Efficient Ranking on Websites Using ScaleRank with Interpolation Search

Tinku Varghese[1], Subha Sreekumar[2]

Abstract— **A system designed for efficient ranking on website. This system helps us to refine the details as per the user's interest and used for efficient authority flow ranking on websites using pre-computed database. Ranking is a technique which helps us to derive the data on the basis of some basic characteristics like number of visits and time spend on the page. This system will provide a solution for how efficiently the authority flow ranking can take place. For efficient ranking, this paper uses an approximation algorithm ScaleRank with interpolation search. The existing system based on ScaleRank with binary search, so this paper contains a comparison between them. ScaleRank is an algorithm which approximates the ranking algorithm DataApprox. The result obtained from this technique is that the website works according to the need of the user.**

Index Terms— **Authority Flow, DataApprox, Interpolation Search, Ranking, ScaleRank.**

I. INTRODUCTION

Data mining is the process of analyzing data from different perspective for summarizing the data into a useful format. The main function of data mining is to extract, transform the information from dataset and save for future use. This system gives more importance to web mining and database searching. Web mining is an area in data mining where the data are retrieved from a webpage. Here the concept web mining is done with personalized web search. In database searching, the data are already stored in repository are derived according to the user query. The four functionalities of data mining is that (1) Stores and manages data; (2) Provide access to the authorized users; (3) Analyze the data using different tools; and (4) Present the data in a useful and understandable format like tables or graphs. In ranking, if we are considering two items then the comparison is like that one item is equal to, greater than or less than the second item. Likewise here we are considering ranking to derive the data as per the user interest.

Authority places an important role in this system by measuring the importance of an object. Authority flow is the way of representing the rating of pages for each user. In existing system the authority flow ranking can be done with different authority flow techniques like PageRank [1], [4], [6], ObjectRank [1], [2], [3], [4], [6], [9] etc. PageRank [1], [4], [6] is a method used for the rating of webpages objectively and mechanically by using link structure of webpage. Link structure can be explained with an example by considering three pages A, B and C. Webpages A and B are connected to the webpage C correspondingly. Then this depict that webpages A and B are C's backlinks and webpage C is A and B's forward link.

PageRank [1], [4], [6] implementation can be done in five steps: (1) Convert each URL into a unique integer and store each hyperlinks in a database using this integer IDs(identities) to identify each webpage; (2) These IDs are used for the sorting in link structure; (3) Remove all dangling links from database; dangling links are links that points to any page with no outgoing link; (4) After removing dangling links, start iteration for making initial assignment of ranking and (5) Removed dangling links are added back for next iterations. PageRank [1], [4], [6] is not efficient for keyword searching, so the concept of ObjectRank introduced. ObjectRank [1], [2], [3], [4], [6], [9] gives higher results that either contain keywords of query or are semantically associated to keyword of query. The result of keyword query is a list of objects of database ranked according to the query.

Different ranking algorithms available are: HubRank [1], [5] which employs query log statistics to select, computes and store small fraction of nodes as fingerprints. BinRank [1], [2], [3] performs ObjectRank on one graph for any keyword query. For authorized personalization uses an entity relationship graph [1], [4], [8]. This entity relationship graph does not use any distance method for approximation. So there are two methods for candidate ranking based on distance method, which are SchemaApprox [1], [4], [7] and DataApprox [1], [4], [7]. SchemaApprox uses a Euclidean distance to choose the m- candidates. DataApprox uses an objective function to choose m-candidate from a data graph level. These two approximation algorithm are very expensive in query interaction time. So introduce a heuristics ScaleRank algorithm.

For efficient execution ScaleRank algorithm includes a searching method known as binary searching. But the binary searching method is very complex, if the number of m-candidate value increases. For solving this problem, we consider the ScaleRank algorithm with interpolation search. Interpolation search is more efficient than binary search in terms of complexity. This paper also contains the comparison between ScaleRank with two searching methods binary and interpolation search.

II. RELATED WORKS

A. Ranking with PageRank Algorithm

PageRank [1], [4], [6] is a method for rating the importance of webpages objectively and mechanically using

Manuscript received Feb, 2013.
***Tinku Varghese**, CSE, M. G. University, Kottayam, India.*
***Subha Sreekumar,** CSE, M. G. University, Kottayam, India,*

ISSN: 2278 – 1323

a link structure. PageRank scheme consist of node with and without output links. PageRank implementation consist of five steps: First, URLs are converted into unique integers and stored into the database as hyperlinks with this integer IDs to identify each webpage. Then the sorting in link structure can takes place with these unique IDs. Then remove all the dangling links from the database. Then make the initial assignment of rank and start the iteration. Lastly, add the dangling links back to the database. PageRank criteria are: freshness and relevance of content, number of visits and time spend on page. PageRank is an algorithm used for ranking webpages.

B. Ranking with ObjectRank Algorithm

ObjectRank [1], [2], [3], [4], [6], [9] is a keyword searching algorithm. This is the algorithm used for bibliographic database. ObjectRank [1], [2], [3], [4], [6], [9] system consist of two models: (1) Offline mode: Here approximation is done by a materialized sub graph, this can be pre-computed in this mode for supporting the online query, (2) Online mode: Once the query receives, ranking algorithm starts working. The output obtained from this algorithm is a bin of terms. BinRank algorithm is used for the processing of this bin of terms. BinRank [2], [3] is an algorithm which approximate ObjectRank utilizing an approach inspired by a traditional query processing. The goal in constructing term bin is that these bins will control the execution time. The demerit of this system is that the computation is very expensive.

C. Ranking on Entity Relation Graph

Authorities are provided by flow based ranking technique and entity relationship graph [8]. The key feature of entity relationship graph which provide personalization to the system. In entity relationship graph, the authority flow parameters are adjusted with the help of edge or relation type. Here authority can be originates in two ways from a query and a set of objects and spread through edges. In an entity relationship graph, all queries first compute a base set and from the base set, the authority is spread to the whole graph. For authority flow personalization in entity relationship graph ObjectRank [1], [2], [3], [4], [6], [9] and HubRank [1], [5] algorithms are used.

In ObjectRank, first computes a base set that contain set of objects, create nodes from objects according to the entity type and edges are created on the basis of edge type. HubRank is a new system introduced for fast and dynamic space efficient proximity search in entity relation graph. A personalized PageRank vector (PPV) [5] is used for personalization. PPV provides a ranking mechanism which creates a personalized view of individual user. The PPV can consist of a hub node (pages pointing to many important pages) which is based on query logs, chosen words and other entity nodes for PPVs. The problem in this technique is that distance method is not used and cannot implement distance method in entity relationship graph.

D. Ranking with DataApprox and SchemaApprox

The above methods are not efficient for distance method so introducing two approximation algorithms which is based on distance method which are DataApprox [1], [4], [7] and SchemaApprox [1], [4], [7]. SchemaApprox [1], [4], [7] is defined at schema level and consist of a schema level matric. For choosing m-candidates, minimize the distance between different candidates in schema level matric using Euclidean distance. For choosing m-candidates, uses objective functions which minimize the distance between the different candidates at the data graph level. Both DataApprox [1], [4], [7] and SchemaApprox [1], [4], [7] are too expensive to facilitate interactive query response.

To minimize the expense there arise the concept ScaleRank with binary search [1], [4], [7]. ScaleRank is an approximation of DataApprox. Input for the ScaleRank algorithm is a weight assignment vector (WAV) and output are top *K* objects based on authority score. ScaleRank which maintains a repository and which consist of WAV and ranking vector for each candidate.

III. PROPOSED MODEL

A. ScaleRank with Interpolation search

ScaleRank [1], [4] is an algorithm used to approximate, the approximation algorithm DataApprox [1], [4]. ScaleRank algorithm is used to scale personalized ranking. In the previous system the ScaleRank algorithm works on binary search [1], [4]. Place the items into an array and sort them either ascending or descending order with respect to the key. In binary search, first compare the item in the middle position of array with the key. If the key is lesser than the middle item then the item can be place to the lower half of the array and if the key is greater than the middle item then the item can be place to the upper half of the array. This procedure will repeat till the completion of comparison between each item. The comparison can be repeated for *n* times, where *n* is the number of elements. This system can be time consuming, so we introduce the ScaleRank with interpolation search. This system will help us to obtain the search result as soon as possible.

Interpolation search forms better results than a binary search for a sorted and uniformly distributed array. In interpolation search, *log(log(n))* comparison is possible where *n* number of elements is considered. This method is searching for a given key value in an array. Here we consider each set of keys as search spaces and find whether the particular key is coming under the search space or not. If that search space does not contain the key we do not consider the search space for further comparison. So the result can be obtained in a limited time. In experimental results, compare the execution time of both interpolation search and binary search in ScaleRank algorithm. The result will be based on the comparison done on the system for executing a keyword query. The following figure (Fig. 1) [1] represents the architecture of this system which will explain where this interpolation search is used in the system and how efficiently the searching can be done.

ISSN: 2278 – 1323

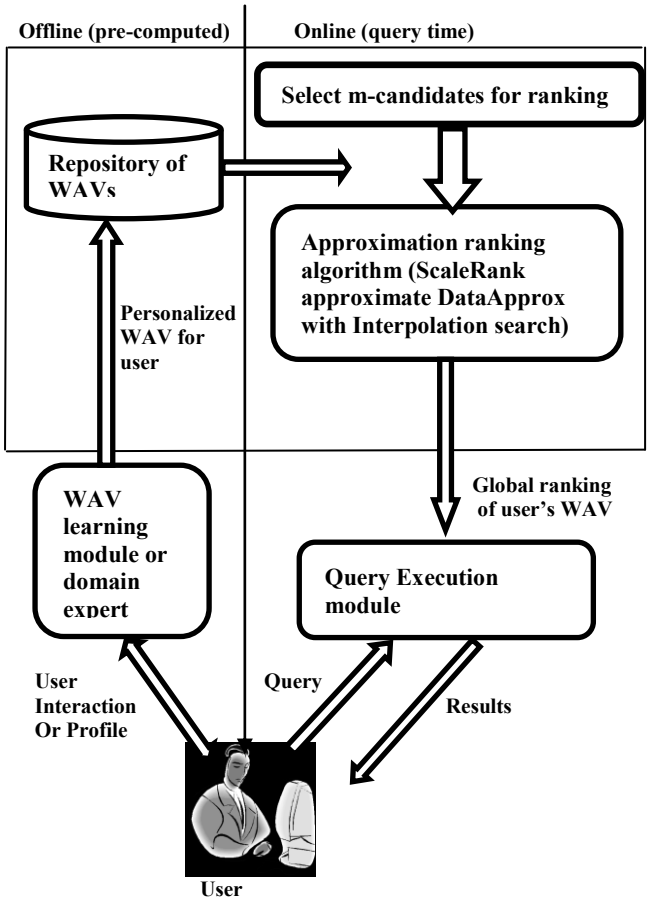

Fig.1: System architecture of the system

System works on two modes: offline (pre-computed) and online (query time) mode. The working of the system starts from offline mode. First the user needs to register to the system or update his profile. With the help of user's detail, WAV learning module sets up the weight assignment vector for respective users. The WAV can be set up in two ways: by a domain expert with the help of user's profile or automatically with the help of user's interaction and user's feedback. After setting up the WAV, the personalized WAV of the user can be update to the WAV repository. After the completion of offline mode, online mode starts execution. Online mode, first selects the m-candidates and gives this m-candidates and the WAV of a user to the approximation ranking algorithm i.e. ScaleRank approximates DataApprox with interpolation search. Interpolation search work in the system in a way that this consider the WAV as the key and the ranking can be done in the approximation ranking algorithm module. Here global ranking of user's WAV is done and this will be stored to the query execution module. When a user requests for a query as a keyword, then the result can be generated with respect to the WAV of each user.

IV. EXPERIMENTAL RESULTS

Here in this section, we show the results by comparing the two searching techniques, binary search and interpolation search in ScaleRank algorithm. Section B contains the ScaleRank algorithm with Binary Search. Section C contains the ScaleRank with Interpolation Search. Section D contains comparison between both of the search methods.

A. ScaleRank Algorithm

The input of ScaleRank algorithm is WAV of a single object, select m-candidate and finds the top K objects on personalized authority flow. The main highlight of this algorithm is that m-candidates are selected with respect to the WAV. ScaleRank algorithm is also known as hybrid algorithm because its uses SchemaApprox [1] distance in the first step and in the next step this algorithm solves DataApprox [1]. But doesn't mean that ScaleRank algorithm approximates SchemaApprox, this only approximate DataApprox. ScaleRank maintain a repository of m-candidate rankings. WAV and ranking vector are stored for each user. Fig.2 represents the ScaleRank algorithm framework.

Fig.2: ScaleRank algorithm framework

B. ScaleRank algorithm with Binary Search

In binary search [1], we compare the key with the middle key and if the key is less than the middle key then the item sought to the lower half or else sought to the upper half. In this algorithm, we consider upper bound as u, lower bound as I and accuracy requirement as τ. The search continuous until the condition $|u-I| < \tau$ is reached. The comparison possible in binary search is n, where n number of elements. The figure (Fig.3) shows the top K objects execution in ScaleRank algorithm with binary search. This figure shows that the top K values increases execution time.

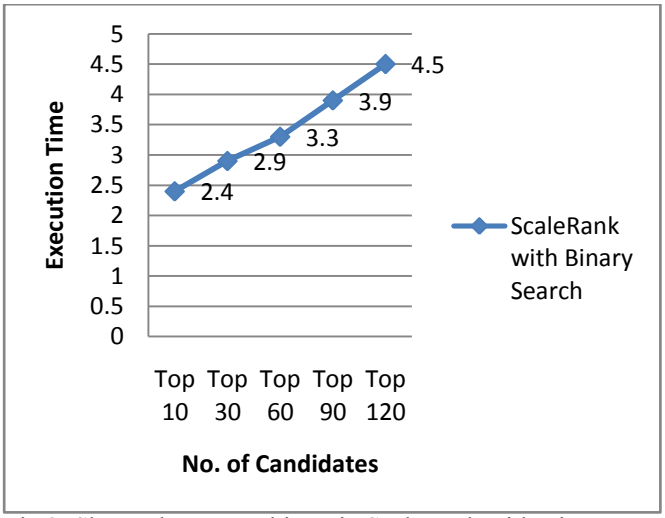

Fig.3: Shows the top K objects in ScaleRank with Binary Search

C. ScaleRank with Interpolation Search

Interpolation search is used for searching a given key value in an indexed array ordered by the value of key. Searching can be done in a way that, each step analyze where the remaining item set can be located. Key value should come under the boundary of search space. After obtaining the key value, that will be compared with the key values being sought. If the key is not equal then we do not consider that portion of search space for searching but only consider the

ISSN: 2278 – 1323

remaining portion. On average the interpolation search makes about *log(log(n))* comparisons, where *n* is the number of elements to be searched. The following figure (Fig.4) shows the execution time taken by interpolation search when the number of data sets increases.

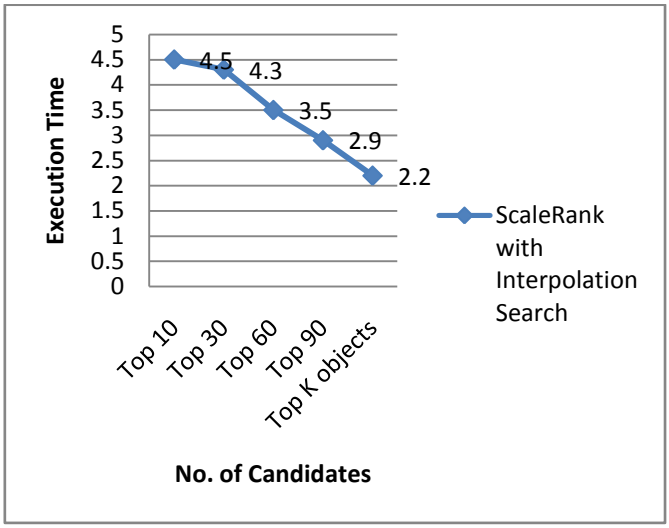

Fig.4: Shows top *K* objects in ScaleRank with Interpolation Search

D. Comparison between Interpolation and Binary Search

In binary search, we choose the middle key value of the search space but in interpolation search first we search whether the search space contain any such key then the comparison can be started else this goes for other search space. So the elements for the comparison can be minimized. If we are comparing the execution time of both interpolation and binary search in ScaleRank algorithm, then the time required for interpolation search is less than the time required for binary search. Here the execution time can be calculated with respect to the comparison taken place by the system at the time of searching. In case of binary search *n* comparison can be possible but in case of interpolation search *log(log(n))* can be possible, where *n* is the number of elements. The following figure (Fig.5) shows the comparison between interpolation and binary search.

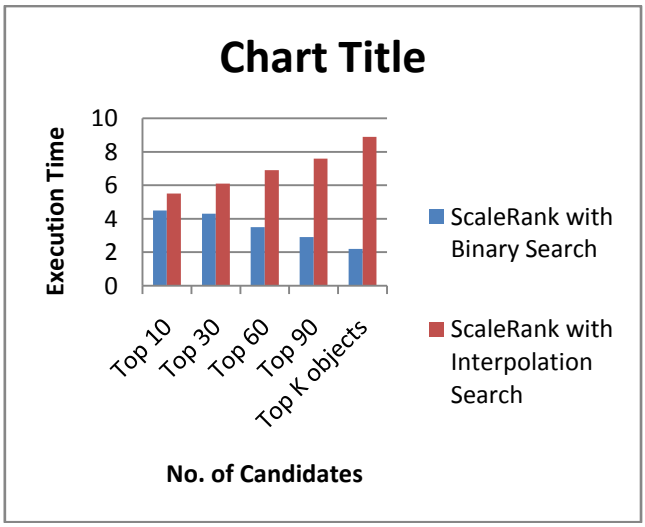

Fig.5: Shows the comparison between Interpolation and Binary Search

V. Conclusion

This paper concludes that the interpolation searching than the binary searching method applied to the ScaleRank algorithm. In binary search time of execution of keyword increases by increasing the search results in the database, but in interpolation search the execution time decreases because if a particular search space does not contain the key, then we won't consider that data sets. This paper also contain the comparison between two searching method which will help us to conclude the result obtained is correct

Acknowledgment

The first author would like to thank all the people guided and supported. Without their valuable support and guidance the task cannot be completed. Also like to thank all the colleagues for their valuable suggestions.

References

[1] Vagelis Hristidis, Yao Wu and Louiqa Rasschid, "Efficient Ranking on Entity Graphs with Personalized Relationships", *IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 4, pp. 850-863, April 2014.*

[2] T. D. Khadtare, P. R. Thakare and S. A. J. Patel, "An Efficient Personalized Wed Search Mechanism using BinRank Algorithm", *International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, vol. 2, issue. 4, March 2013.*

[3] Heasoo Hwang, Andrey Balmin, Berthold Reinwald and Erik Niikamp, "BinRank: Scaling Dynamic Authority Based Search using Materialized Sudgraph", *IEEE International Conference on Data Engineering, pp. 66-77, 2009.*

[4] Lara Srour, Ayman Kayssi and Ali Chahab, "Personalized Webpage Ranking using Trust and Similarity*", IEEE International Conference on Data Engineering, pp. 454-457, 2007.*

[5] Soumen Chakrabathi, "Dynamic Personalized PageRank in Entity Relation Graph", *International World Wide Web Conference Committee, pp. 571-580, 2007.*

[6] Ramakrishna Varadarajan, Vagelis Hristidis and louiqan Raschid, "Explaining and Reformulating Authority Flow Queries", *School of Computing and Information Science, Florida International University, 2008.*

[7] Yao Wu and Louiqa Raschid, "ApproxRank: Estimating Rank for a Sudgraph", *IEEE International Conference on Data Engineering, pp. 54-65, 2009.*

[8] Vagelis Histridis, Yao Wu and Louiqa Raschid, "Scalable Link Based Personalization for Ranking in Entity Relationship Graphs*", IEEE International Conference on Data Engineering, 2007.*

[9] C. Vijaya Ram, C, H, Srinivasulu and T. Jacob Sanjay Kumar, "Inverse ObjectRank: Dynamic Authority Based Search in Databases", *International Journal of Computer Science and Information Technologies, vol. 2(5), pp. 2193-2194, 2011.*

Tinku Varghese, Department of Computer Science & Engineering, Mangalam College of Engineering, Ettumanoor, Kerala, India

Subha Sreekumar, Assistant Professor, Department of Computer Science & Engineering, Mangalam College of Engineering, Ettumanoor, Kerala, India

ISSN: 2278 – 1323