Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (21): 4892-4896, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.4892.4896
© 2013 Asian Network for Scientific Information

Efficient Personalized Pagerank Computation: A Survey

Hou Honglun and Wu Minghui
Department of Computer Science and Engineering, Zhejiang University City College,
Hangzhou, 310015, China

Abstract: As personalized PageRank has been widely leveraged for ranking on graph-structured scenarios; its computation efficiency becomes a prominent issue. In this study, the authors survey on an array of work that concentrate on efficient and scalable Personalized PageRank (PPV) computation, ranging from earlier work that attempt to use partial precomputation to improve online efficiency, to recent work that estimate approximate PPV for full personalization and the hybrid methods. A comparison about these methods in terms of the ability of personalization, scalability, online/offline efficiency and accuracy and a few possible research directions are presented at the end of this study.

Key words: Web search, personalized ranking, PageRank algorithm

INTRODUCTION

As the variant of traditional PageRank algorithm, Personalized PageRank has been widely applied in ranking scenarios where the dataset can be represented using graph models, such as Web search (Fogaras *et al*., 2005; Jeh and Widom, 2003; Haveliwala, 2003; Nie *et al*., 2005; Page *et al*., 1999; Richardson and Domingos, 2002), Entity-Relation Graph search (Chakrabarti, 2007), keyword-based relational database search (Bahmani *et al*., 2010) and so on. Similarly, Personalized PageRank also leverages the linkage structure of data graph to recursively compute the weight of each node in a random walk model.

However, different from the traditional PageRank, when computing the weight of each node, Personalized PageRank also considers the user-specific information such as the bookmarks, the query logs, to compute the personalized scores for each query. Therefore, Personalized PageRank is a query dependent algorithm and the Personalized PageRank Value (PPV) of each node actually reflects the relevance to the query.

Unfortunately, computing exact PPV's is infeasible even on a moderately large graph, regardless of online computation or offline precomputation, due to the prohibitive time or space cost. On the one hand, online computation is impractical since it requires a power iteration algorithm over the entire graph. On the other hand, complete offline precomputation is also infeasible, since the space requirement to store the precomputation is at least quadratic, as Fogaras *et al*. (2005) shows. Thus, designing efficient algorithms for personalized PageRank has become an important research area.

This study reviews an array of work that attempt to make personalized PageRank feasible and scalable, ranging from the earlier work that attempt to solve exact PPVs, to recent work that instead estimate approximate PPV.

PRELIMINARIES

Before presenting the Personalized PageRank, there is a brief review on the traditional PageRank algorithm.

Traditional PageRank algorithm: PageRank algorithm was proposed in (Brin and Page, 1998) to compute the authority of each page in the Web. Specifically, if there exists a link from page p to page q, then p passes its authority to q. Therefore, by a recursive definition of page authority, pages with more in-coming links would gain higher authority. Formally, over a directed graph G = <V, E> where nodes V represent web pages and edges E represent links between pages, the authority of each node is computed using a random walk model, where the random surfer starts walking from arbitrary node, with 1-\alpha probability the user would visiting one of its neighbors at random and with \alpha probability, she would jump to an arbitrary node to start a new walk. This random walk behavior repeats until the visiting probability of each node in G converges to a steady state, when the

Corresponding Author: Wu Minghui, Department of Computer Science and Engineering, Zhejiang University City College, Hangzhou, 310015, China

distribution of visiting probability is a n×1 vector with each value representing the authority of a node, formally defined as:

$$r = (1-\alpha) Mr+\alpha u \quad (1)$$

where, α is the jumping probability or damping factor, usually set to be 0.15, M is the adjacency matrix corresponding to the graph G and u is a uniform vector with each element has the value 1/n. The solution to Eq. 1 can be computed iteratively and is guaranteed to converge uniquely due to the fundamental theorem of Markov chains.

Personalized PageRank: Personalized PageRank is query-dependent which is different from the traditional PageRank algorithm. Specifically, personalized PageRank can be directly used to measure the similarity of nodes in a graph, e.g., the relevance of each node to the query node. Conceptually, in the random walk model, the random surfer would only jump to some preferred nodes each time when she teleports and starts a new random walk. Therefore, in the steady state, those preferred nodes and nodes in the neighborhood would have higher visiting probability. This biased random walk can be formalized as:

$$R = (1-\alpha) Mr+\alpha v \quad (2)$$

where, v is the personalized vector which biases towards query-related nodes (or simply called the query nodes). Specifically, given q asquery nodes, their corresponding entries in v are set to 1/q and all other entries to 0. In terms of the random surfer model, with α probability, instead of teleporting to all nodes with equal probability, now the surfer jumps only to query nodes with equal probability.

One of the typical applications of the personalized pagerank is to rank the search results according to the similarity to the queries. For example, in Web search, with the personalized vector encoding the keyword-matched pages, the corresponding PPV represents the relevance of each page in the Web to the query. However, since the personalized vector can be determined only in the query time, the personalized PageRank vector can be computed in an iterative manner during query time as well, which can take hours on very large graphs such as the Web. This contrasts with the typical sub-second response time to a query as generally expected by users. Therefore, the computational efficiency of PPV becomes a big challenge and many works has been focusing on improving the efficiency of PPV computation. There are mainly three kinds of studies in this area, ranging from the exact, partial personalization methods, approximate to full personalization methods.

Partial and exact personalization: Partial and exact personalization algorithm is the early PPV direction. Such algorithms are able to precompute partial PPVs by controlling the granularity of personalization or limiting the scope of the precomputation. Specifically, Haveliwala (2003) and Kamvar *et al*. (2003a) proposes a coarse-grained PPV calculation method which precomputes a PPV for a set of nodes (instead of each node) with the same type; while Jeh and Widom, 2003 proposes the hub based algorithms which compute PPVs only for those important nodes (hub nodes) and construct the PPVs w.r.t. the queries by combining the hub PPVs according to the linear theorem.

Topic-sensitive personalization: Topic-sensitive PageRank algorithm (Haveliwala, 2003) was in the context-aware Web search application. It aims at computing a personalized vector for each of the 16 top topics in Web Open Directory (ODP), that is, it precomputes a personalized PageRank vector ri for each topic Ti during offline. In Eq. 1, the entries of personalized vector depend on whether a page belongs to a certain topic, for example, if a page p belongs to topic T, then the corresponding entry v_p in the personalized vector has non-zero value. Therefore, in terms of the random walk model, the random user can only jumps to a page belonging to the T with equal probability so that the steady-state probability distribution reflects the relevance of each page regarding to the topic T. Therefore, this partial personalized algorithm is called topic sensitive PageRank algorithm.

Topic sensitive personalized PageRank algorithm reduces the granularity of personalization to achieve possible offline PPV precomputation. For the datasets of any size, topic sensitive PageRank algorithm only needs to compute 16 topic related personalized PageRank vectors and thus has the advantage in both offline processing time and storage space.

However, in the practical application, it often requires the relevance regarding each page rather than each topic. Hence, during query time, PPV w.r.t the topics would be assembled for the query-specific PPV. Haveliwala, 2003) proposed to compute a belonging probability Pr (Ti|q) for each query at real-time so that by a linear combination of the corresponding PPV, the similarity of any page p and query q can be calculated as:

$$score(p) = \sum_j Pr(T_j \mid q)\, r_j(p) \quad (3)$$

However, one big drawback of this coarse-grained algorithm is it is difficult to control the accuracy of the algorithm. There are hundreds of millions of pages in the Web (corresponding to a large number of potential queries), however the topic sensitive PageRank only supports 16 topic PPV, which is quite a rough mapping from pages to topics; on the other hand, the errors in mapping pages to topics will inevitably affect the accuracy of the algorithm.

Block-based personalization

Kamvar *et al.* (2003b) proposed a block based personalization algorithm: BlockRank. The intuition behind BlockRank comes from an experimental finding: in the linkage graph, more than 80% links are intra-host links, while less than 20% are inter-host links. Therefore, the links between Web pages present a block structure. Based on this discovery, Kamvar proposed a site-level modeling method for Web links in which each node represents a site (or a block), rather than a page and the edges between nodes represent all possible links between pages in that block.

This site-level abstraction provides a good foundation for personalized PageRank. On the one hand, BlockRank calculates a local PPV for each page inside a site during offline and on the other hand, at the query time, BlockRank computes a site-level PPV according to user's preference and assembles it with the local PPV to generate the PPV for each page.

Conceptually, there is a two-stage random walk model supporting BlockRank algorithm. First, the random user would follow the links between sites and teleports between sites (i.e., Site-level PPV computation), then when visiting a specific site, the random user would follow inter-host links between pages (i.e., page-level local PPV).

The essence of BlockRank is similar as PageRank, but constraint the precomputation to a set of particular nodes (i.e., topics or sites) rather than each page. Therefore, it is a coarse-grained PPV personalization algorithm. However, since BlockRank captures the block structure of Web graph, the site-level PPVs are able to represent the global PPV based on the entire graph, while combined with local PPV at the query time, the accuracy would be improved.

Hub-based personalization: Jeh and Widom (2003) proposed a hub-based personalized PageRank algorithm which is different from the above two methods. It no longer precomputes the PPVs w.r.t a set of nodes to control the granularity of computation but only precomputes PPVs for some specific pages, known as hubs, in the Web. The hub based method computes a PPV for each hub node, also known as hub vector and assembles them to build any hub-based PPVs through linear theorem.

Theorem 1 (Linearity). For any two preference vectors u1 and u2, r1 and r2 denote the corresponding PPV, then for arbitrary constants β_1, $\beta_2 \geq \beta_1 + \beta_2 = 1$, the following equation holds:

$$\beta_1 r_1 + \beta_2 r_2 = (1-\alpha)\, M(\beta_1 r_1 + \beta_2 r_2) + \alpha(\beta_1 r_1 + \beta_2 r_2) \quad (4)$$

Based on the linearity theorem, 2^k PPV can be constructed from the precomputed PPVs of k hubs, supporting a larger scaled personalization. Jeh also proposed an efficient PPV precomputation method by reusing the overlapping link structure between hubs. Specifically, Jeh proved the equality between PPV calculated by Eq. 2 and the inverse P-distance between nodes:

Definition 1 inverse P-distance: For any two nodes p and q, the inverse P-distance from q to p is defined as:

$$r'_q(p) = \sum_{t: q \to p} P[t]\, \alpha\, (1-\alpha)^{l(t)} \quad (5)$$

where, P [t] represents the visiting probability of q to p through an arbitrary path t, i.e.. the sum of probabilities of following each step in the random walk model and l (t) represents the length of t, i.e., the number of edges in t.

The inverse P-distance defines the PPV value of node p w.r.t. query node q as the total probability from q to p through all possible paths. According to inverse P-distance, the computation of PPV can be divided into two parts: Paths through hubs and paths not through hubs:

$$r_q = r_q^{\bar{H}} + r_q^{H} \quad (6)$$

where, $r_q^{\bar{H}}$ represents the visiting probability through paths that do not contain hubs, a.k.a, partial vector.

In order to improve the computational efficiency, only the partial vectors and the inter-probability between hubs (i.e., hub skeleton) were stored during offline, while the hub vectors are constructed at the query time as Hubs theorem shows.

Theorem 2 (Hubs): For arbitray nodes $p \in V$, $H \subseteq V$:

$$r_q^{H} = \frac{1}{c} \sum_{h \in H} (r_q(h) - cx_q(h))(r_h^{\bar{H}} - cx_h) \quad (7)$$

where $r_q(H) = \{ (h, r_q(h)) | h \in H \}$ is the hub skeleton and $r_q^{\bar{H}}$ are the partial vectors of hubs.

Compared with the above methods, hub based algorithms are able to support a large scaled personalization, however, its drawback is that this personalization only woks for hub-based queries rather than the arbitrary query.

Full and approximate personalization: One major drawback of the earlier study of efficient PPV computation is the partial personalization. To achieve full personalization, some researchers propose the approximate PPV computation methods that reduce the computational cost by giving estimate rather than exact PPVs. Since this kind of approximate methods have no constraint on either the number or the type of PPV, they can achieve full personalization on the arbitrary query.

Web skeleton: Web skeletons Jeh and Widom (2003) is an approximate method, proposed to conquer the drawback of the hub based algorithm, i.e., the personalization is limited to a set of hubs.

Web skeleton is an extension of Hub skeleton method that is also based on Inverse P-distance. Since hubs are the nodes with a large number of links, it is quite possible that paths between two nodes contain one or more hubs, in other words, there are quite a few paths that do not pass through hubs. Therefore, given a well-selected hub set, the probability of paths passing through hubs is a good estimate to the exact probability, i.e., the probability of all possible paths. Thus, for any query node q, the exact probability can be estimated by the probability of paths passing through hubs, i.e., r_q, r^H_q.

According to Hubs Theorem, $r_q^{\bar{H}}$ can be assembled by the partial vectors of hub nodes and the probability of visiting each hub from q. To improve the online query efficiency, the probability of visiting each hub can be precomputed during offline. For all possible query q, the precomputed visiting probability is called Web skeleton, i.e., $W = \{r_q(H) | q \in V\}$. Given the web skeleton, the full personalization can be achieved at query time.

Monte carlo stimulation: Fogaras proposes a Monte Carlo stimulation method which is different from the traditional Power Method for PPV estimation. Monte Carlo stimulation is still based on the Inverse P-distance definition, ie, the PPV value of a node can be calculated by summing up the visiting probability of all paths arriving at it. Based on this definition, Fogaras proposes a random step walk model called fingerprint to sample the paths and uses the probability of sample paths as the estimate of the exact PPV.

Definition 2 (Fingerprint): Suppose in a random walk, the surfer stops at current node with \alpha probability and with 1-\alpha probabiliity visits next node. Then the steps of such random walk L has the probability distribution:

$$\Pr(L = i) = \alpha (1-\alpha)^t \quad (8)$$

Each fingerprint represents the random path within L steps and thus the PPV of node q can be estimated by N random walks from q, ie, the visiting probability distribution of such N fingerprints represent the PPV value of each node being reached from q. This estimation can be formalized as:

$$r_q(p) = \Pr\{\text{the fingerprint ands at p}\} \quad (9)$$

Fingerprint estimation has a major property that the scale of personalization can be controlled the accuracy of estimation. With less tour samples, the efficiency of approximation would become higher and thus the full personalization can be achieved during offline phase. However, on the other hand, less tour samples would also affect the accuracy of estimate. Therefore, the accuracy and efficiency of Fingerprint would highly depend on the number of tour samples in computation.

CONCLUSION AND FUTURE WORK

As a popular ranking algorithm, the efficiency of Personalized PageRank is a key fact affecting the performance of search system. Many works focus on the efficient computation of PPV ranging from the earlier work of exact, partial personalization to recent work of approximate, full personalization. The comparison of the related works are summarized in Table 1.

Direction of future work: While past decades have amassed a large amount of research in efficient computation of personalized PageRank, this line of work is far from finished. A few possible future research directions are pointed out as follows:

- Top K methods. In generally, since the total number of nodes in a graph can be very large; users are usually only interested in top k results (a typical value of k is 10 or 100). The exact ranking of the majority nodes at lower ranks is not crucial. While a few preliminary methods on fast top k query (Bahmani *et al*., 2010; Berkhin, 2006; Ilyas *et al*., 2008) have been proposed in recent years, there is an abundance of research opportunities in this direction

Table 1: Efficient PPV computation algorithms comparison

	Scalability	Offline efficiency	Online efficiency	Accuracy
Topic sensitive	16 topic	Efficient for computing only 16 PPVs	Combine topic PPVs to answer query	Exact topic PPV but the accuracy of query would be affected by how a page matches a topic
lockRank	No. of sites	Efficient local PPV computation	Combine local PPV with site-level PPV	The accuracy of local PPV can be improved by assembling site-level PPV
Hub-based	No. of hubs	Only need to compute partial PPV and hub skeleton	Assembly hub PPV by hub equation	Accurate PPV for hubs
Web skeleton	Arbitrary node	Only need to compute partial PPV and web skeleton	Estimate any PPV by hub equation	Accuracy depends on the selection of hubs
Fingerprint	Arbitrary node	Need to compute the fingerprints of sample tours	Directly load from fingerprint index	Accuracy depends on the No. of sampled tours

- **Evolving graph:** All previous methods on efficient computation of personalized PageRank assume a static graph. However, in the real application scenarios, graphs are generally fast-evolving. For example, there are new nodes being added to or removed from the Web constantly, virtually every second. Previous methods require to redo the precomputation once the graph is changed. Thus, to enable full personalization with respect to an evolving graph, incremental approaches that do not need to re-compute everything again is preferable.

ACKNOWLEDGEMENT

Present research work is partly supported by the Science Foundation of Zhejiang Province under Grand No. 2010R50009.

REFERENCES

Bahmani, B., A. Chowdhury and A. Goel, 2010. Fast incremental and personalized PageRank. Proc. VLDB Endowment, 4: 173-184.

Berkhin, P., 2006. Bookmark-coloring algorithm for personalized pagerank computing. Internet Math., 3: 41-62.

Chakrabarti, S., 2007. Dynamic personalized pagerank in entity-relation graphs. Proceedings of the 16th International Conference on World Wide Web, May 8-12, 2007, Banff, Canada, pp: 571-580.

Fogaras, D., B. Racz, K. Csalogany and T. Sarlos, 2005. Towards scaling fully personalized pagerank: Algorithms, lower bounds and experiments. Internet Math., 2: 333-358.

Haveliwala, T.H., 2003. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. IEEE Trans. Knowledge Data Eng., 15: 784-796.

Ilyas, I.F., G. Beskales and M.A. Soliman, 2008. A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv., Vol. 40, No. 4. 10.1145/1391729.1391730

Jeh, G. and J. Widom, 2003. Scaling personalized web search. Proceedings of the 12th International Conference on World Wide Web, May 20-24, 2003, Budapest, Hungary, pp: 271-279.

Kamvar, S., T.H. Haveliwala, C.D. Manning and G.H. Golub, 2003. Exploiting the block structure of the web for computing PageRank. Stanford University Technical Report. http://www.stanford.edu/~sdkamvar/papers/blockrank.pdf

Kamvar, S.D., T.H. Haveliwala, C.D. Manning and G.H. Golub, 2003. Extrapolation methods for accelerating PageRank computations. Proceedings of the 12th International Conference on World Wide Web, May 20-24, 2003, Budapest, Hungary, pp: 261-270.

Nie, Z., Y. Zhang, J. Wen and W. Ma, 2005. Object-level ranking: Bringing order to web objects. Proceedings of the 14th International Conference on World Wide Web, May 10-14, 2005, Chiba, Japan, pp: 567-574.

Page, L., S. Brin, R. Motwani and T. Winograd, 1999. The pagerank citation ranking: Bringing order to the web. Technical Report Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/.

Richardson, M. and P. Domingos, 2002. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In: Advances in Neural Information Processing Systems 14, Dietterich, T.G., S. Becker and Z. Ghahramani (Eds.). Vol. 1, MIT Press, USA., ISBN-13: 9780262042062, pp: 1441-1448.