Ijfans International Journal Of Food And Nutritional Sciences
ISSN PRINT 2319 1775 Online 2320 7876
Research Paper Journal UGC CARE Listed (Group-I) Volume 11 , Issue 04, Apr 2022

Operators of Genetic Page-Rank Algorithm

Rakesh Kumar Giri
Associate Professor, Department of Computer Science & Engineering, Saisha Institutions, Chennai, India
Email: rakeshgiriap@gmail.com

Abstract: Genetic algorithm operators are techniques used to evolve solutions, primarily including selection (choosing fit parents), crossover (recombining parents to create offspring), and mutation (randomly altering offspring to maintain diversity). These operators iteratively improve a population of solutions, mimicking natural selection to solve complex optimization problems. Data mining is extracting and automatic discovering the web based information has been used as web mining. It is one of the most universal and a dominant application on the Internet and it becomes increasing in size and search tools that combine the results of multiple search engines are becoming more valuable. But, almost none of these studies deals with genetic page rank algorithm (GPRA), where GPRA is one of the evolutionary methods with graph structure. GPRA was designed to both increase the effectiveness of search engine and improve their efficiency. GPRA considers the correlation coefficient between stock brands as strength, which indicates the relation between nodes in each individual of GPRA. The reduced number of hyperlinks provided by GPRA in the final generation consists of only the most similar hyperlinks with respect to the query. But, the end user's not satisfied fully. To improve the satisfaction of user by using Page rank algorithm to measure the importance of a page and to prioritize pages returned from a GPRA. It will reduce the user's searching time. Page Rank algorithm works to allocate rank for filtered links based on number of keyword occurred in the content.

Keywords: Genetic Page Rank Algorithm (GPRA), Page Rank, Web Mining, Selection, Cross Over, Mutation

I. INTRODUCTION

The web is huge, diverse and dynamic and thus raises the scalability, multimedia data and temporal issues respectively. We currently drowning in information and facing information overload. Information user could encounter, among others the following problems when interacting with web: finding relevant information, creating new knowledge out of the information available on the web, learning about consumers etc., These problems are solved by web mining in directly or indirectly, not only these problems are solved by web mining and also other techniques are available to solve the above problem i.e., Information retrieval, database, natural language processing. Web mining is the use of data mining techniques to automatically discover and extract information from web documents and services and the recent interest in research is e-commerce.

IR is the automatic retrieval of all relevant documents while at the same time retrieving as few of the irrelevant document as possible. Its primary goal is to indexing the text and searching for useful information in a collection and nowadays research in IR. Web mining sub tasks are resource finding Information selection and preprocessing, generalization, and analysis. Web mining is part of information environment. Web mining categories are web content web structure and web usage mining. Multimedia data mining is instance of web content mining. Web content data are unstructured and more structured In Genetic algorithm search does not optimize the parameters of searching agents but rather directly deals with points in search space. The link structure of the web to produce a global importance ranking of every web page. This ranking called Page Rank, helps search engines and users quickly make sense of the vast heterogeneity of the WWW. Personalized information filtering system has to satisfy three requirements. They are specialization, adaptation, exploration.

II. BACKGROUNG

PageRank Algorithm

What is the PageRank algorithm?

- The PageRank algorithm is a crucial part of search ranking systems
- It was originally developed by Larry Page and Sergey Brin, the founders of Google
- Widely used: Many search engines, especially Google, rely on PageRank to help order search results
- Purpose: It evaluates and ranks web pages based on their perceived relevance and importance
- Impact: Pages that are considered more important (often because they are linked to by other high-quality pages) rank higher in search results

III. AN IMPLEMENTATION OF WEB PERSONALIZATION USING WEB MINING

Web mining is the application of data mining techniques to extract knowledge from Web. Web mining has been explored to a vast degree and different techniques have been proposed for a variety of applications that includes Web Search, Classification and Personalization etc. Most research on Web mining has been from a 'data-centric' point of view. In this paper, we highlight the significance of studying the evolving nature of the Web personalization. Web usage mining is used to discover interesting user navigation patterns and can be applied to many real-world problems, such as improving Web sites/pages, making additional topic or product recommendations, user/customer behavior studies, etc. A Web usage mining system performs five major tasks: i) data gathering, ii) data preparation, iii) navigation pattern discovery, iv) pattern analysis and visualization, and v) pattern applications. Each task is explained in detail and its related technologies are introduced. The Web mining research is a converging research area from several research communities, such as Databases, Information Retrieval and Artificial Intelligence.

Why is the PageRank algorithm important?

- The PageRank algorithm was created to tackle the difficulty of determining the importance of web pages with the immense amount of information available
- The purpose of the algorithm is to provide better search results that are more precise and related by taking into account various factors beyond just matching keywords

Key elements of the PageRank algorithm

- There are 4 key elements to the PageRank algorithm:
- Link analysis
- Link weight distribution
- Iterative calculation
- Damping factor

Link analysis

- The PageRank algorithm analyses the structure of links between pages on the web
- Web pages are given importance by the algorithm, which considers the quantity and quality of inbound links from other pages
- Each link acts as a "vote" for the target page, with the voting weight determined by the importance of the linking page
- Websites that have more high-quality links pointing towards them are deemed to be more valuable and pertinent and have a higher weight
- Webpages with a higher weight will score more highly and have a higher ranking

Link weight distribution

- The importance of a webpage is calculated by PageRank, which takes into account the total number of "votes" it has received
- The algorithm distributes the importance of a page to the pages it links to by sharing a portion of its importance with each outgoing link
- By following this process, pages of superior quality are given greater importance and make a larger impact in determining the ranking of other pages

Iterative calculation

- The PageRank algorithm uses a repetitive calculation process. At the beginning, every webpage is given the same value to start with
- In subsequent iterations, the significance of each page is re-evaluated by considering the weighted impact of inbound links
- The process continues until the rankings become stable

Damping factor

- The damping factor is a value between 0 and 1 (usually 0.85)
- It represents the probability that a user will not follow a link on a page and will instead jump to a random new page
- It prevents the algorithm from getting stuck in infinite loops and makes the model more realistic

IV. PROBLEM DEFINITION

Web users are facing the problems of information overload and drowning due to the significant and rapid growth in the amount of information and the number of users. However, the assessment of automatic semantic measures is limited by the coverage of user studies, which do not scale with the size, heterogeneity, and growth of the Web. In Google search engine output has more 100 links. when using GRA , it may reduce the hyperlinks into 50 or 40 links .But it not satisfy the user, because it may takes more time for searching.

Factors influencing Page Rank

Although the initial Page Rank algorithm mainly concentrated on link analysis, present-day search engines consider many factors to improve search results rankings. These factors may include:

- Relevance
- User engagement
- Authority and trust
- Content freshness
- Mobile-friendliness

Relevance

- The content of a web page is a crucial factor in determining its ranking in search results. This is influenced by the keywords used, the quality of the content, and how relevant it is to the search query

User engagement

- The way users interact with a website can be measured through metrics like click-through rates, time spent on a page (dwell time), and bounce rates. These metrics can reveal the level of user engagement
- Pages that receive greater engagement from users may be deemed more valuable

Authority & trust

- The reputation and authority of a webpage or website play a crucial role
- Several factors can enhance a website's ranking, including the age of the domain, quality backlinks from reputable sources e.g. government website or the BBC, and trustworthy content

Content freshness

- Search engines value fresh and up-to-date content
- Search queries may give priority to web pages that are frequently updated or have up-to-date information

Mobile-friendliness

- As mobile devices became more prominent, search engines started to factor in the mobile compatibility of web pages when determining their ranking
- Google primarily uses the mobile version of a site's content to rank pages from that site
- Having a responsive design and optimising the user experience on mobile devices can have a positive impact on a website's rankings

V. SYSTEM IMPLEMENTATION

Initially many individual solutions are (usually) randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, allowing the entire range of possible solutions (the *search space*). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

The fitness function evaluates the GRA individuals so that the similarities between hyperlinks are maximized. The proposed method tarts with a query request generated by users. Then, using standard searching engines (ex. Google [10],etc.), the initial population of GRA individuals is created randomly from the results obtained by the search engines. That is, GRA nodes encode the hyperlinks generated by any standard search engine.

How does PageRank work?

- To illustrate how PageRank works, let's use players in a football match where:
- Each player represents a page
- Each pass between 2 players represents a link between 2 pages

PageRank Analogy - a team of football players

- The main things PageRank uses are:
- The number of links the page gets (or the number of passes a player receives)
- The importance of a page is determined by the number of links pointing towards it or by how frequently the player who passed the ball is passed to
- The PageRank of each player gets updated every time they receive the ball
- This continues throughout the game

PageRank Analogy - the players receive a numerical rating based on number and frequency of passes

- As more passes are made, the PageRank of each player undergoes changes
- As a result, the PageRank of every player they pass to will be altered
- The number represents each player's PageRank - the higher the number, the better
- Once the game concludes, players can be ranked to determine the best performer

VI. CODING

Breakdown of Headers

- **<algorithm>**: Provides functions for common operations like sorting, searching, and shuffling data containers.
- **<iostream>**: Essential for standard input and output (e.g., using std::cout and std::cin).
- **<numeric>**: Contains generalized numeric operations, such as std::accumulate for summing or std::iota for filling a range with sequential values.
- **<random>**: Offers modern facilities for generating random numbers using specific engines and distributions.
- **<vector>**: Defines the std::vector container, a dynamic array that can change size.

//Genetic and Page Rank Implementation for Data Mining

In data mining, combining **Genetic Algorithms (GA)** and **PageRank** is a strategy used to optimize information retrieval and web mining. While PageRank determines the authority of a page based on link structure, Genetic Algorithms are used to evolve the ranking parameters or refine search results to better match user queries.

1. The PageRank Algorithm

PageRank measures a webpage's importance by counting the number and quality of links pointing to it. It operates on the "vote of confidence" principle: a link from page to page is a vote for 's importance.

- **Convergence**: The algorithm runs iteratively until page scores stabilize below a specific threshold.

2. Genetic Algorithm (GA) Integration

Genetic Algorithms mimic biological evolution (selection, crossover, and mutation) to solve optimization problems. In web mining, they are integrated with PageRank to create a **Genetic PageRank Algorithm (GPRA)**.

- **Initial Population**: High-quality "seed" links or initial search results from standard engines.
- **Fitness Function**: Evaluates how well a page matches a query. It often combines PageRank (link authority) with content features like keyword frequency and synonyms.
- **Selection & Crossover**: The best-performing pages are selected. "Crossover" in this context might involve combining different ranking factors—such as hyperlink count and keyword frequency—to generate a superior ranking score.
- **Mutation**: Randomly explores new or less-relevant pages to ensure the search doesn't get stuck on a single "topic drift".

```
int main()
int n=4;  //No. of data nodes
double d=0.85; //Damping Factor
```



```
vector <double> pr(n,1.0/n);
vactor<int>links={
{0,1,1,0}   // Node 0 links to 1,2
{0,0,0,1}  // Node 1 links to 3
{1,0,0,1}  //Node 2 links to 0,3
{ 0,0,0,0}  //Node 3 is a dead end (not outgoing)
};
```

Dead Ends: Node 3 has no outgoing links. In a standard PageRank model, you usually treat dead ends as if they link to every node in the graph (including themselves) to ensure the total rank remains 1.0.

//Calculate out degree for each node

The **out-degree** of a node is the total number of directed edges (links) pointing away from it. In both genetic regulatory networks and PageRank models, this represents the "influence" or "broadcasting" power of a node.

1.. Calculate for Genetic Networks

In genetic networks (like transcription factor networks), the out-degree represents the number of target genes a specific gene regulates.

- **Gene A**: If Gene A regulates Gene B and Gene C, its **out-degree is 2**.
- **Gene B**: If Gene B only regulates Gene C, its **out-degree is 1**.
- **Gene C**: If Gene C only regulates Gene A, its **out-degree is 1**.

2. Calculate for PageRank

In the PageRank algorithm, the out-degree is used to distribute a page's "rank" equally among all the pages it links to.

- **Page 1**: If Page 1 links only to Page 2, its **out-degree is 1**.
- **Page 2**: If Page 2 links to Page 1 and Page 3, its **out-degree is 2**.
- **Page 3**: If Page 3 links only back to Page 2, its **out-degree is 1**.

```
vector<int> outdegree (n,0);
for(int i=0; i<n; ++i){
for(int j=0;j<n; ++j){
if(links[i][j]) OutDegree[i]++;}}
```

//Iterative Page Rank Calculation

```
for(int iter=0; iter<10;++ iter){
vector<double>next_pr(n(1.0-d)/n);
for(int i=0; i<n;++i){
for(int j=0; j<n;++j){
if(links[i][j]) {
next_pr[j]+=d*(pr[i]/Out degree[i])
```

```
}
}
}
pr=next_pr;

//Selection
Individual Selection(vector<individual>&pop)
{
int i=rand()% pop.size();
int j=rand()% pop.size();
return(pop[i].fitness>pop[j].fitness)?
pop[i]:pop[j];

//Crossover
void crossover(individual p1, individual p2,individual & c1, individual & c2)
{
for(int i=0; i<GENE_SIZE; i++) {
if(i< cp >) {
c1.genes[i]=p1.genes[i];
c2.genes[i]=p2.genes[i];
}
else
c1.genes[i]=p2.genes[i];
c2.genes[i]=p1.genes[i];
}
}
}

//Mutate
Void mutate( individual  & ind)
cout<< “Initial best fitness:”<<endl;

//Simple one generation evolution
individual parent1= selection(population);
individual parent2= selection(population);
individual child 1, child 2

crossover(parent 1,parent 2, child 1,child2)
mutate(child 1);
child 1.calculate Fitness();

cout<<”Child Fitness after crossover and mutation: “child1.fitness<<end l;
return 0;
```



```
}
```

// Fitness function combining Precision and Recall

```
Double calculate_fitness(double precision, double recall)
{
If (precision + recall==0) return 0;
return (2*precision*recall)/(precision + recall);
}
```

//Display Results

```
cout<<"Final Page rank (Data Quality Scores);"<<end l;
for(int i=0; i<n;++i){
cout <<"Node"<<end l;
}
return 0;
}
```

FRONT END

C++ code must include “extern C” to ensure “C” linkage compatibility, which is required for Extern “C”

DEPLOYING AND PL/SQL “Call Space”

Register the library

Sql

```
CREATE OR REPLACE LIBRARY pagerank_lib  AS '/path/to/pagerank_lib.so';
```

DEFINE THE PL/SQL PROCEDURE (CALL SPECIFICATION)

```
CREATE OR REPLACE PROCEDURE compute_pagerank_ext(
from_nodes  IN RAW,
to_nodes  IN RAW,
dqs  IN RAW,
num_edges  IN BINARY_INTEGER,
num_nodes  IN BINARY_INTEGER,
damping IN NUMBER,
ranks  OUT RAW
)
AS LANGUAGE C++
```

Ijfans International Journal Of Food And Nutritional Sciences
ISSN PRINT 2319 1775 Online 2320 7876
Research Paper © 2012 IJFANS. All Rights Reserved, Journal UGC CARE Listed (Group-I) Volume 11 , Issue 04, Apr 2022

```
NAME "calculate_pagerank"
LIBRARY pagerank_lib
PARAMETERS (
from_nodes, to_nodes, dqs, num_edges, num_nodes, damping, ranks
);
/
```

VII. LIMITATIONS

- **Stochastic and Non-Deterministic Results:** Because genetic algorithms are probabilistic, running the same algorithm multiple times on the same data can yield different ranking results, which can be inefficient and inconsistent.
- **High Computational Cost:** While GA-based methods can be faster to converge than standard PageRank, they are generally time-consuming due to the iterative nature of selection, crossover, and mutation.
- **Premature Convergence:** Genetic algorithms may fall into local optima, meaning they settle on a "good enough" ranking solution rather than finding the globally optimal ranking.
- **"Rich-get-Richer" Bias:** Algorithms focused purely on in-links to determine popularity can create a bias where already popular pages are ranked significantly higher, reducing the ranking diversity for new or less-connected pages.
- **Parameter Tuning Difficulty:** Genetic algorithms require careful tuning of parameters (like population size, mutation rate, and crossover rate), which can be difficult to manage.
- **Difficulty with Large Networks:** As the network or web graph size increases, the computational overhead of running the genetic algorithm increases proportionally, making it less efficient for massive datasets.

VIII. CONCLUSION

Web mining is a very broad research area trying to solve issues that arise due to the WWW phenomenon. GPRA selects the hyperlinks considering their similarity and quality. GRA provides the most similar hyperlinks with respect to the query. The quality of the resultant pages is improved, when we using Page ranking with GPRA , compared to the standard search engines and GPRA. GPRA method can effectively provide a search strategy with minimal user's intervention but increase searching time. The proposed algorithm will reduce user's intervention and searching time.

REFERENCES

1. Eloy Gonzales, Shingo Mabu, Karla Taboada and KotaroHirasawa, Web Mining using Genetic Relation Algorithm, SICE Annual Conference2010,¥400©2010SICE,August18-21,2010.
2. Omid Panah,Amol, Iran, Amir Panah, "Evaluating the data mining techniques and their roles in increasing the searchspeed data in web", 978-1-4244-5539-3/10/$26.00 ©2010 IEEE.

ISSN PRINT 2319 1775 Online 2320 7876
 Journal UGC CARE Listed (Group-I) Volume 11 , Issue 04, Apr 2022

3. Sufal Das ,Banani Saha ,"Data Quality Mining using Genetic Algorithm", International Journal of Computer Science andSecurity, (IJCSS) Volume (3) : Issue (2).
4. Rekha Jain, Dr. G. N. Purohit, "Page Ranking Algorithms for Web Mining", International Journal of Computer Applications(0975 – 8887) Volume 13– No.5, January 2011.
5. M. Srinivas, and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms", IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No. 4, April 1994.
6. Janene K. Smith ,Edward G. Lewis," Web Mining Tools to Actualize the Semantic Web And Future Natural Language Processing ", MCIS Capstone Course, November 19, 2009.
7. Ana G. Maguitman, Filippo Menczer, Heather Roinestad, Alessandro Vespignani, "Algorithmic Detection of Semantic Similarity", International World Wide Web Conference Committee (IW3C2). WWW 2005, May 1014, 2005, Chiba, Japan. ACM 1595930469/ 05/0005.
8. Guandong Xu ,"Web Mining Techniques for Recommendation and Personalization", Doctor of Philosophy, March 2008.
9. S. A. Inamdar1 and G. N. Shinde, "An Agent Based Intelligent Search Engine System For Web Mining", Research, Reflections and Innovations in Integrating ICT in Education.
10. Bing Liu," Web Content Mining", The 14th International World Wide Web Conference (WWW-2005).
11. Sergey Brin and Lawrence Page," The anatomy of a large-scale hyper textual web search engine".