Applied Soft Computing 13 (2013) 1686–1692

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Applying reinforcement learning for web pages ranking algorithms

Vali Derhami*, Elahe Khodadadian, Mohammad Ghasemzadeh, Ali Mohammad Zareh Bidoki

Electrical and Computer Engineering Department, Yazd University, Yazd, Iran

ARTICLE INFO

Article history:
Received 28 June 2012
Received in revised form
16 December 2012
Accepted 30 December 2012
Available online 11 January 2013

Keywords:
Ranking
Search engine
Reinforcement Learning
Artificial intelligence
Value function
Agent

ABSTRACT

Ranking web pages for presenting the most relevant web pages to user's queries is one of the main issues in any search engine. In this paper, two new ranking algorithms are offered, using Reinforcement Learning (RL) concepts. RL is a powerful technique of modern artificial intelligence that tunes agent's parameters, interactively. In the first step, with formulation of ranking as an RL problem, a new connectivity-based ranking algorithm, called RL_Rank, is proposed. In RL_Rank, agent is considered as a surfer who travels between web pages by clicking randomly on a link in the current page. Each web page is considered as a state and value function of state is used to determine the score of that state (page). Reward is corresponded to number of out links from the current page. Rank scores in RL_Rank are computed in a recursive way. Convergence of these scores is proved. In the next step, we introduce a new hybrid approach using combination of BM25 as a content-based algorithm and RL_Rank. Both proposed algorithms are evaluated by well known benchmark datasets and analyzed according to concerning criteria. Experimental results show using RL concepts leads significant improvements in raking algorithms.

1. Introduction

Nowadays World Wide Web (WWW) is considered to be the best source of information. Its importance mainly is due to easy access, low-cost and being responsive to users' requests in the shortest time [1]. Search engines are the predominant tools for finding and getting access to the contents on the web. Whenever users seek information, enter their query in search engine. The search engine searches through web pages and return a list of relevant ones.

Generally, search engines involve three processing stages. The first stage is called crawling. A crawler visits a web page, and follows all the links provided in that page. This operation leads to constructing a web graph (a web graph consists of nodes and edges, where nodes stand for web pages and edges show the links which are available from each page to other pages). After collecting web pages, content of each page is analyzed to determine how it should be indexed (e.g. words are extracted from the titles, headings, or special fields). Indexing allows information to be found as quickly as possible. Ranking is the final stage. In this stage millions of web pages were recorded in the previous stage are sifted to find matching cases for a specified query and sorting them based on the users' requests or preferences. Due to the huge size of the web, it is very common that a large number of relevant results are returned for a given query. Moreover, studies have shown that users do not have the time and the patience to go through all of them to find the ones which they are interested in. They often consider the top 10 or 20 results [2]. Therefore, an efficient ranking algorithm is required. This algorithm enables search engines to present the best related pages to users in response to their queries.

In this paper, we propose a new algorithm for ranking web pages based on web graph. The objective is determining the score of each web page based on paths which can be reached to that web page from other web pages as well as the out-degree (number of out links) of pages in the traverse paths. Consider a random surfer who transfers between pages randomly. After visiting a web page; she selects next page by clicking randomly on one of the links in that page. This process can be considered as a Markov Decision Process (MDP). Therefore, we formulate it as a Reinforcement Learning (RL) [3] problem where the objective is "policy evaluation". Elements of RL in this problem are defined as follows: 1 – *states*: web pages, 2 – *Actions*: out links on each page, 3 – *Policy*: agent (surfer) selects the next page by clicking randomly on one of the out links in current page. 4 – *Reward*: inverse of the out-degree of the source page. 5 – *Value function*: value function of each state (page) is the total amount of rewards that surfer can expect to accumulate during traveling through pages to reach that page. The proposed approach is called RL_Rank. Based on the above definitions, value function of each page is considered as the score of the page.

Since RL_Rank is a connectivity-based algorithm; in the next step, we combine it with BM25 which is a content-based algorithm and propose a hybrid ranking algorithm.

* Corresponding author.
E-mail addresses: vderhami@yazduni.ac.ir (V. Derhami), khodadadi@stu.yazduni.ac.ir (E. Khodadadian), m.ghasemzadeh@yazduni.ac.ir (M. Ghasemzadeh), alizareh@yazduni.ac.ir (A.M. Zareh Bidoki).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.12.023

The remainder of this paper is organized as follow: Ranking algorithms are discussed in Section 2. In Section 3 we illustrate our proposed ranking algorithms. Experimental analysis and their results are presented in Section 4. Finally, in Section 5 we summarize our main contributions and discuss some possible further improvements on our proposed method.

2. Ranking algorithms

Web pages ranking algorithms divided into two categories namely content-based and connectivity-based algorithms.

Content-based algorithms usually work based on matching words in documents. In other words, for each query the documents with the most similar content to the query will be selected. Vector space [4], TF-IDF [5] and BM25 [6] are examples of these algorithms. These algorithms are suitable for well structured environments such as digital libraries, rather than the web pages which usually include large number of unstructured contents. Connectivity-based algorithms use links between web pages. Links carry information which can be used to evaluate the importance of pages and the relevancy of pages to the user query. These algorithms are divided into two major classes "query-independent" and "query-dependent". Instances of query-independent algorithms are PageRank [7], HostRank [8] and DistanceRank [9]. These algorithms use the entire web graph and compute the score of web pages offline, whereas query-dependent algorithms such as HITS [10] involve the construction of a query-specific graph, in other words these algorithms are online.

In the first step of this research, we concentrate on connectivity-based algorithms which are offline. Among these algorithms, PageRank as a well known and mostly used algorithm is at the center of our attention. In the second step, we present a hybrid approach with combination of a connectivity algorithm and a content algorithm. BM25 is used as a content-based algorithm in the proposed hybrid approach.

2.1. PageRank algorithm

PageRank is a popular ranking algorithm used by Google search engine. PageRank models the users' browsing behaviors as a random surfer model. In this model, a user surfs the web by randomly clicking links on the visited pages and sometimes jumps to another page at random. In this algorithm, fraction of time the surfer spends on a page is defined as the score of that page [11]. PageRank measures the importance of web pages as follows: the score of a page such as i, based on PageRank method, can be approximated by the following recursive formula [7]:

$$R(i) = \sum_{j \in B(i)} \frac{R(j)}{O(j)} \tag{1}$$

where $R(i)$ and $R(j)$ are rank scores of pages i and j, respectively. $O(j)$ is the number of out links in page j which is called out-degree of page j. $B(i)$ is the set of pages that point to page i.

In fact, PageRank supposes that a link from page p_1 to p_2 indicates that the author of p_1 is interested in page p_2. If a page has many links in other pages, it can be concluded that many people are interested in that page and the page should be considered an important one. PageRank takes the backlinks (incoming links to a web page) into account and propagates the ranking through links: a page has a high rank if the sum of the ranks of its backlinks is high. Fig. 1 is an example to show how score of a page is computed in each step with PageRank algorithm that the score of page p in each step is updated with:

$$R(p) = \frac{R(A)}{4} + \frac{R(B)}{3} \tag{2}$$

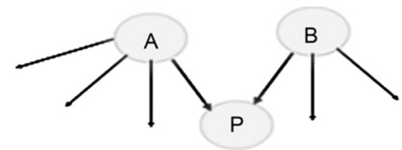

Fig. 1. A portion of a web graph: pages A and B point to page p and the out-degree for A and B is 4 and 3, respectively.

The PageRank formula in Eq. (1) is not suitable for disconnected web graphs, because it will not converge. Hence, the score of a page such as i ($R(i)$) can be approximated by the following recursive formula [7]:

$$R(i) = \left(d \times \sum_{j \in B(i)} R(j)/O(j)) + (1 - d/n \right) \tag{3}$$

where $R(i)$ and $R(j)$ show score of pages i and j, respectively. d is the damping factor, n is the total number of pages and $B(i)$ and $O(j)$ are the set of pages pointed to page i and the out-degree of the page j, respectively.

The presence of the damping factor is necessary, because the web graph is not a strongly connected graph (SCG), so damping factor used to guarantee the convergence of PageRank and remove the effects of sink pages (pages with no out-link).

2.2. BM25 algorithm

The BM25 formula was proposed by Robertson et al. [6]. In BM25, documents are ordered by decreasing probability of their relevance to the query. The formulation takes into account the number of times a query term appears in a document (tf), the proportion of other documents which contain the query term (idf), and the relative length of the document. A score for each document is calculated by summing the match weights for each query term [12]. Given a query Q, containing keywords $q_1, \ldots, q_n$; BM25 score of a document D is [6]:

$$S(D, Q) = \sum_{i=1}^{n} \frac{IDF(q_i)(f(q_i, D)(k_1 + 1))}{f(q_i, D) + k_1(1 - b + b(\left|D\right| / avgdl))} \tag{4}$$

where $f(q_i,D)$ is frequency of term q_i in the document D, $|D|$ is the length of the document in words, and $avgdl$ is the average of document's length in the text collection from which documents are drawn. k_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]$ and b is 0.75. $IDF(q_i)$ is the IDF (inverse document frequency) weight of the query term q_i. It is usually computed as

$$IDF(q_i) = \log \frac{(N - n(q_i) + 0.5)}{(n(q_i) + 0.5)} \tag{5}$$

where N is the total number of documents in the collection and $n(q_i)$ is the number of documents containing q_i.

3. The proposed algorithm

RL_Rank algorithm inspired from reinforcement learning concepts. So in this section, we first review reinforcement learning concepts. Afterwards, two proposed algorithms: RL_Rank and hybrid algorithm are introduced.

3.1. Reinforcement learning (RL)

Reinforcement learning, one of the machine learning techniques, learns by interactive in dynamic environment. Also, it is