Computer Networks 33 (2000) 449–455

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

An efficient algorithm to rank Web resources

Dell Zhang *, Yisheng Dong

Department of Computer Science and Engineering, Southeast University, Nanjing, 210096, China

Abstract

How to rank Web resources is critical to Web Resource Discovery (Search Engine). This paper not only points out the weakness of current approaches, but also presents in-depth analysis of the multidimensionality and subjectivity of rank algorithms. From a dynamics viewpoint, this paper abstracts a user's Web surfing action as a Markov model. Based on this model, we propose a new rank algorithm. The result of our rank algorithm, which synthesizes the relevance, authority, integrativity and novelty of each Web resource, can be computed efficiently not by iteration but through solving a group of linear equations.

Keywords: Web; Resource discovery; Search engine; Rank algorithm

1. Introduction

The World Wide Web is rapidly emerging as an important medium for the dissemination of information related to a wide range of topics [1]. There are about 300 million pages on the Web today with about 1 million being added daily. According to most predictions, the majority of human information will be available on the Web in 10 years. But, it is widely believed that 99% of the information on the Web is of no interest to 99% of the people. Looking for something valuable in this tremendous amount of information is as difficult as looking for a needle in a haystack.

Searching for valuable information on the Web is called resource discovery (RD). The IETF-RD group argues that resource discovery should provide the user a consistent, organized view of information [15]. In a typical RD procedure, the user submits a query Q, which is simply a list of keywords (with some additional operators), to the RD server (RDS), then RDS returns a set of related Web page URLs: R_1, R_2, ... R_n. There are many search engines to support RD on the Web, such as **Yahoo!** [1], **AltaVista** [2], **Excite** [3], **Hotbot** [4], **Infoseek** [5], etc. A search engine usually collects Web pages on the Internet through a robot (spider, crawler) program, then these Web pages are automatically scanned to build giant indices, so you can quickly retrieve the set of all Web pages containing the given keywords.

RD on the Web is especially difficult due to the following five characteristics of the Web data source: (1) huge and ubiquitous; (2) mostly semistructured or unstructured; (3) diverse in quality; (4) dynamic; (5) distributed and autonomous. In particular, a topic of any breadth will typically contain several thousand or

* Corresponding author. E-mail: dell.z@ieee.org

[1] http://www.yahoo.com
[2] http://altavista.digital.com
[3] http://www.excite.com
[4] http://www.hotbot.com
[5] http://www.infoseek.com

1389-1286/00/$ – see front matter
PII: S1389-1286(00)00056-6

million relevant Web pages. For instance, if you enter the search engine AltaVista, input 'data mining', over 50,000 Web pages will be found. Yet, a user will be willing, typically, to look at only a few of these pages.

The rank algorithm, can then help a user to select the *correct* ones (those of most value to him or her), from this sea of Web resources. Given a Web resource r and a user's query q, the rank algorithm will compute a score rank(r, q). The bigger rank(r, q) is, the more valuable r to q, i.e., the more valuable for the user. In practice, RD on the Web can be viewed as fuzzy queries driven by the rank algorithm, but not SQL-style precise queries. Moreover, a good rank algorithm is very helpful to crawl the Web more efficiently [7]. Meta-search engines also need the rank algorithm to synthesize the results from other search engines [2,10]. All in all, we argue that the rank algorithm is the core technique of a search engine, and it is critical to improve the quality of RD on the Web.

This paper is organized as follows. Section 2 points out the weakness of current approaches in ranking Web resources. Section 3 presents in-depth analysis of the multidimensionality and subjectivity of rank algorithms. Section 4 abstracts a user's Web surfing action as a Markov model and we propose a new rank algorithm based on this model. Section 5 concludes this paper and discusses future work.

2. State of the art

At the present time, most rank algorithms of Web resources are using the similarity measure based on the vector-space model, which has been well studied by the Information Retrieval (IR) community. To compute the similarities, we can view each document as an n-dimensional vector $\langle w_1, \ldots, w_n \rangle$. The term w_i in this vector represents the ith word in the vocabulary. If w_i does not appear in the document, then w_i is zero. If it does appear, w_i is set to represent the significance of the word. One common way to compute the significance w_i is TF $\times$ IDF, i.e., to multiply the number of times the ith word appears in the document (TF) by the inverse document frequency (IDF) of the ith word. The IDF factor is 1 divided by the number of times the word appears in the entire 'collection', which in this case would be the entire Web. The IDF factor corresponds to the content discriminating power of a word: a term that appears rarely in documents (e.g., 'algebra') has a high IDF, while a term that occurs in many documents (e.g., 'the') has a low IDF. The similarity between query Q and document R can then be defined as the inner product of the vectors of Q and R. Another option is to use the cosine similarity measure, which is the inner product of the normalized vectors. The w_i terms can also take into account where on a HTML page the word appears, for instance, words appearing in the title may be given a higher weight than other words in the body [6]. Along with the popularization of Web meta-data standards such as RDF, it becomes feasible to take advantage of the meta-data of Web resources, which can also improve the rank algorithm's accuracy [13].

But the rank algorithms derived from IR have lots of limitations, as they only evaluate the content, but totally neglect the quality of Web resources. So these rank algorithms can be easily cheated. Webmasters can make their sites highly ranked through inserting some irrelevant but popular words (e.g., 'Clinton', 'sex') into important places (e.g., title page) or meta-data. This phenomenon is called Search Engine Persuasion (SEP) or Web Spamming [12,14].

Recent researches in this area concentrate on mining the linkage structure of Web resources to support RD on the Web [3,5,7,12,16]. A typical one in such rank algorithms is PageRank [3], which is proposed by the Stanford University and has been applied in the famous search engine **Google** [6]. The PageRank metric, PR(P), recursively defines the importance of a page P to be the weighted sum of the back-links to it. Such a metric has been found to be very useful in ranking results of user queries. More formally, if a page has no outgoing link, we assume that it has outgoing links to every single page. Consider a page P that is pointed at by pages $T_1, T_2, \ldots, T_n$. Let $C(T_i)$ be the number of links going out of page T_i. Also, let d be a damping factor. Then, the weighted back-link count of page P is given by

$$\mathrm{PR}(P) = (1-d) + d(\mathrm{PR}(T_1)/C(T_1) + \cdots + \mathrm{PR}(T_n)/C(T_n)).$$

[6] http://www.google.com/

This leads to one equation per Web page, with an equal number of unknowns. The equations can be solved iteratively, starting with all PR values equal to 1. At each step, the new PR(P) value is computed from the old PR(T_i) values (using the equation above), until the values converge. Through the famous Perron–Frobenius Theorem [18], we can find out that this calculation corresponds to computing the principal eigenvector of the linkage matrix.

Theorem 1 (Perron–Frobenius Theorem). *If an n-dimensional matrix $\boldsymbol{A}$ is positive or non-negative irreducible, then:*

- *the spectrum radius of $\boldsymbol{A}$, ρ, is also a latent root of $\boldsymbol{A}$;*
- *there is a positive eigenvector of $\boldsymbol{A}$ corresponding to ρ;*
- *the eigenfunction of $\boldsymbol{A}$ has a single root ρ, i.e., $mult\rho(\boldsymbol{A}) = 1$,*
- ... ...

Although the rank algorithms based on linkage structure break through the limitation of traditional IR technology, they still have some shortcomings:

- only the authority metric has been taken into account;
- the iterative computation results in bad performance;
- it is difficult to deal with the overflow or underflow problem during iteration;
- because of the 'rank sink problem', the iterative computation may not converge.

The last situation, 'rank sink problem', is illustrated in Fig. 1. Consider two Web pages that point to each other but to no other page, and suppose there is some Web page that points to one of them. During iteration, this loop will accumulate rank but never distribute any rank (since there are no outgoing edges). The loop forms a sort of trap called a 'rank sink' [3]. This problem can be analyzed more formally as follows.

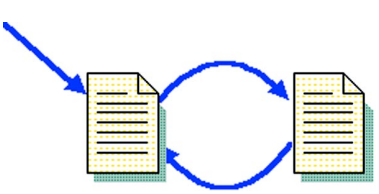

Fig. 1. The 'rank sink problem'.

Definition 1. $\boldsymbol{A} = (a_{ij})_{n\times n}$ is an n-dimensional matrix, the graph, $D(\boldsymbol{A}) = \{(i, j) \mid a_{ij} \neq 0\}$ is named the adjoint directed graph of $\boldsymbol{A}$.

Theorem 2. *$\boldsymbol{A}$ is an n-dimensional non-negative matrix, so $\boldsymbol{A}$ is irreducible, if and only if, the adjoint directed graph of $\boldsymbol{A}$ is strongly connected (there exists a path from u to v for any node u and v in the graph) [18].*

It is obvious that the 'rank sink problem' makes the Perron–Frobenius Theorem no longer applicable, so the iteration computation method loses its foundation.

To sum up, there are still many weaknesses of current rank algorithms. Simply inheriting IR technology and merely mining the linkage structure are not enough.

3. Analysis of the rank algorithm

3.1. Definition

The rank function of Web resources can be formally defined as rank : $R \times Q \to \mathbf{R}^+ \cup \{0\}$, here R represents the set of relevant Web resources, Q represents the set user's queries. Without loss of generality, we can view R as the relevant Web pages found by the search engine. Given $\forall r \in R, \forall q \in Q$, the bigger rank$(r, q)$, the more valuable r to q, i.e., the more valuable for the user.

If the function 'rank' satisfies

(1) $\forall r \in R, \forall q \in Q,\ \ 0 < \text{rank}(r, q) < 1$,

(2) $\forall q \in Q,\ \ \sum_{r\in R} \text{rank}(r, q) = 1$,

then it is called a normal rank function. Since all rank functions can be transformed to equivalent normal rank functions, we assume all rank functions are normal in the following discussion.

3.2. Multidimensionality

In our opinion, a rational rank algorithm of Web resources should be multidimensional, at least it should include the following metrics.

- *Relevance*. The relevance metric means the distance between the content of a Web resource r and a user's query q. It is also the metric used by most search engines. The normalized relevance function can be defined as corr : $R \times Q \to [0, 1]$.

As stated in Section 2, the method to calculate relevance can be derived from IR technology, based on TF, IDF, word weight, meta-data, etc.

- *Authority*. The authority metric means how many Web resources refer to the Web resource r. Moreover, the Web resources referred to by higher-quality resources should be assigned with higher authority.
- *Integrativity*. The integrativity metric means how many Web resources are pointed by the Web resource r. Moreover, the Web resources pointed to higher-quality resources should be assigned with higher integrativity. A Web resource with high integrativity is just like a good 'survey' or 'review' style academic paper, it can lead users to valuable information.
- *Novelty*. The novelty metric means in which degree the Web resource r is different from others, i.e., provide novel information. Analysis of query logs has demonstrated that users are impatient, rarely examining more than the first page of results (usually displaying 7–12 URLs). Hence we hope that the top 30 Web resources will be very representative. Such Web resources should have few direct links between themselves, because they will act as roadmaps so that users can easily follow the links embedded in them to find other valuable resources.

3.3. Subjectivity

The value of a Web resource r depends not only on the query q, but also on the user's nation, age, gender, career, culture, hobby, etc. So there is no absolute best rank function. But we can still evaluate the rank algorithm based on the user's reaction.

Assuming 'RANK' represents the set of all possible rank functions, the user's satisfaction function can be defined as sat: $Q \times \mathrm{RANK} \to [0,1]$, sat($q$, rank) will be proportional to the user's satisfaction of this rank function. Given the Web sources set $R = \{r_1, r_2, \ldots, r_n\}$, without loss of generality, we suppose $r_1, r_2, \ldots, r_n$ are decreasingly ordered by their value according to the user's judgement. Define the 'reverse order number' of the Web resource r_i under the rank function as

$$\psi(r_i) = |\{r_k \mid (r_k \in R) \wedge (1 \le k < i) \wedge (\mathrm{rank}(r_k) < \mathrm{rank}(r_i))\}|$$

then

$$\mathrm{sat}(q, \mathrm{rank}) = \frac{\sum_{i=1}^{n} \psi(r_i)}{(n-1)(n-2)/2}.$$

Given the queries set $Q = \{q_1, q_2, \ldots, q_m\}$, the average satisfaction function should be

$$\mathrm{sat}(Q, \mathrm{rank}) = \frac{\sum_{j=1}^{m} \mathrm{sat}(q_j, \mathrm{rank})}{m}.$$

4. The rank algorithm based on Markov model

The above rank algorithms of the Web resources are all from a statistic approach, but this paper presents a user-centered rank algorithm from a dynamics viewpoint. In fact, surfing on the Web can be viewed as a dynamic procedure in that a user jumps from one Web resource to another. We abstract this surfing procedure as a Markov chain.

Definition 2. Suppose $\{x_t, t \ge 0\}$ is a series of random variables on a finite state space $S = \{s_1, s_2, \ldots, s_n\}$. If the state of x_{k+1} only depends on x_k, but not on $x_0, x_1, \ldots, x_{k-1}$, i.e., for any $k \ge 0$ and positive integers $i_0, i_1, \ldots, i_k, i_{k+1}$ the equation $P(x_{k+1} = s_{i_{k+1}} \mid x_0 = s_{i_0}, \ldots, x_k = s_{i_k}) = P(x_{k+1} = s_{i_{k+1}} \mid x_k = s_{i_k})$ is always true, then $\{x_t, t \ge 0\}$ is named a finite Markov chain. $P(x_{k+1} = s_j \mid x_k = s_i)$, $p_{ij}(t)$ for short, represents the probability to transit from the state s_i to s_j in the time t. If the transition probability $p_{ij}(t)$ is independent of t, i.e., for any $s_i, s_j \in S$ and any t_1, t_2, $p_{ij}(t_1) = p_{ij}(t_2)$, then these types of Markov chains are called homogeneous ones. $\boldsymbol{P} = (p_{ij})_{n\times n}$ is the transition probability matrix of this homogeneous Markov chain, where $p_{ij} \in [0,1]$, $1 \le i, j \le n$, $\sum_{j=1}^{n} p_{ij} = 1$, $1 \le i \le n$.

For a query q, $R = \{r_1, r_2, \ldots, r_n\}$ denotes the set of related Web resources found by the search engine. We can use R as the state space (one Web resource corresponds to one state). And then we consider a virtual user surfing on the Web, in time t; he is browsing the Web resource r_i in probability $p_i(t)$, and will jump to the Web resource r_j in probability

p_{ij}. It is in this way that the user's surfing action on the Web can be abstracted as a homogeneous Markov chain. Although this modeling is rather simple and intuitive, we believe that it has grasped the spirit of the surfing procedure.

Definition 3. A finite Markov chain's distribution vector in time t is $\boldsymbol{p}(t) = (p_1(t), p_2(t), \ldots, p_n(t))$, where $p_j(t) = P(x_t = s_j)$, $p_j(t) \in [0,1]$, $\sum_{j=1}^{n} p_j(t) = 1$.

Theorem 3. *A homogeneous Markov chain's behavior can be determined by its initial distribution vector* $\boldsymbol{p}(0)$ *and its transition probability matrix* $\boldsymbol{P}$, $\boldsymbol{p}(t) = \boldsymbol{p}(0)\boldsymbol{P}^t$.

Suppose in time t, the virtual user is in state r_i, i.e., browsing the Web resource r_i, then in the next time $t+1$, he may have the following choices:

- continue browsing the Web resource r_i;
- click a hyperlink in r_i and jump to a new Web resource;
- press the 'BACK' button in the browser and return to the last browsing Web resource;
- select another Web resource from the results of the search engine, R.

Facing each of the above choices, the virtual user's tendencies are measured as $\alpha \times \mathrm{sim}(r_i, q)$, β, γ and ε separately, where $\mathrm{corr}(r_i, q)$ is the relevance function defined in Section 2, and $\alpha, \beta, \gamma, \varepsilon$ are four constants to a specific user, which satisfy the condition $0 < \alpha, \beta, \gamma, \varepsilon < 1$, $\alpha + \beta + \gamma + \varepsilon = 1$.

Definition 4. The linkage structure graph of the Web resources, $G = (V, E)$, is a directed graph, where V is the set of nodes ($|V| = n$) (a node corresponds to a resource in R), and E is the set of edges, $E = \{(v_i, v_j) \mid v_i, v_j \in V \text{ and } r_i \text{ points to } r_j \text{ through hyperlink}\}$, $\mathrm{in}(v_i) = |\{(v_k, v_i) \mid (v_k, v_i) \in E\}|$, $\mathrm{out}(v_i) = |\{(v_i, v_k) \mid (v_i, v_k) \in E\}|$.

Definition 5. The tendency matrix for the set of related Web resources, R, is

$$\boldsymbol{U} = (u_{ij})_{n\times n}, \quad u_{ij} = \begin{cases} \alpha \times \mathrm{sim}(r_i, q) & \text{if } i = j, \\ \beta & \text{if } (v_i, v_j) \in E, \\ \gamma & \text{if } (v_j, v_i) \in E, \\ \varepsilon & \text{otherwise.} \end{cases}$$

Note that the tendency matrix here has synthesized the four metrics mentioned above (relevance, authority, integrativity and novelty).

After normalizing the tendency matrix, we get the transition probability matrix in the Markov model for user's surfing procedure on the Web.

Theorem 4. *The transition probability matrix for the set of related Web resources, R, is*

$$\boldsymbol{P} = (p_{ij})_{n\times n}, \qquad p_{ij} = \frac{u_{ij}}{\sum_{j=1}^{n} u_{ij}}.$$

Through Theorems 3 and 4, the probability distribution vector at any time t can be calculated easily. Now it is obvious that $\alpha, \beta, \gamma, \varepsilon$ actually reflect the relative importance, in the user's opinion, of the relevance, authority, integrativity and novelty metrics. So we also call α the relevance parameter, β the authority parameter, γ the integrativity parameter, and ε the novelty parameter.

Definition 6. A homogeneous Markov chain, on the state space $S = \{s_1, s_2, \ldots, s_n\}$, is holomorphic, if for any $\forall s_i, s_j \in S$ there exists a positive integer k, the state can transit from s_i to s_j in a positive probability, within k steps.

Definition 7. $\boldsymbol{A} = (a_{ij})_{n\times n}$ is an n-dimensional positive matrix.

(1) $\boldsymbol{A}$ is called stochastic when $\sum_{j=1}^{n} a_{ij}(t) = 1$, $i = 1, 2, \ldots, n$.
(2) $\boldsymbol{A}$ is called primitive when there exists a positive integer k, $\boldsymbol{A}^k > 0$ (that is, every element in $\boldsymbol{A}^k$ is positive).

It is obvious that multiplying several stochastic matrices produces a stochastic matrix, and every positive matrix is sure to be primitive.

Theorem 5. *A homogeneous Markov chain, on the state space* $S = \{s_1, s_2, \ldots, s_n\}$, *is holomorphic, if and only if, its transition probability matrix* $\boldsymbol{P}$ *is a primitive stochastic matrix.*

Based on Theorems 4 and 5, it is easy to discover that the Markov chain corresponding to the user's surfing procedure is a holomorphic Markov chain.

Theorem 6. *A holomorphic and homogeneous Markov chain,* $\{x_t, t \geq 0\}$*, with* $S = \{s_1, s_2, \ldots, s_n\}$ *as its state space,* $\boldsymbol{P}$ *as its transition probability matrix, and* $\boldsymbol{p}(0)$ *as its initial distribution vector, will converge to a unique ultimate distribution when* $t \to \infty$*, that is*

$$\lim_{t\to\infty} \boldsymbol{p}(t) = \boldsymbol{\pi}$$

The ultimate (stable) distribution vector, $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$, is the unique solution of the equation $\boldsymbol{\pi P} = \boldsymbol{\pi}$ that satisfies $\pi_i > 0$, $\sum_{i=1}^{n} \pi_i = 1$ [8,9].

Assuming that the virtual user is rational and experienced enough, we argue that the probability of browsing the Web resource r_i should be proportional to its worthiness. In practice, we can also establish out that a user usually spends more time on the Web resources he cares for most. At first, the user has no knowledge about the value of each Web resource, he selects Web resources blindly or randomly. The user becomes more and more experienced while browsing more and more Web resources, so his judgement on the value of resources becomes more and more accurate. With time, the ultimate probability of browsing each Web resource should reflect the worthiness of each Web resource accurately. This is our main idea.

From Theorem 6, we know that the ultimate distribution vector $\boldsymbol{\pi}$ of a holomorphic Markov chain is independent of the initial distribution vector $\boldsymbol{p}(0)$, but totally determined by the transition probability matrix $\boldsymbol{P}$. So given a set of related Web resources, R, we can construct the transition probability matrix $\boldsymbol{P}$ through Theorem 4, then calculate the ultimate distribution vector based on Theorem 6. The ultimate distribution vector is just the rank of Web resources we are looking for. The group of equations in Theorem 6 can be solved using the 'Gaussian method' without any iteration. Some ad-hoc mathematical software (such as MatLab) has already provided such capability. Our initial implementation shows that the rank algorithm is efficient and scalable in practice.

The parameters in this rank algorithm $(\alpha, \beta, \gamma, \varepsilon)$ will be different for different users, and can be adjusted for particular needs. These parameters can also be automatically estimated based on the user's surfing history, which is discussed in another paper. In our experiment, the parameters are initially set as $\alpha = 0.6$, $\beta = 0.2$, $\gamma = 0.19$, $\varepsilon = 0.01$.

5. Conclusion

From a dynamics viewpoint, this paper provides a rank algorithm of Web resources based on a Markov model. The advantages of this rank algorithm are:

- several metrics (relevance, authority, integrativity and novelty) have been synthesized;
- the result can be calculated efficiently through solving a group of linear equations, without any iteration;
- the parameters can be customized and dynamically adjusted by the user.

Several researchers have pointed out that there is plenty of information buried in the user's bookmark and historic visits log [4]. Now we are investigating how to leverage this information in our rank algorithm. Classifying and clustering the Web resources automatically are also very important to RD on the Web [11,17]. We believe that the Markov model proposed in this paper can also be applied.

The Web can be viewed as a 'complex system', and nonlinear dynamics (chaos, fractal, and so on) should be very helpful to manage and make use of the Web. We believe that the World Wide Web (WWW) will eventually become an information retrieval tool whoever, whenever, wherever (WWW) you are.

References

[1] P. Bernstein, M. Brodie, S. Ceri, et al., The Asilomar Report on Database Research, Technical Report MSTR-TR-98-57, Microsoft Research, Microsoft Corporation, September 1998.
[2] K. Bharat and A. Broder, A technique for measuring the relative size and overlap of public Web search engines, Computer Networks and ISDN Systems 30 (1998) 379–388.
[3] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems 30 (1998) 107–117.
[4] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, Hypersearching the web, Scientific American, June 1999.
[5] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson and J. Kleinberg, Automatic resource compilation by analyzing hyperlinkage structure and associated text, Computer Networks and ISDN Systems 30 (1998) 65–74.
[6] Y. Chen, Web as A Data Source, Ph.D. Thesis, Department of Computer Science and Engineering, Southeast University, Nanjing, 1999.

[7] J. Cho, H. Garcia-Molina and L. Page, Efficient crawling through URL ordering, 7th International Web Conference (WWW 98), Brisbane, April 14–18, 1998.
[8] M. Iosifecu, Finite Markov Processes and Their Applications, Wiley, Chichester, 1980.
[9] Q.Y. Jiang, Mathematical Model (version 2), High Education Press, Beijing, 1993.
[10] S. Lorence and C.L. Giles, Inquirus, the NECI meta search engine, Computer Networks and ISDN Systems 30 (1998) 95–105.
[11] S.A. Macskassy, A. Banerjee, B.D. Davison, et al., Human performance on clustering Web pages: a preliminary study, in: Proc. of the 4th International Conference on Knowledge Discovery and Data Mining, New York, August 1998.
[12] M. Marchiori, The quest for correct information on the Web: hyper search engines, Computer Networks and ISDN Systems 29 (1997) 1225–1235.
[13] M. Marchiori, The limits of Web metadata, and beyond, Computer Networks and ISDN Systems 30 (1998) 1–9.
[14] G. Pringle, L. Allison and D.L. Dowe, What is a tall poppy among Web pages? Computer Networks and ISDN Systems 30 (1998) 369–377.
[15] C.M. Rowman, Scalable Internet resource discovery: research problems and approaches, Communications of the ACM 37 (8) (1994) 98–107.
[16] E. Spertus, ParaSite: mining structural information on the Web, Computer Networks and ISDN Systems 29 (1997) 1205–1215.
[17] M.R. Wulfekuhler and W.F. Punch, Finding salient feature for personal Web page categories, Computer Networks and ISDN Systems 29 (1997) 1147–1156.
[18] D.Y. Zhu, Mathematical Modeling Cases, Southeast University Press, Nanjing, 1999.

Dell Zhang was born in July 1976 in Yangzhou, China. He received his bachelor's degree in Computer Science from Southeast University, China, and is now a Ph.D. candidate in Computer Science there. He is a member of ACM and IEEE Computer Society. His main research interests are data mining, information retrieval, and evolutionary computing.

Yisheng Dong is a full professor of Computer Science, and also the dean of the Department of Computer Science and Engineering at Southeast University, China. He graduated in 1965, from Nanjing Institute of Technology, China. His research domain includes databases, software engineering and information systems.