Application of PageRank Algorithm on Sorting Problem

Su weijun[1, a]

[1]Department of mathematics, Gansu normal university for nationalities, Hezuo, 747000, China

[a]962391696@qq.com

Keywords: PageRank, Tournament, Random matrix, Eigenvector

Abstract. In the social life, people often face a wide range of scheduling problems, such as appraisal of excellence, race rankings. These are often hot and sensitive issues, so the research of this type of problems has practical and economic value. But due to socio-cultural diversity and pluralism of values, sorting problems would be restricted on a set of priority level or justice principles. As well-known Arrow theory told us: in a certain sense, there are not a set of the justice axiom to satisfy the so called sort rules. This article is stimulated by Google's PageRank algorithm and the example of national college mathematical modeling contest problem B in 1993. Constructing random matrix shows the application of PageRank algorithm on sorting problem.

Introduction

In modern life, whether individual or collective often encounter the problem of ranking for a class of objects, such as appraisal of excellence, race rankings, evaluation of students' scholarship, electric, coal and oil arrangement during snowstorm period. These problems are often hot and sensitive issues, and research on these issues has some economic value and social value. But due to the diversity of social and cultural life and pluralism of values, sorting problems would be restricted to a set of priority level or justice principles. As well-known Arrow theory told us: in a certain sense, there does not a set of the justice axiom to satisfy the so called sort rules. In this case, analytical hierarchy process [1] and tournament methods [2] are often applied to this case, but the first methods must construct a matrix by expert system, structural consistency of judgement matrix in practice for a long time, and the data with poor quality so that can't satisfy certain requirements. The second methods require tournaments bidirectional connectivity, construct the adjacency matrix for data with harsh terms, this situation makes people confused to deal with the sorting or scheduling problems. The work is stimulated by Google's PageRank algorithm, by the example of 1993 national college mathematical modeling contest problem B, constructing random matrix shows the application of PageRank algorithm on sorting problem. This approach, with few rules, simple and efficient calculations, and sort results easily accepted by all, is a portable application.

About PageRank

Pagerank is a method that Google used to test the importance of a Web page. Google provides search engine features. Google's success comes not only from the wealth of complete information, but also through a number of revolutionary new technology. This text includes sophisticated technology and advanced PageRank sorting technology which has won user's love.

The basic idea of PageRank is mostly made up of literature citations analysis of traditional philology, that is, the more one article is cited by others, the higher the quality is. Specifically, the PageRank technology is based on the network structure, and the link structure of the Web itself. In essence, when you link from page *A* to page *B*, Google considers "page *A* voted for page *B*". Google evaluates the importance of a web page according to its vote [3]. However, in addition to considering Web votes links outside of pure quantity, Google will also be analyzed for their vote on the Web page, because the vote from important pages is naturally essential. Important and high quality page will get a higher page rank and get higher ranking in search results. This composite index of the importance of Google PageRank will be in the level of web page instead of basing on specific search.

In short, Google achieves the ranking of Web pages in its search results pages by following few steps:

(1) Find all websites and browse pages with a public entrance;

(2) Data index page in order to efficiently find matching keywords or phrases;

(3) Determine the importance level for all web pages so that users are able to find the information they want when they search on internet, and the more important pages should rank in front.

Thus PageRank can ensure the objectivity and fairness of the ranking. The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.

Assume a small universe of four web pages: *A*, *B*, *C* and *D*. Links from a page to itself, or multiple outbound links from one single page to another single page, are ignored. PageRank is initialized to the same value for all pages. In the original form of PageRank, the sum of PageRank over all pages was the total number of pages on the web at that time, so each page in this example would have an initial value of 1. However, later versions of PageRank, and the remainder of this section, assume a probability distribution between 0 and 1. Hence the initial value for each page is 0.25.

The PageRank transferred from a given page to the targets of its outbound links upon the next iteration is divided equally among all outbound links.

If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 PageRank to A upon the next iteration, for a total of 0.75.

$$PK(A) = PK(B) + PK(C) + PK(D) \tag{1}$$

Suppose instead that page *B* had a link to pages *C* and *A*, page *C* had a link to page *A*, and page *D* had links to all three pages. Thus, upon the first iteration, page *B* would transfer half of its existing value, or 0.125, to page *A* and the other half, or 0.125, to page *C*. Page *C* would transfer all of its existing value, 0.25, to the only page it links to, *A*. Since *D* had three outbound links, it would transfer one third of its existing value, or approximately 0.083, to *A*. At the completion of this iteration, page *A* will have a PageRank of 0.458.

$$PK(A) = \frac{1}{2}PK(B) + PK(C) + \frac{1}{3}PK(D) \tag{2}$$

In other words, the PageRank conferred by an outbound link is equal to the document's own PageRank score divided by the number of outbound links out.

$$PK(A) = \frac{1}{out(B)}PK(B) + \frac{1}{out(C)}PK(C) + \frac{1}{out(D)}PK(D) \tag{3}$$

In the general case, the PageRank value for any page *u* can be expressed as:

$$PK(u) = \sum_{v \in B_u} \frac{PK(u)}{out(v)} \tag{4}$$

i.e. the PageRank value for a page u is dependent on the PageRank values for each page v contained in the set Bu (the set containing all pages linking to page u), divided by the number out(v) of links from page v.

The meaning of formula (4) is self-explanatory without the need of any further comment, a page *u* level value or importance value, score by a link to its page *v* provides, and the more the links Point(*u*) the higher level value; the bigger *v* out the number of links Out(*v*) the smaller *v* contribution to scoring *u*.

In *n* Web page problem, we construct a directed graph of *n* vertices to study, usually expressed as a linked matrix. Link matrix is defined as follows:

$$P_{uv} = \begin{cases} \frac{1}{Out(v)}, u \text{ linked to } v; \\ 0, u \text{ not linked to } v. \end{cases} \tag{5}$$

Clearly, matrix *P* is a random matrix all elements are non-negative, and the squares of the elements in each column sum to 1. Random matrices have very good column properties: random matrix *P* with eigenvalues 1, namely, the existence of non-zero vectors *x*, *Px* = *x*.

As two matrices of a matrix and it's transposes with the same eigenvalue, of n-rank random matrix *P* and *n*-dimensional non-zero vectors *x* = (*a*, *b*, ..., *c*), there is a clear $P^T x = x$.

This eigenvector belonging to the eigenvalue 1 is the level vector of theoretical problems in the sorting problem (or an important value, score), which is also our priority vector.

Technique and Example of Sorting Problem

In practice, PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor *d*. Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85. The damping factor is subtracted from 1 (and in some variations of the algorithm, the result is divided by the number of documents (*n*) in the collection) and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores.

When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection. Their PageRank scores are therefore divided evenly among all other pages. In other words, to be fair with pages that are not sinks, these random transitions are added to all nodes in the Web, with a residual probability usually set to *d* = 0.85, estimated from the frequency that an average surfer uses his or her browser's bookmark feature.

The eigenvector belonging to the eigenvalue 1 is not necessarily unique, which caused the sort results to be not unique. The PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor *d*.

The damping factor is subtracted from 1 (and in some variations of the algorithm, the result is divided by the number of documents (*N*) in the collection) and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores. That is,

$$PK(A) = \frac{1-d}{N} + d\left(\frac{1}{out(B)}PK(B) + \frac{1}{out(C)}PK(C) + \frac{1}{out(D)}PK(D) + \ldots\right) \tag{6}$$

So any page's PageRank is derived in large part from the PageRanks of other pages. The damping factor adjusts the derived value downward. If a page has no links to other pages, it becomes a sink and therefore terminates the random surfing process. If the random surfer arrives at a sink page, it picks another URL at random and continues surfing again.

When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection. Their PageRank scores are therefore divided evenly among all other pages. In other words, to be fair with pages that are not sinks, these random transitions are added to all nodes in the Web, with a residual probability usually set to d = 0.85, estimated from the frequency that an average surfer uses his or her browser's bookmark feature. So, the equation is as follows:

$$PK(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PK(p_i)}{out(p_j)} \qquad (7)$$

where, p_1, p_2, …, p_N are the pages under consideration, $M(p_i)$ is the set of pages that link to $p_{i,}$ $out(p_j)$ is the number of outbound links on page p_j, and N is the total number of pages.

The PageRank values are the entries of the dominant right eigenvector of the modified adjacency matrix. This makes PageRank a particularly elegant metric, R is the solution of the equation

$$R = \begin{pmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{pmatrix} + d \begin{pmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,N} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N,1} & e_{N,2} & \cdots & e_{N,N} \end{pmatrix} R \qquad (8)$$

where, the adjacency function $e_{i,j}$ is 0 if page p_j does not link to p_i, and normalized such that, for each j that

$$\sum_{i=1}^{N} e_{i,j} = 1 \qquad (9)$$

i.e. the elements of each column sum up to 1, so the matrix is a stochastic matrix (for more details see the computation section below). Thus we construct a positive matrices M (all elements in the array is a positive number) replaced m-matrix P, the equation is as follows:

$$M = dP + (1 - d)S \qquad (10)$$

Matrix S for which all the elements is reciprocal of the matrix rank number, and $0 \le d \le 1$. Google calculated by d = 0.85. Clearly, S is a random matrix, M is a positive random matrix and eigenvector belonging to the eigenvalue 1 of M component of all positive or all negative [4, proposition 2].

According to the famous Perron-Frobenius theorem [5], M of positive random matrix ensures its $0 < d \le 1$ eigenvectors belonging to the eigenvalue 1 must be unique. Subspace belonging to the eigenvalue 1 is a one-dimensional invariant subspace. Commonly calculating the eigenvectors of M iterative algorithm, such as singular value decomposition method and power method [5] in practice.

For example, the national mathematical modeling contest in problem B in 1993 illustrated the application of PageRank on sorting problem.

As we all know, football match makes a good or poor score according to victory points, win three points in a win match and enough in a lose match; if the scores are equal, compared to the outcome of the match between them, then compared to net goals and all goals each other, as shown in the most occasion, it often can't determine the position of the teams. But we firmly believe that a football game to score as the climax, players, spectators hope that their team scores more goal as well as existing rules will encourage.

(1) Statistic score at a match and the match data of teams conceded, we deal with data simply obtained link matrix.

Table 1. Match data of 12 teams.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	win score	match number
T1	-	1:1	3:4	6:1	3:1	1:0	1:4	2:3	5:0	2:2	+	+	24	19
T2	1:1	-	3:4	2:0	1:1	2:1	2:2	0:0	3:1	0:2	+	+	14	19
T3	4:3	4:3	-	5:3	2:1	3:0	2:4	3:2	3:3	2:1	+	+	28	19
T4	1:6	0:2	3:5	-	2:3	0:1	2:8	3:4	0:1	1:2	+	+	12	19
T5	1:3	1:1	1:2	3:2	-	0:1	+	+	+	+	2:2	1:1	9	9
T6	0:1	1:2	0:3	1:0	1:0	-	+	+	+	+	+	+	3	5
T7	4:1	2:2	4:2	8:2	+	+	-	3:0	6:1	8:3	3:1	2:0	40	19
T8	3:2	0:0	2:3	4:3	+	+	0:3	-	3:3	2:2	3:1	0:0	17	19
T9	0:5	1:3	3:3	1:0	+	+	1:6	3:3	-	4:0	1:0	1:0	15	19
T10	2:2	2:0	1:2	2:1	+	+	3:8	2:2	0:4	-	1:0	2:0	15	19
T11	+	+	+	+	2:2	+	1:3	1:3	0:1	0:1	-	3:4	7	9
T12	+	+	+	+	1:1	+	0:2	0:0	0:1	0:2	4:3	-	5	9
lose score	16	12	20	32	12	6	12	17	20	19	14	9		

(2) Constructing random matrix whose elements is ratio of the win score and the lose score, obtained the matrix is:

$$P = \begin{pmatrix}
0 & \frac{1}{12} & \frac{3}{20} & \frac{3}{16} & \frac{1}{4} & \frac{1}{6} & \frac{1}{12} & \frac{2}{17} & \frac{1}{4} & \frac{2}{19} & 0 & 0 \\
\frac{1}{16} & 0 & \frac{3}{20} & \frac{1}{16} & \frac{1}{12} & \frac{1}{3} & \frac{1}{6} & 0 & \frac{3}{20} & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{3} & 0 & \frac{5}{32} & \frac{1}{6} & \frac{1}{2} & \frac{1}{6} & \frac{3}{17} & \frac{3}{20} & \frac{2}{19} & 0 & 0 \\
\frac{1}{16} & 0 & \frac{3}{20} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{3}{17} & 0 & \frac{1}{19} & 0 & 0 \\
\frac{1}{16} & \frac{1}{12} & \frac{1}{20} & \frac{3}{32} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} & \frac{1}{9} \\
0 & \frac{1}{12} & 0 & \frac{1}{32} & \frac{1}{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{6} & \frac{1}{5} & \frac{1}{4} & 0 & 0 & 0 & \frac{3}{17} & \frac{3}{10} & \frac{8}{19} & \frac{3}{14} & \frac{2}{9} \\
\frac{3}{16} & 0 & \frac{1}{10} & \frac{1}{8} & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{2}{19} & \frac{3}{14} & 0 \\
0 & \frac{1}{12} & \frac{3}{20} & \frac{1}{32} & 0 & 0 & \frac{1}{12} & \frac{3}{17} & 0 & \frac{4}{19} & \frac{1}{14} & \frac{1}{9} \\
\frac{1}{8} & \frac{1}{6} & \frac{1}{20} & \frac{1}{13} & 0 & 0 & \frac{1}{4} & \frac{2}{17} & 0 & 0 & \frac{1}{14} & \frac{2}{9} \\
0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{12} & \frac{1}{17} & 0 & 0 & 0 & \frac{1}{3} \\
0 & 0 & 0 & 0 & \frac{1}{12} & 0 & 0 & 0 & 0 & 0 & \frac{2}{7} & 0
\end{pmatrix}$$

(3) Transformation, use $M = dP + (1 - d)S$ instead of P, and $d = 0.85$, S all the elements $\frac{1}{12}$. In order to get answers to our questions, we use basic computer algebra system Maple 15 solution of eigenvalue and eigenvector of a matrix, made simpler:

With(linalg):

P:=matrix(12, 12, [0, $\frac{1}{12}$, $\frac{1}{12}$,…, 0, $\frac{2}{7}$, 0]):

S:=matrix(12, 12, [$\frac{1}{12}$, $\frac{1}{12}$, $\frac{1}{12}$,…, $\frac{1}{12}$, $\frac{1}{12}$, $\frac{1}{12}$]):

M:=evalm(0.85*P + 0.15*S);

eigenvects(M);

Find the eigenvector belonging to the eigenvalue 1, operating eigenvalue is 1.000000004, after the corresponding eigenvector and its unitize of weight vector that is listed, as the following Table 2:

Table 2 Weight vector.

Team	Component of eigenvector	Component of unitized eigenvector	Team rank
T1	0.557046704	0.110688366	3
T2	0.436287815	0.086692884	5
T3	0.727039964	0.144466999	2
T4	0.422221014	0.08389773	7
T5	0.226210013	0.044949223	9
T6	0.121049273	0.024053183	12
T7	0.868742048	0.172624013	1
T8	0.397147086	0.078915397	8
T9	0.433468487	0.086132667	6
T10	0.49920697	0.09919528	4
T11	0.213393883	0.042402585	10
T12	0.130754535	0.025981674	11

Compare with the know results [2], they have very good consistency, the correlation coefficient of two results is 0.811188811, especially the top three rankings are exactly the same. For the last bit of the rankings, there is no unity. Because T6 only 5 matches and rarely win but more lose, mainly did not match T7, which is explained easily by PageRank algorithm.

Conclusion

In the modern life, people often have to compared several object and sort them. Due to the deficiency of the needed data, people often consider using expert system method, but the method would last a long period of time, and with a high cost, but an important question that conclusions does not acceptable by the each participator, the method in the paper not only the sufficient on theory but also manipulate simpler in practice, and importantly, operation conclusion guarantee uniqueness, this means the method convenient, and it has promotion value in practice.

Acknowledgement

This subject supported by president scientific research fund(approved no. 11-01), and fundamental mathematics constructed fund of Gansu Normal University for Nationalities.

References

[1] Q. U. Jiang, J. X. Xie, Y. Jun, Mathematical Modeling, Beijing: Higher education press, 3rd Edition, 2003.

[2] L. F. Qi, W. Mao, B. Ma, One method of ranking football team: B problem in CUMCM 1993, Pract. Knowl. Math. 2 (1994) 86-94.

[3] Information on http://mcm.edu.cn/mcm92_96/cumcm1993 problems.pdf

[4] K. Bryan T. Leise, The $25,000,000,000 EigenVector: The Linear Algebra behind Google. http://www.rose-hulman.edu/~bryan/googleFinalVersionFixed.pdf

[5] M. T. Heath, Introduction to scientific calculate, Beijing: Tsinghua U. Press, 2001.