Expert Systems with Applications 38 (2011) 11534–11537

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Optimizing search engines results using linear programming

Gholam R. Amin [a,*], Ali Emrouznejad [b]

[a] *Department of Computer Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran*
[b] *Aston Business School, Aston University, Birmingham, United Kingdom*

ARTICLE INFO

Keywords:
Linear programming
Web search engine
Optimization
Metasearch

ABSTRACT

When a query is passed to multiple search engines, each search engine returns a ranked list of documents. Researchers have demonstrated that combining results, in the form of a "metasearch engine", produces a significant improvement in coverage and search effectiveness. This paper proposes a linear programming mathematical model for optimizing the ranked list result of a given group of Web search engines for an issued query. An application with a numerical illustration shows the advantages of the proposed method.

1. Introduction

The World Wide Web (WWW) is a main place to find information about any area. Searching is a key activity on the Web and the major search engines such as Google, Live, Yahoo, etc. are the most frequently used tools for locating specific information on the vast expanse of the WWW. Several attempts have been reported in the literature to compare, rank and measure the performance of major search engines (Diaz, De, & Raghavan, 2005, 2007; Emrouznejad, 2008; Emrouznejad & Amin, 2010; Jansen & Spink, 2006). Many researchers have demonstrated that combining the results of multiple search engines in the form of a metasearch engine can significantly improve the search effectiveness (Bar-Ilan, Mat-Hassan, & Levene, 2006; Spink, Jansen, Blakely, & Koshman, 2006; Spoerri, 2007; Vaughan, 2004). Spink et al. (2006) studied the dispersion and overlap between the results of the major Web search engines. Spoerri (2007) investigated the ranking effects in search engine results and Vaughan (2004), Mowshowitz and Kawaguchi (2005) and Bar-Ilan et al. (2006) compared the results of several search engines. The results of different search engines show that only 45% of the relevant results are likely to be located by a single search engine and therefore combining the results of different search engines can significantly improve the results quality of the search engines (Keyhanipour, Moshiri, Kazemian, Piroozmand, & Lucas, 2007). All above studies concluded that search engines use different methods and the results of finding materials may also be differently ranked within them. Hence it would be impossible to find related information to the queries submitted to multi-search engines on the Web without a sophisticated method to combine the results and find the best related information. Consequently, finding relevant data on the Web in a timely and cost-effective way is a problem of wide interest and many believe that employing a single general-purpose search engine for all data on the Web is unrealistic (Höchstötter & Lewandowski, 2009; Lempel & Moran, 2004; Meng, Yu, & Liu, 2002; Mowshowitz & Kawaguchi, 2005). Moreover, researchers have demonstrated that combining results of different search engines produces a significant improvement in coverage and search effectiveness (Diaz et al., 2007; Höchstötter & Lewandowski, 2009). A metasearch engine is a system that supports unified access to multiple existing Web search engines. When a query is passed to a metasearch engine, the query is sent to a set of search engines, it then extracts the results from the returned pages, and aggregates them into a single ranked list (Diaz et al., 2005, 2007; Emrouznejad & Amin, 2010). Keyhanipour et al. (2007) and Emrouznejad (2008) used ordered weighted averaging (OWA) operator for aggregation of Web search engines. Within literature, no single research is reported to optimize the search engines results of a specific query using mathematical optimization theory. This paper aims to introduce a linear programming (LP) model to combine the results obtained in a metasearch. In summary the method first ranks the documents resulted for a specific query from each search engine then we use the state-of-art in linear programming to combine the rank and to find the optimal rank for each document in the search engines results. The originality of this study is that, for the first time the optimal results of Web search engines are analyzed using linear programming. Also, the proposed model finds the score of each document retrieved from a search engine using an optimization model and without including a subjective procedure. The rest of this paper is organized as follows. Section 2 gives a brief explanation of linear programming in general. Section 3 introduces a LP model for

[*] Corresponding author. Address: Department of Computer Engineering, Postgraduate Engineering Centre, Islamic Azad University, South Tehran Branch, Tehran, Iran, Postal Code: 1418765663. Tel.: +98 21 88347425.
E-mail address: G_Amin@azad.ac.ir (G.R. Amin).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.030

finding the optimal list of search engines results. This is followed by a numerical illustration in Section 4. A discussion on the results and advantage of using the proposed model is given in Section 5. Finally, remarks and conclusions are given in Section 6.

2. Linear programming problem

Linear programming (LP) deals with the problem of minimizing or maximizing a linear function in the presence of linear equality and/or inequality constraints or a set of restrictions (Bazarra, Jarvis, & Sherali, 2005; Vanderbei, 1997). Linear programming has proven to be an extremely powerful tool, both in modeling real-world problems and as a widely applicable mathematical theory. Thirteen of the Nobel Prize Laureates in “Economics” from 1969 to 1992 were authors or co-authors of papers or books in linear programming. Today LP has gained a wide range of successful applications for solving the real world problems and saved millions of dollars annually for business throughout the world. Consequently linear programming, as a powerful optimization tool, appeared in many fields of computer science such asautomatic control (Martins & Goncalves, 2009), for decoding LDPC codes (Burshtein, 2009), decision making (Chen, Liu, Chai, & Bao, 2009), image reconstruction (Tsuda & Rätsch, 2005), and many other applications. No one reported the use of linear programming for optimizing the Web search engines results. Mathematically a linear programming problem can be formulated in the following form (Bazarra et al., 2005)

$$\begin{aligned}
&\max\ z=\sum_{j=1}^{n}c_jx_j\\
&\text{s.t.}\\
&\quad\sum_{j=1}^{n}a_{ij}x_j\leqslant b_i\quad i-1,\ldots,m,\\
&\qquad x_j\geqslant 0\quad j-1,\ldots,n,
\end{aligned}\tag{1}$$

where, c_j is the jth coefficient of objective function, $x_j\geqslant 0$ is the jth non-negative decision variable (for each $j=1,\ldots,n$). Also a_{ij} ($i=1,\ldots,m, j=1,\ldots,n$) is called the technological coefficient. The inequality $\sum_{j=1}^{n}a_{ij}x_j\leqslant b_i$ denotes the ith constraint or restriction ($i=1,\ldots,m$) in the model. The next section shows how the problem of finding the optimal list of search engines results can be formulated as a linear programming problem.

3. Formulating search engines results

Suppose we have k search engines ($k\geqslant 2$) denoted by $SE_1,\ldots, SE_k$. Assume a specified query q consisting of some keywords or phrases is passed to the existing search engines and each of them returns the first m ($m\geqslant 2$) ranked list of documents as shown in Table 1.

Where, $D_i(j)$ denotes the document at jth place given by the ith search engine, $i=1,\ldots,k$ and $j=1,\ldots,m$. Now let’s assume we indicate the set of all documents given in Table 1 by

$$D=\{D_i(j):i=1,\ldots,k,\quad j=1,\ldots,m\}.$$

Table 1
The search engines results for query q.

Search engines /places	First place	...	jth place	...	mth place
SE_1	$D_1(1)$	...	$D_1(j)$	...	$D_1(m)$
⋮	⋮	⋮	⋮	⋮	⋮
SE_i	$D_i(1)$	...	$D_i(j)$	...	$D_i(m)$
⋮	⋮	⋮	⋮	⋮	⋮
SE_k	$D_k(1)$	...	$D_k(j)$	...	$D_k(m)$

Table 2
The place (rank) of the documents in the result list for each search engine.

Documents /places	First place	...	jth place	...	mth place
D_1	$N_1(1)$	...	$N_1(j)$	...	$N_1(m)$
...	...	...	...	...	...
D_l	$N_l(1)$	...	$N_l(j)$	...	$N_l(m)$
...	...	...	...	...	...
D_r	$N_r(1)$	...	$N_r(j)$	...	$N_r(m)$

Obviously it can be noted that the minimum and maximum cardinality of D is m and $k\times m$, respectively, that is $m\leqslant r=|D|\leqslant k\times m$.

The minimum cardinality occurs when the results of all search engines are the same and the maximum cardinality of D happens if each search engine gives a distinct list of documents. More precisely, the defined set D can be interpreted as the distinct documents of Table 1. That is

$$D=\{D_l:l=1,\ldots,r\},$$

where $D_l\neq D_p$ for all $l,p=1,\ldots,r, l\neq p$.

Define $N_l(j)$ as the number of search engines that give the lth document in the jth column (place) of Table 1, ($l=1,\ldots,r, j=1,\ldots,m$). As an example, if five search engines give the second document in the third place, then $N_2(3)=5$. Accordingly we construct Table 2.

Now the problem of finding the optimal ranked results of search engines can be expressed as follows:

Among the documents belonging to set D, determine the first mth ranked documents for the issued query q for which the returned ranked documents have the desirability with the issued query as much as possible. To formulate the problem as a linear programming model we define the decision variables w_j as the unknown weight corresponding to the jth column (or jth place, $j=1,\ldots,m$). Now we define the desirability index of the l_0th document by the following formula

$$z_{l_0}=\sum_{j=1}^{m}N_{l_0}(j)w_j,$$

where w_j is to be determined by the proposed model and z_{l_0} can be interpreted as the aggregate optimized rank for document l_0 to be placed in the jth order. We propose the linear programming model (2) to aggregate the optimal rank which will be given to document l_0 by all search engines. The program seeks to find an optimal weight vector $\mathbf{w}^*=(w_1^*,\ldots,w_m^*)$ that maximizes the desirability index of document l_0.

$$\begin{aligned}
&z_{l_0}^*=\max,\quad z_{l_0}=\sum_{j=1}^{m}N_{l_0}(j)w_j\\
&\quad\text{s.t.}\\
&\quad\sum_{j=1}^{m}N_l(j)w_j\leqslant 1\quad l=1,\ldots,r,\\
&\quad w_j-w_{j+1}\geqslant\varepsilon\quad j=1,\ldots,m-1,\\
&\quad w_m\geqslant\varepsilon,\\
&\quad w_j\geqslant 0\quad j=1,\ldots,m.
\end{aligned}\tag{2}$$

In this problem, the first type constraints bound the desirability index of each document l, $l=1,\ldots,r$, and ε in the second type constraints is a discriminating parameter between two adjacent weights, that is the weight is given to the document in place j must be no less than the weight is given to document in place $j+1$. These constraints, which we refer to as the weight restrictions, are added to the model for which the document is permitted to choose the most favorable weights to be applied to its rank (first place, second place, etc.). Therefore for obtaining the optimal ranked list results we need to solve the proposed linear programming models r times,