] require several "iterations"(until it converges) and are determined probabilistically. A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.

2.2 Simplified form of Google PageRank Algorithm

There are millions of pages available in the web server and out of all these many pages the search engine needs to return the page to the user based on the user's need. So to do so how does Google respond to the user's query that meets the users need? This is done using page Ranking. The algorithm was first introduced by Page and Brin [1]. During their works they came up with many algorithm to rank the page according to score and this is among one of their findings.

To find out the page score one must consider that the surfer can select any page. It is not always that they select the pages sequentially but there might be a situation where the surfer may click the pages randomly. Though most of the time, a surfer will follow links from a page sequentially, i.e from a page *i* the surfer will follow the outgoing links and move on to one of the neighbors of *i*. But this may not happen always, a smaller, but positive percentage of the time, the surfer will dump the current page and choose arbitrarily a different page from the web and "teleport" there. So to overcome such situation Page and Bring introduced a factor called as the damping factor *d*, that reflects the probability that the surfer drops the current page and "teleports" to a new one. Since he/she can teleport to any web page, each page has 1/n probability to be chosen [1][16].

Therefore the formula is given as below.

In general, to compute page rank PR of page P the formula given by Page and Brin[1][16] is:

$$PR(P) = \frac{1-d}{N} + d[\frac{PR(Pi)}{Oi}] \qquad (1)$$

Here d is a damping factor such that $0 \leq d \leq 1$and O_i is the number of outgoing links of page i. The damping factor d is assumed to be given as values 0.85. In order to explain simplified Google page rank algorithm following assumptions are made [1]:

a. Three web pages: **A**, **B**, **C** are considered

b. Links from a page to itself, or multiple outbound links from one single page to another single page, are ignored.

c. PageRank is initialized to the same value for all pages.

Consider an example of a web pages having following links:

- page **A** had a link to pages **C** and **B**
- page **C** had a link to page **A**, and
- page B had links to all A and C pages.

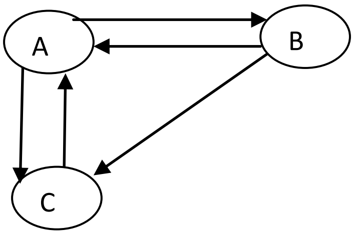

Fig2.2.1Web pages for A, B and C

In this scenario during the first iteration the page rank of A,B,C will be calculated as follows:

PR(A)=0.15/3+.85 [PR(B)/2+PR(C)/1]=0.432

PR(B)=0.15/3+.85[PR(A)/2]=0.233

PR(C)=0.15/3+0.85[PR(A)/2+PR(B)/2]=0.3

This computation continues until page rank gets converged or there do not exist further changes and until the sum value is not equal to 1.

Based on the score it is found that A has highest score and B the least. This is due to the reason that A has two outgoing links and 2 incoming links. B has one incoming and 2 outgoing. C has two incoming and one outgoing. Therefore more the number of incoming links higher will be the score. And in this scenario Google would be selecting page A and responding to the user.

Therefore it is seen that one page's PageRank is calculated by the PageRank of other pages. Google is always recalculating the Page Ranks. If all pages are given a PageRank of any number (except 0) and constantly recalculate everything, all Page Ranks will change and tend to stabilize at some point. It is at this point where the PageRank is used by the search engine. This algorithm is the original and simplified algorithm used by Google search Engine. The Google has also been using other algorithm to rank pages and determine page for the selection by the user. Below, the other algorithms, which Google uses, have been discussed. They are Markov Chain Model and Hidden Markov Model.

3. MARKOV CHAIN MODEL

Markov model was first introduced by A. A. Markov, to predict the behavior of a system that makes a transition from one state to another [6]. They consider only the present state. Markov model has wide application in various field of science and engineering but the most recent application of this model is on the Google search Engine. Google uses the concept of page Ranking as discussed above and this model Markov chain also finds its major role in page ranking of Google search engine.

As in [6] Markov chain is a random process where all information about the future is contained in the present state. There is no need of examining the past to determine the future. The model can be represented by FSM[7] with their states and transition that transit from one state to another. This model is a Non Deterministic Finite State Machine .The state at each step can be predicted by a probability distribution associated with the current state. It is a mathematical system model that describes transitions from one state to another on a

state space. The state space of the model depends on the current state and not on the sequence of events that preceded it. This is also known as the Markov property.
Markov Model can have Nth order [6]:

1. 0th order models depend on no prior state.
2. 1st order models depend on one previous state.
3. *n*th order models depend on *n* previous states.

Markov property [16]:

1. Behavior at time t depends only on its state at time t-1.
2. sequence of outputs produced by a Markov model is called a Markov chain
3. Process of Markov chain model in order to perform page Ranking:

In a system at any given time t=1,2,3…n occupies one of a finite number of states. At each time t the system moves from state v to u with probability P_{uv} that does not depend on t.
P_{uv} is called as transition probability and this is what determines the next state of the object considering only the current state.
Transition Matrix:
Transition Matrix T is n x n matrix. n represents the number of states. The matrix is formed from transition probability of Markov process. Each entry in the transition matrix t_{uv} is equal to the probability of moving from state v to u at time t.
Therefore $0<=t_{uv}<=1$ must be true for all u and v.

Consider the following Web graph:

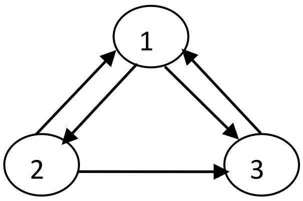

Fig3.1Web graph for 1,2,3.

There are three states or pages {1,2,3}
The transition probability of this graph is

$$t_{uv}= \begin{pmatrix} 0 & .5 & 1 \\ .5 & 0 & 0 \\ .5 & .5 & 0 \end{pmatrix}$$

This matrix determines the probability of making a transition from one state to another. There are three state 1,2 and 3.1 reaches 2 and 3 with probability of 0.5 each. From state 2 it reaches 1 and 3 with 0.5 probability and state 3 reaches 1 and 2 with 0.5 each.

Property of Transition Matrix [16]:

1. The matrix must be non negative and should be square matrix i. e the no of rows should be equal to the no of column.Each row and column represents state.

2. The entries in the matrix represents probability i. e it should be within 0 and 1.

3. The sum of entries in a row is the sum of the transition probabilities from one state to another state, therefore the sum of row value should be equal to 1.This kind of matrix is called stochastic matrix.

As per [16] Formally, a Markov model is a triple

$$M = (K, \pi, A)$$

1. Where K is a finite set of states.(in our example we have 1,2,3 states)
2. π is a vector of initial probabilities of each of the states

Here in our example vector is a matrix of 3rows with

$$\pi= \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$$

A [u, v] = Pr (state v at time t | state u at t - 1) the probability that, if M is in u, it will go to v next.

One of the major internet applications of Markov model is PageRank of a webpage as used by Google. It is defined by a Markov chain. It is the probability to be at page *i* in the stationary distribution on a Markov model on all web pages. The states determined above can be considered as web pages. Imagine random surfer surfing the web, selecting one page and going to other randomly by choosing an outgoing link from a page. This can sometime lead to page from where there are no outgoing links. So a certain fraction of time the surfer chooses a random page from the web. This theoretical random walk is known as Markov chain or Markov process. This limiting probability that an infinitely dedicated random surfer visits any particular page is its page Rank [16].

To deal with random surfers, Markov chain is being used to rank a page by Google search Engine.

The number of links to and from the page helps to find out the importance of any page and hence their page rank. The more incoming a page has, the more important the page is. Back links from more important pages carries more weight than back links from less important pages. If good page links to several other pages than the weight will be distributed equally to all those pages [16].

3.1Generalized PageRank using Markov chain

Let C = {c_{ij}} denote the adjacency matrix, a n × n matrix with c_{ij} = 1 if there is an hyperlink from page i to page j and c_{ij} = 0 otherwise. The outgoing degree of page i, meaning the number of pages that can be reached from page i, will be the row sums, denoted as s_i (based on the property of Transition matrix stated above):

$$Si = \sum_{j=1}^{n} Cij \qquad (2)$$

Now normalizing the adjacency matrix C by its row sums, the output is w_{ij} , defined by

$$w_{ij} = \begin{cases} c_{ij}/s_i, & \text{if } s_i \geq 1. \\ 1/n, & \text{if } s_i = 0. \end{cases} \qquad (3)$$

This models when the internet user is at page i. If there are S_i outgoing links on page i, the user will pick one, with equal probability, as its next page to visit. If there is no outgoing link on page i, the user will pick a random page. This is a simplified model of the user behavior.

In a transition matrix if the sum of any row is equal to 0 than the node is considered to be a dangling nodes or hanging nodes. These are node having no outgoing links. Dangling nodes cannot present in the web if it is to be presented using a Markov model. So to overcome this problem dangling nodes need to be eliminated. Langville et.al.[16] proposed a method to handle dangling nodes by replacing all the row value with 1/n, where n is the number of states or nodes. This makes the transition matrix stochastic matrix.

The Markov chain can also be elaborated by introducing an additional tuning parameter or damping factor γ/d, and probability should be between 0 and 1.

Now, the transition probability matrix of the Markov chain used in PageRank is given as follows:

$$P = \gamma \bar{W} + (1 - \gamma)\frac{1}{n}E \quad (3)$$

Where E is the n × n matrix with all entries being 1. This Markov chain describes the behavior of an internet user who, with probability γ follow an outgoing link on the current web page with equal probabilities or, if the page has no outgoing links, jumps to another page randomly. Also, with probability 1−γ, the user jumps to a page randomly with equal probability. On proper choice of γ, this Markov chain is finite, irreducible also aperiodic, and there exists a unique stationary distribution π. This stationary distribution is used to rank all the pages in W: the page i with the largest πi will be ranked first, the second largest be ranked second, and so on. In the computation of PageRank, γ is usually set to 0.85.

The process continues or iterates till the matrix converges and this convergence is possible only if the entries to the transition matrix satisfy0<=t_{uv}<=1.[16]

4. HIDDEN MARKOV MODEL

A Hidden Markov model (HMM) is a statistical Markov model .It is a nondeterministic finite state machine also called as stochastic finite state machine. It is the system assumed to be a Markov process with *hidden* states hence the name given as Hidden Markov Model. In a simpler Markov Model (like a Markov chain), the state is visible to the user, and therefore the state transition probabilities are the only parameters. In Hidden Markov Model there are two basic parameters, observed and hidden. The hidden state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. The effect of the hidden states may be observed in HMM.

This model provides three kinds of probabilistic information [13].

1. Probability of the machine when it starts.
2. If p and q are the states than the transition from state p to q is labeled with probability that it will go to the next state. If no transition exists from one state to another the probability is 0.
3. Each output state from q is labeled with the probability of reaching final state.

Formally according to the study in [8] an HMM *M* is a quintuple (*K*, *O*, π, *A*, *B*), where:

• *K* is a finite set of states,

• *O* is the output alphabet,

• π is a vector of initial probabilities of the states,

• *A* is a matrix of transition probabilities:

A[*p*, *q*] = Pr(state *q* at time *t* | state *p* at time *t* – 1),

• *B*, the confusion matrix of output probabilities.

B[*q*, *o*] = Pr(output *o* | state *q*).

HMM is a tool to find underlying processes to a given sequence. It is used to model situations like the transition of states with random variable p based on the current state of q, where q is the hidden state. Hidden state effects the observed variable p.

There are three main problems[18] for HMMs:

1. Evaluation: Likelihood a given model generated a given observed sequence.
   a. Forward Algorithm
2. Decoding: Most likely hidden sequence for a given observed sequence and model.
   a. Viterbi Algorithm
3. Training: Most likely model that generated a given observed sequence (unsupervised) or a given observed and hidden sequence (supervised).
   a. Baum-Welch Algorithm

Evaluation:

Given a set of M states and O as an output, it determines the probability of each states that provides the output state.This problem is solved using Forward algorithm

Decoding:

HMM Computes the most likely sequence of hidden states for a given model M and a given observation sequence.It finds the best path.

Training:

Using the Forward Backward algorithm also known as Baum-Welch

HMMs [13] can be applied whenever an underlying process generates sequential data: It can be used for Speech Recognition, Handwritten Letter Recognition, Part-of-speech tagging, Genome Analysis, Customer Behavior Analysis, Context aware Search etc.

Typically, HMMs can also be used within Mapper Reducer.

Hidden Markov Models are often compact. The trained HMMs can be efficiently used within Mappers/Reducers. It provides an approach to parallelizing learning.

On account of this, [17]Google inc. is using the approach of Map Reduce. It is a search engine which processes large amounts of raw data, such as crawled documents, web request logs, structure of web documents, summaries of the number of pages crawled per host, the set of most frequent queries in a given day, etc. For the large computation the input data is a huge and distributed across hundreds or thousands of machines. The issues of how to parallelize the computation in an abstraction is one major focus of Google, and to perform this it uses the map and reduce function.

Mapper takes the huge data record, splits them into nodes and using key/value pair as input, it computes a set of intermediate key/value pairs, and then

applying a reduce operation to all the values that shared the same key, in order to combine the derived data appropriately.

The use of a functional model with user specified map and reduce operations allows the search engine to parallelize large computations easily.

To carry out this function Hidden Markov Model can be used inherently within the Mapper/Reducer function of Google.

5. CONCLUSION

The concept of "PAGE RANK" determines the importance of web pages. Every web page will be given a rank based on their link structure. It is a probability distribution which is used to represent the likelihood that a person randomly clicking on links will arrive at any particular page.

The study has found that the most widely used search engine i.e. Google uses Page rank algorithm and Markov model for determining the score of the page in web server. When the user queries for any page the Google returns the page having highest score. This score for each page is determined using page Rank and Markov model. It works on the condition that the users are a random surfer. The other model, HMM also finds its application in many aspects. From the study it can also be determined that the search engine have implemented the HMM (Hidden Markov model) for map reduce function or for distributed sorting of data. Mapper Reducer functions are the abstract way of handling a huge about of data that is being collected in the web and for this HMM can be incorporated with Mapper and Reducer to perform parallization.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/PageRank

[2] Page Ranking Based on Number of Visits of Links of Web Page Gyanendra Kumar1, Neelam Duhan2, A. K. Sharma3 IEEE, International Conference on Computer & Communication Technology (ICCCT)-2011

[3] The Application of Hidden Markov Models in Speech Recognition, Mark Gales1 and Steve Young2, Foundations and TrendsR_ in Signal Processing Vol. 1, No. 3 (2007) 195–304_c 2008 M. Gales and S. Young.

[4] Hidden Markov Models and other Finite State Automata for Sequence Processing Herv´e Bourlard*y;z* and Samy Bengio*y y* Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)/handbook.

[5] Hidden Markov models David M. Blei March 12, 2012Finite State Machines, inst.eecs.berkeley.edu /~cs61c /sp08/labs/10/PH-B10.pdf

[6] Markov Chain Interpretation of Google Page RankJia Li December 1, 2005.

[7] Speech Recognition. B. Paul .Speech Recognition Using. Hidden Makov Models. https://www.ll.mit.edu /publications/journal/3.1.3.pdf

[8] A Look at Markov Chains and their Use in Google Rebecca Atherton Iowa State University MSM Creative Component Summer 2005

[9] The Performance of Page Rank Algorithm underDegree Preserving Perturbations (Senanayake, Peter Szot, Mahendra Piraveenan, Dharshana Kasthurirathna)University of Sydney, NSW 2006, Australia.

[10] Using a Layered Markov Model for Distributed Web Ranking Computation. J Wu, K Aberer ICDCS 2005. Proceedings. 25th IEEE, 2005.

[11] Reduced-Rank Hidden Markov ModelsAn Introduction to Hidden Markov Models, Max Heimel 07.10.2010.

[12] PageRank Ryan Tibshirani Data Mining: 36—462/36~662 January 22 2013.

[13] An Introduction to Hidden Markov Modelsisabel-drost.de/hadoop/slides/HMM.pdf

[14] Application of Markov Chain in the page rank algorithm.Ravi kumar, Alex GOh Kwang Leng,Ashutosh kumar Singh.

[15] MapReduce: Simpli_ed Data Processing on Large Clusters Jeffrey Dean and Sanjay Ghemawat jeff @ google .com, sanjay@google.com Google, Inc.

[16] Stochastic finite state machine for Markov Model and HMM.pg no 108-112