Jenny Savani, 0675557

An overview of PageRank algorithm

Introduction:

With the increasing amount of information on the internet, retrieval of information becomes a challenge. Hence, it becomes essential for a search engine technology to cope up with the growth of information on the web and maintain the scalability. With the increasing amount of information, the number of users on the internet and the number of queries the search engine handles, also increases. This requires a technology which can handle the queries quickly, store indices of web documents efficiently, and process large data in the documents effectively. These tasks eventually become challenging with increasing users and information on the internet. To overcome these challenges, both the web and technological aspects, like hardware performance, disk seek time, robustness of operating system, et cetera should be considered.

Google is designed considering all the above mentioned aspects and it has favorable scaling properties as a search engine. It has capability to scale large data in an efficacious manner. It stores the indices of the documents by making efficient use of storage space and possesses data structure with rapid access. Moreover, the cost required in indexing and storing information is less when compared to the amount available. The Google search engine is capable of producing results with high precision, mainly because it uses link structure of the web in order to calculate comparative ranking for each web page, known as PageRank method, and the other reason is, it uses links to improve the search results.

PageRank method measures relative ranking of web pages and determines their 'importance'. It prioritizes the results obtained from the searches of web keywords. Large data in the form of graphs or with links between them, are processed using the PageRank method to find the most important information amongst them.

This paper contains a detailed study of PageRank algorithm, its application by Google as a ranking algorithm on the internet, its working mechanism and related concepts. This paper also discusses strengths and some demerits of the PageRank algorithm. Moreover, certain variations and several areas where PageRank algorithm is used or can be used other than the internet, are discussed as well. It also contains brief information about similar ranking algorithms like HITS, SALSA, and OPIC along with their advantages and disadvantages.

Overview of PageRank method:

PageRank algorithm is a very commonly used algorithm. It is used for ranking web pages. Ranking of web pages is based upon the importance, comparing the other web pages. The web pages are linked with each other forming graph-like structure. All the links refer to each other in the graph. The page, which possesses the most number of links with the other pages becomes the important page. On the other hand, the other pages, which have comparatively less number links with other web pages, are considered to be not that important. PageRank algorithm is used to rank the importance of each of these web pages present in the link structure of the web. Moreover, if a web page possesses any link from an important page, then it is also considered important. In simple words, rank of the page depends upon its in-degree and importance of the web pages linked with it.

Google uses PageRank to yield the important web page, by moving all the related important web pages up in the result's page, whenever a user searches for anything using keywords or tags. When put in general words, the ordering works according to the following mentioned steps:

- Keywords used for searching the content are used to find all the matching or relating web pages.

- Compute the ranks of web pages with the help of page factors like keywords.
- The inbound anchor text in HTML hyperlink is calculated.
- Arrange the web pages obtained as results by scores obtained using PageRank algorithm.

Using the PageRank algorithm, the results are obtained in the form of probability distribution. This indicates the chances of a person arriving at any particular page by clicking on the links of web pages randomly. The PageRank can be computed on any size of collections consisting of web pages. Initially, all the web pages in a collection contain the equal distribution. Computation takes place in the form of several iterations. During these iterations through the web pages in collections, the approximate PageRank values are adjusted according to the expected or theoretical values.

Since, the PageRank algorithm assigns ranking in the form of probability distribution, values lie between 0 and 1. If a web page in a collection has 0.80 probability or PageRank value, it means there is a 80 percent chance a user clicking any random link in the collection will be directed to this particular web page.

Working of PageRank algorithm:

According to Google, if we consider two pages, page X and page Y, and let's say page X is linked to page Y, then page Y is considered to be an important page.

Now consider four web pages- P, Q, R, and S. Initially PageRank is initialized with the same value for all the web pages. In the next iteration, for a given web page, the PageRank is transferred equally amongst all the neighboring outbound linked web pages. Hence, if there are links from web pages P, Q, and R to S, then in the next iteration each link i.e. P, Q, and R would transfer 0.25 PageRank to S. This would totally make transfer of 0.75 to S, making it the most important web page.

Now let us assume web page Q has links to pages P and R, R has a link to P, and web page S has links to all the three web pages P, Q, and R. Initially all the web pages would have the same PageRank value i.e. 0.25 for each web page. For the first iteration web page Q would transfer 0.125 or say half of the quarter value to both of the web pages P and R. Since, R has a link to P, it would transfer 0.25 to P, which is the existing value of R. Next, S has links to all the three web pages, P, Q, and R and it would transfer one third of its existing value (0.25 / 3 = 0.083), 0.083 to all three web pages. At the end of this iteration, web page P would have PageRank around 0.458, Q would have 0.083, and R would have 0.208. This can mathematically be given as,

$$PR(P) = \frac{PR(Q)}{2} + \frac{PR(R)}{1} + \frac{PR(S)}{3}$$

$$= \frac{0.25}{2} + \frac{0.25}{1} + \frac{0.25}{3} = 0.125 + 0.25 + 0.083 = 0.458$$

In the above formula, 2, 1, and 3 represents the number of outbound links from web pages Q, R, and S respectively.

Thus, general formula to calculate PageRank value at any node x can be given as,

$$PR(x) = \Sigma \ (PR(y_i) / L(y_i))$$

Where PR(y) is the PageRank values of each web page y (with inbound links to the page x) and L(y) is the total number of outbound links from page y.

The above discussed formula is simplified from the algorithm. But another factor which is also considered in PageRank theory is regarding the random surfer. Any surfer would click on the links randomly, and after sometime he/she will stop clicking the links. The probability that a surfer will stop clicking the links present in the current page and will request any other random page (like directly requesting a new URL and not clicking or following any link from the current page), at any step, is known as damping factor *d.* This damping factor is used in the PageRank formula, by adding it to a single page or group of web pages. After testing different values, 0.85 is set as the damping factor. It can be explained as, if the damping factor is set to 0.85 or 85%, then there is a 15% chance that a surfer or user will not continue clicking the links in the current page, rather he/she will open a new URL randomly.

Hence, formula with damping factor *d* becomes something like,

$$PR(x) = (1 - d) / N + d (\sum (PR(y_i) / L(y_i))$$

Where $PR(x)$ is PageRank for page x, d is the damping factor which is set to 0.85 generally, N is the total number of web pages or documents in the collection, $PR(y_i)$ is the PageRank of all the inbound web pages linked to x (where i $\in$ 1 to N), and $L(y_i)$ is the total outbound links from page y_i.

The term $d(\sum (PR(y_i) / L(y_i))$, actually stops other pages creating influences and the value is damped down, when multiplied with d.

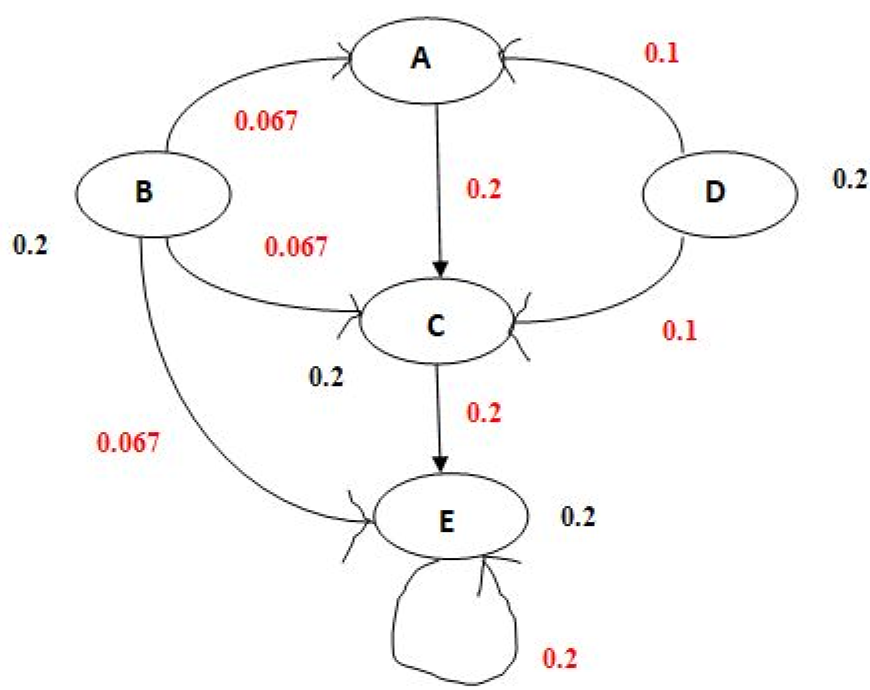
The sum of all the web pages in a collection will be equal to one. The values in PageRank are entered as the dominant right eigenvector, which is modified by a rescaled adjacency matrix. Also, the PageRank can be calculated either algebraically or iteratively. Iterative method is similar to the power method in mathematical operations.

The above mentioned formula is considered to be the general formula. However sometimes there are several variations and extensions made to it. Like in the original paper proposed by Sergey Brin and Lawrence Page, they mentioned the formula without including N (in denominator). And hence, according to that formula, the sum of all the web pages in the collection would not result in one, instead would result in N.

We have certain assumptions/rules while calculating PageRank theoretically. If a web page has any link to itself, then it is ignored. Also, if there are more than one outbound links from one page to another web page, then all the links are treated as a single link. Initially all the web pages in the collection are assigned the same and equal PageRank value.

Numerical example of PageRank:

Consider the following graphical structure of five vertices A, B, C, D, and E and eight edges connecting them with incoming and outgoing links. The initial probability is 1 and it gets divided equally amongst all the vertices. Hence, each vertex gets 0.2 PageRank value. (shown in black color in the image below)



The numbers in red color shows weight each edge would share depending upon the number of outgoing links.

Now depending on the outgoing links each vertex will share its pagerank in the following way:

- Vertex A has one outgoing link, hence it shares 0.2 with vertex C.
- Vertex B has three outgoing links, hence it will share 0.2 / 3 = 0.67 pagerank with each outgoing link towards A, C, and E.
- Vertex C has one outgoing link, hence it shares 0.2 with vertex E.
- Vertex D has two outgoing links, it shares 0.2 /2 = 0.1 pagerank with A and C.
- While vertex E has one outgoing link to itself, thus it share and receives 0.2 from itself.

Using formula for calculating PageRank for the first iteration,

$$PR(x) = (1 - d) / N + d (\sum (PR(y_i) / L(y_i))$$

Here N = 5, total number of pages (vertices) in the collection. d is the damping factor which is 0.85 generally.

For vertex (page) A, since it has two incoming links from page B and D with pagerank = 0.2. Page B has totally 3 outgoing links, while page D has totally 2 outgoing links, hence pagerank formula looks something like below,

PR(A) = (1 - 0.85) / 5 + 0.85 (0.2 / 2 + 0.2 / 3) = 0.172

Similarly for page B, C, D, and E,

PR(B) = (1 - 0.85) / 5 + 0.85 (0) = 0.03 (because it has 0 incoming links)

PR(C) = (1 - 0.85) / 5 + 0.85 (0.2 / 1 + 0.2 / 3 + 0.2 / 2) = 0.342 (A has 1 outgoing link, B has 3, and D has 2)

PR(D) = (1 - 0.85) / 5 + 0.85 (0) = 0.03 (because it has 0 incoming links)

PR(E) = (1 - 0.85) / 5 + 0.85 (0.2 / 3 + 0.2 / 1 + 0.2 / 1) = 0.426 (Page B has 3 outgoing links, page C has 1, and page E also has 1 to itself)

Hence after the first iteration, these values of PageRank are transferred, which all when summed becomes equal to 1. The iterations will continue for the defined number of iterations.

Markov's chain in PageRank:

It can be noticed that the working of PageRank method more or less relies upon the Markov chains. Markov property makes the study of random processes easier. For a given random process, if there is any value used by the process at the current time, we can not obtain any information or details about the future behaviour of this process by analyzing and collecting information about the past of the process. Mathematically it can be written as, at any given time, for any process, the conditional distribution of the future states in the random process rely upon only the present state and not upon any of the past states (Assuming both the states, past and present are given at the time). This is memoryless property in simple words. A random process exhibiting Markov property is known as Markov process.

When we define discrete space and time with the Markov process, we get the Markov chain. Hence, the Markov chain consists of a discrete sequence of states which follows Markov property and are randomly drawn from a discrete space of states.

Markov chains are irreducible, hence we can traverse from one step to any other, for any chain of stages presented in any form of strongly connected graphs. Also, in an irreducible Markov chain, if any one state is aperiodic in the entire chain, then all the other states are also aperiodic. This means while going from one state, let's say state A to the next state, state B in the chain and again returning to the previous state, state A from state B, it would require only one step. And this holds true for all the stages in the entire chain. States in the Markov chains are also recurrent, which means that once we leave any state, we will surely come back (return) to that state although in any time, with probability equal to 1. In more details, all the stages are in positive recurrent states, it means that we can calculate expected recurrence time which is finite expected return time while leaving any recurrent state.

Any Markov chain with irreducibility and positive recurrent states, shows stationary probability distribution. When a chain has stationary probability distribution, it does not evolve with time. The probability distribution in the initial state will remain the same throughout all the steps in future. Also, when a chain is aperiodic, irreducible, and recurrent positive, then despite what the initial probabilities are, when time steps tend towards infinity, the probability distribution of the chain starts converging and this means the chain have limiting distribution, which is again called stationary distribution in other words. Another property of the Markov chain is ergodicity. With increase in time, the chain shows the same statistical behavior when averaged over time just as the entire space of states in the chain would. Hence, the temporal mean becomes equal to the spatial mean when an infinitely long run for trajectory is considered.

The working of PageRank algorithm can be interpreted with the concept of Markov chain and its properties. Markov chain concept can be used to explain how the pages in a collection can be ranked given the links between them. As explained earlier about a user who randomly surfs through web pages in a collection, he/she would start surfing at some initial time with one of the web pages within a collection. From this initial web page, all the other linked web pages would have equal probability or chances to be clicked by this user. Different pages in a collection can be compared with different states in a Markov chain. Transition probabilities are considered as the links between the pages, where each linked page has equal probability of getting chosen. The memoryless behavior of PageRank can totally be related with the behavior of the user (who randomly surfs throughout the pages in a collection). By assuming that the chain of the web pages in the collection has aperiodic and recurrent positive properties, then after considerably a long time, the probability distribution of the current page would converge to stationary distribution. Hence, it does not matter what a starting page is according to the user, after considerably a long time, if a random time step is picked, all the pages in the collection will have almost fixed probability to be the current page.

The whole concept behind PageRank is that the page which is likely to be most probable in the stationary distribution will be the most important page. We end up visiting such pages seldom since they receive links from pages which are also frequently visited during the entire cycle. The stationary probability distribution is then used to determine the PageRank value for each state.

Variations of PageRank:

PageRank algorithm can be varied according to the requirements. This is done by changing or amending some features or properties from the original algorithm. With certain variations, it can be used in applications of different contexts other than just on the internet. Following are some of the variations of PageRank which are widely used:

1) Weighted PageRank algorithm:

   In this modification of the original algorithm, the rank scores are decided based upon the popularity of the pages, by taking into account the number of incoming and outgoing links of the pages. In this algorithm, the rank is not divided equally amongst all the outgoing links like the traditional algorithm. Rather the most popular pages get the higher value of rank. The popularity of pages is decided by observing the number of incoming and outgoing links it has.

2) Second PageRank algorithm:

   This is not very different from the original method but it explains the concept of random surfer in a better way. It explains the effectiveness of PageRank in mapping the probability that a random surfer will finish up at any given page. Also, it can determine the probability effectively that a random surfer will visit a page.

3) PageRank for undirected graph:

   PageRank was designed traditionally for directed graphs, but it can also be applied to undirected graphs. In such a case, the out-degree of the vertex should be equal to the in-degree of the vertex. Since the number of edges are proportional to the number of vertices, undirected graphs seem to be having more convergence curves in gradual manner. As the number of edges increases, connectivity also increases and the convergence can be achieved just after a few iterations.

4) Personalized PageRank algorithm:

   One type of variation of the PageRank algorithm is the personalized PageRank algorithm. The only thing which differentiates it from the original PageRank algorithm is that the traversing of the graph is personalized or more biased towards the starting vertices of the graph. The jumps from vertices are back to the set of the starting vertices at the end.

Other applications of PageRank algorithm:

PageRank is Google's most widely used ranking algorithm. It is designed to determine the web page's relevance on the internet and is highly associated with the search engines. PageRank is designed in a way that it uses the data in the form of graph structure and determines the importance of each node in the graph. The PageRank algorithm can not only be used just for the web context, but also for a plethora of applications as far as the data to be used is in the form of graph structure. In PageRank applications other than the internet context, interpretation of the importance of a node varies according to the application and it should maintain the basic notion of determining the importance. PageRank algorithm is used in several applications with some personalization and certain variations (Few variations are mentioned in the previous topic).

- In analysing flight data and to find the most important/ busiest airports

  The normal flight data can be converted into graph structure by using airports as the vertex or nodes of the graph and routes between the airports as edges/links in the graph. The airport which has the most number of incoming and outgoing links with the other airport is the busiest airport or the most important airport. The links depict the routes which will be maximum for the busiest airport.

- In Twitter for suggesting friends or other users:

  PageRank can be used to provide friend suggestions over Twitter. Users of Twitter are divided according to their role as consumers and producers. Two copies of each individual are made in the network. One is a producer copy and the other is a consumer copy. Then with the concept that user A follows user B, user A is considered to be the consumer and user B is considered to be the producer. Using this, an edge is created from A to B and another edge is created from B to A. This forms a bipartite graph. When this new graph is created, personalized PageRank is performed on it, which will provide new friend (which is producer in this case) suggestion for all the users.

- In toxic waste cleanup and management:

  PageRank algorithm is sometimes used in the management of toxic waste. PageRank is used to determine the position of water molecules in an ionic solution, this helps in finding the best methods to remove toxic chemicals and other nuclear wastes. Using PageRank toxic chemicals can be located where they are likely to pool in the solution. This makes the waste cleanup more efficient and faster.

- In prediction of foot and road traffic:

  PageRank algorithm can predict the traffic flow on the road maps, both individual and connected, which are represented in the form of graphical structure. Intersections are used as edges while the streets are considered to be the streets. It is also used to reflect the observed human mobility in some urban spaces, very accurately.

- In debugging errors:

  One of the variations of the PageRank algorithm is the MonitorRank version. It is designed to analyze engineered systems which are complex in nature. This algorithm returns a list of systems based on their contributions or participations in any anonymous situation. MonitorRank does not crawl through pages containing errors and debug the callbacks. It rather analyzes the structure of the system containing errors and bugs itself and identifies the probable cause of the problems or errors.

- In analysing scientific impact of researchers:

  An index named PageRank-index similar to h-index is used to determine a ranking system for individual publications and citations which propagates to individual authors. It makes use of underlying networks of collaborations and citations along with the PageRank algorithm.

- In ranking the academic doctoral programs:

  PageRank algorithm is used in ranking doctoral programs in academics. It is ranked based upon the records of placing their graduates in the position of faculty. The academic departments are linked to one another by hiring their faculty from each other and also from themselves.

Advantages of PageRank algorithm:

- PageRank algorithm takes less time for computation because it pre computes the rank score. Hence, it yields the results faster
- PageRank algorithm is more efficient than other ranking algorithms because it carries out the computation at the time of web crawling and combines the computed result with the traditional information retrieval score during the query time.
- Since the PageRank algorithm computes at the crawl time and not during the query time, it is more feasible for complex and large graph data like web pages.
- As the PageRank algorithm generates the ranking of the entire graph data at once, and not as a small subset of the data, it is less susceptible to the localized link spams.
- PageRank algorithm determines the important pages based on the pages having the highest PageRank score or if there are any pages with high PageRank score pointing towards any pages. Sometimes even only one link from an important citation like Yahoo!'s homepage is enough to make a page important. It might happen sometimes that the page has a broken link or may not have high quality, hence Yahoo!'s homepage would not link to it. But the PageRank algorithm can handle such problems by recursively propagating weights through the links in the graph structure (of web pages).

Disadvantages:

- The main disadvantage of the PageRank algorithm is that it does not consider new pages, and only supports the older pages in the web structure. No matter how good the new pages are, since they do not have many links in the starting, such pages will not be getting any advantages unless they are part of any existing site which is densely connected with other sites.
- As it does not consider the text or content of the site while ranking, it happens sometimes that the user might not get any desired or relevant results from the search.

- Sometimes pages in a network get into any infinite link cycles, giving rise to rank sinks problems.
- When a page contains links in which the hypertext points to some page with no outgoing links, this creates the problem of dangling links.
- Any references in the site in circular form will reduce the PageRank of the front page.
- If a group of pages does not have any links from inside the group to outside of the group, then it would create a problem in accurate computation of PageRank.
- It might happen sometimes that the PageRank encounters a dead end (pages with no outgoing links). This would also affect the computation and decrease the overall PageRank.

Other ranking algorithms:

Other algorithms for link analysis and ranking, based on user queries like HITS, SALSA, and OPIC are discussed in brief as follow:

HITS:

HITS stands for Hyperlink-Induced Topic Search and it is a web structure mining algorithm. HITS is a search query dependent algorithm and it ranks the web pages by processing all of its out links and in links in the data. Whenever a user uses any query to search something, the search engine returns all the relevant pages. HITS processes these pages and expands the obtained list, then it generates two types of rankings from the expanded set of pages from the list, one is hub ranking and the other is authority ranking. A web page is labeled as authority, if many hyperlinks are pointing to this page, while on the other hand a page is labeled as hub, if any page is pointing to many hyperlinks. Hence, the output of this algorithm generates two types of pages as follow:

- Authority: These are the pages which provide some sort of important and trustworthy information about any search or topic.

- Hub: These are the pages which contain links towards the authorities.

A page can be considered as a good hub and a good authority at the same time. A good authority page is the one which is pointed by many good hub pages for the same subject. While on the other hand a good hub page for a subject is the one, which points to many authority pages for the same subject.

In the HITS algorithm, data is used in the form of graph structure, a directed graph G(V, E) where V is a set of pages as vertices and E is the set of edges that match these links.

The HITS algorithm has two main steps. One is the sampling step and the other step is the iterative step. In the sampling step, relevant pages obtained for the given search or query are collected, and a subgraph from the original graph is retrieved which has a high number of influence pages. While in the second step, the iterative step, hubs and authorities are found using the output of the sampling step.

Advantages:

- HITS focuses on the links of the web pages as well as the content of the web pages, hence it returns the relevant search for the query unlike the PageRank algorithm.
- HITS is sensitive to the user query.
- HITS calculates the authority nodes and hubness accurately.

Disadvantages:

- Since HITS is a query dependent algorithm, it is a time consuming algorithm.
- Implementation of HITS is costly.
- HITS uses traditional search engines to find the relevant set of pages, and then attempts to find hubs and authorities. This computation is carried out at query

time, and hence it is not feasible for search engines used these days, which has to handle millions of searches per day.

SALSA:

SALSA stands for Stochastic Approach for Link Structure Analysis, which is a query dependent link-based ranking algorithm, similar to HITS. SALSA combines the ideas of both the PageRank algorithm and HITS algorithm. SALSA visualizes the graph in a bipartite form, like HITS, where hubs are pointing towards the authorities. SALSA algorithm performs a random walk for traversing on the bipartite hubs and authorities graph, by processing the authority and hub graphs alternatively. This random walk starts from the authority node selected uniformly at random. Alternatively between backward and forward steps, a random walk is carried out. At a node on the authority side of the bipartite graph, one of the incoming links is selected uniformly at random. Then it is moved to a hub node on the hub side. The stationary distribution is defined using authority weights according to this random walk.

Advantages:

- SALSA combines the idea of PageRank and HITS, hence it contains the best features of both of them, which is a strength.
- SALSA does not have any issues like topic drifting, where irrelevant and off topics influence the scores, mostly dominate the score by ignoring the relevant information containing nodes.
- It is less likely to get influenced by spamming because the coupling between authority and hub score is not very strict.

Disadvantages:

- SALSA is effective for performing general queries and not for the complex queries, when compared to HITS.

- Sometimes SALSA returns inconsistent values for authority and hub weight vectors.

OPIC:

OPIC - Adaptive On-Line Importance Computation is similar to the HITS algorithm, but it is used to compute scores for each page. As the name suggests, it works online, and moreover uses less CPU power and disk resources. It can be applied to calculate HITS score as an alternative to calculate the largest eigenvector using the power method. This algorithm does not require the link matrix to be stored like other algorithms. Being online, it updates the estimation of page importance continuously every time a web page is being visited. Therefore, it focuses on the crawling of only the most important pages, which saves utilization of resources. This algorithm uses greedy strategy for page selection and not the random strategies like other algorithms. Because of this error factor decreases much faster, than it would with random strategy. Also, one of the variants of this algorithm is able to update itself dynamically according to the changes by users on the web.

Advantages:

- This algorithm can compute dynamically changing pages by updating the graph accordingly with nodes and edges, making online computation easier and feasible.
- OPIC algorithm mainly focuses its crawling on the most important pages. Also, it is fully integrated in the crawling process.
- It can be biased towards specific fields of interest according to the users and hence utilizes less resources like CPU power and disk space.
- Newly discovered pages are also considered as a part of the computation, unlike PageRank algorithm. The computation does not depend on the order of the visiting pages, but it depends on the entire graph, considering one page at a time.

- It uses the greedy approach for page selection strategy, which helps in reducing error factor on an average twice faster. While other algorithm uses random approach, which makes convergence slower.

Disadvantages:

- It is expensive to compute.
- Due to dynamic computations, pages come, update or disappear affecting the graph structure and edges as well. Hence, crawling of web pages takes more time and sometimes becomes complex.
- The choice of adaptable size and time windows according to the changes in graph is an issue.

Summary:

PageRank has found a profound place in the ranking of web pages searched on Google. Not only for searching queries on the internet but ranking any data which has the graph structure. Not only PageRank, but also the other graph analysis algorithms are widely used to extract meaningful information from the network structure. When several graph analysis algorithms like HITS, SALSA, and OPIC are compared, PageRank and its variants are found to be hard to spam. Also, HITS and SALSA are found to behave unexpectedly on some graphs, in other words they lack consistency in the performance. While the PageRank does not show such behavior. Despite few disadvantages, PageRank is found to be the best fit for the large network of data and for computing millions of searches at the same time. Moreover PageRank can be modified according to the requirement and can be used for numerous applications. This makes the PageRank algorithm more flexible and feasible.

References:

1)http://infolab.stanford.edu/~backrub/google.html

2)http://netsavy.net/Graphics/PageRank.pdf
3)https://en.wikipedia.org/wiki/PageRank
4)https://towardsdatascience.com/brief-introduction-to-markov-chains-2c8cab9c98ab
5)https://blogs.cornell.edu/info2040/2014/11/03/more-than-just-a-web-search-algorithm-googles-pagerank-in-non-internet-contexts/
6)https://web.stanford.edu/class/msande233/handouts/lecture8.pdf
7)https://blogs.oracle.com/bigdataspatialgraph/intuitive-explanation-of-personalized-page-rank-and-its-application-in-recommendation
8)https://www.ijert.org/research/comparative-analysis-of-pagerank-and-hits-algorithms-IJERTV1IS8530.pdf
9)https://www.irjet.net/archives/V4/i12/IRJET-V4I1251.pdf
10)http://www2003.org/cdrom/papers/refereed/p007/p7-abiteboul.html
11)https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/106104189/lec31.pdf